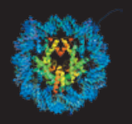


PACIFIC SYMPOSIUM ON
BIOCOMPUTING 2019

The Pacific Symposium on Biocomputing (PSB) 2019 is an international, multidisciplinary conference for the presentation and discussion of current research in the theory and application of computational methods in problems of biological significance. Presentations are rigorously peer reviewed and are published in an archival proceedings volume. PSB 2019 will be held on January 3 – 7, 2019 in Kohala Coast, Hawaii. Tutorials and workshops will be offered prior to the start of the conference.

PSB 2019 will bring together top researchers from the US, the Asian Pacific nations, and around the world to exchange research results and address open issues in all aspects of computational biology. It is a forum for the presentation of work in databases, algorithms, interfaces, visualization, modeling, and other computational methods, as applied to biological problems, with emphasis on applications in data-rich areas of molecular biology.

The PSB has been designed to be responsive to the need for critical mass in sub-disciplines within biocomputing. For that reason, it is the only meeting whose sessions are defined dynamically each year in response to specific proposals. PSB sessions are organized by leaders of research in biocomputing's "hot topics." In this way, the meeting provides an early forum for serious examination of emerging methods and approaches in this rapidly changing field.

Cover image:
This image depicts a molecular model of the Nucleosome (PDB ID: 1aoi, Luger et al. (1997) Nature 389, 251–260) — The nucleosome is the organising principle behind higher ordered chromatin structure. The histone core of the nucleosome exemplifies the many molecular mechanisms that have evolved to regulate access to the DNA in chromatin.

Image by D. Rey Banatao,
Pacific Symposium on Biocomputing.

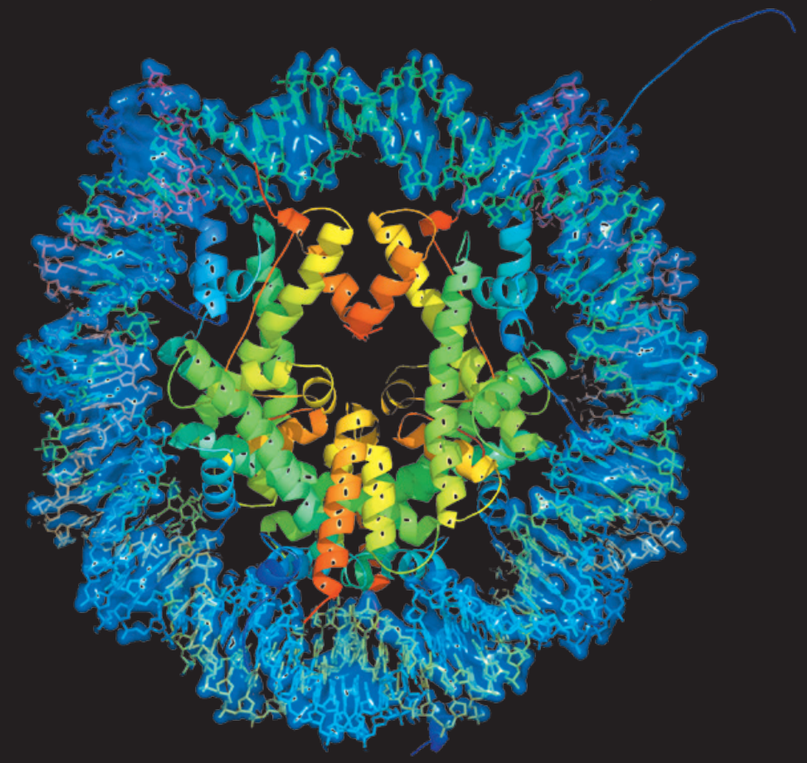
World Scientific
www.worldscientific.com
11263 hc

ISBN 978-981-3279-82-7

R. B. Altman
A. K. Dunker
L. Hunter
M. D. Ritchie
T. Murray
T. E. Klein

PACIFIC SYMPOSIUM ON
BIOCOMPUTING 2019

PACIFIC SYMPOSIUM ON BIOCOMPUTING 2019



Edited by

Russ B. Altman, A. Keith Dunker, Lawrence Hunter, Marylyn D. Ritchie, Tiffany Murray & Teri E. Klein

WORKSHOPS

PACIFIC SYMPOSIUM ON BIOCOMPUTING 2019

2019 marks the 24th Pacific Symposium on Biocomputing (PSB). The world is in a tizzy about big data, data science and AI (especially deep learning). Machine learning is everywhere and many of the tools and approaches that have been discussed at PSB for the last 24 years are becoming mainstream. This is in some ways gratifying and other ways worrisome, as the hype of these technologies is staggering. The PSB community, however, continues to innovate in the application of these ideas to critical problems in biology and medicine. More importantly, through peer review the PSB community has maintained a realistic understanding of the capabilities of emerging technologies. It is our duty to continue applying appropriate pressure on ourselves to test the real-world utility of these techniques, figure out how to optimize their use for problems in biology and medicine, and ensure that we contribute to a scholarly literature that realistically portrays the power and the limitations of emerging technologies. The focus of PSB on emerging scientific questions and methodologies is a clear strength of the conference, and one that we must protect and preserve.

PSB depends on the community to define emerging areas in biomedical computation. Its sessions are usually conceived at the previous PSB meeting as people discuss trends and opportunities for new science. The typical program includes sessions that evolve over two to three years as well as entirely new sessions. This year we revisit topics such as precision medicine, pattern recognition, while nurturing emerging interest in single cell analysis, privacy/ethics and other topics.

In addition to being published by World Scientific and indexed in PubMED, the proceedings from all PSB meetings are available online at http://psb.stanford.edu/psb-online/. PSB has 1125 papers listed in PubMED (as of today). These papers are routinely cited in archival journal articles and often represent important early contributions in new subfields—many times before there is an established literature in more traditional journals; for this reason, many papers have garnered hundreds of citations. The Twitter handle PSB 2019 is @PacSymBiocomp and the hashtag this year will be #psb19.

The efforts of a dedicated group of session organizers have produced an outstanding program. The sessions of PSB 2019 and their hard-working organizers are as follows:

Pattern recognition in biomedical data: challenges in putting big data to work
Shefali S. Verma, Dokyoon Kim, Anurag Verma, Christian Darabos

Precision medicine: improving health through high-resolution analysis of personal data
Steven Brenner, Martha Bulyk, Dana Crawford, Jill Mesirov, Alexander Morgan, Predrag Radivojac

Single cell analysis--what is in the future?
Lana Garmire, Guo-cheng Yuan, Rong Fan, Gene Yeo, John Quackenbush

When biology gets personal: hidden challenges of privacy and ethics in biological big data
Gamze Gursoy, Arif Harmanci, Haixu Tang, Erman Ayday, Steven E. Brenner

We are also pleased to present four workshops in which investigators with a common interest come together to exchange results and new ideas in a format that is more informal than the peer-reviewed sessions. For this year, the workshops and their organizers are:

Merging heterogeneous data to enable knowledge discovery
Martin G. Seneviratne, Tina Hernandez-Boussard, Michael Kahn

Reading between the genes: interpreting noncoding DNA in high throughput
Joanne Berghout, Yves A. Lussier, Francesca Vitali, Martha L. Bulyk, Maricel G. Kann, Jason H. Moore

Text mining and machine learning for precision medicine
Graciela Gonzalez, Hongfang Liu, Zhiyong Lu, Robert Leaman

Translational informatics of population health: how large biomolecular and clinical datasets unite
Yves A. Lussier, Atul Butte, Rong Chen, Haiquan Li, Jason H. Moore

The PSB 2019 keynote speakers are Russ Altman (Science keynote) and Lawrence Hunter (Ethical, Legal and Social Implications keynote).

Tiffany Murray has managed the peer review process and assembly of the proceedings since 2003, and also plays a key role in many aspects of the meeting. We are grateful for the support of the Cleveland Institute for Computational Biology, Second Genome, Icahn Institute for Data Science and Genomic Technology, Cipherome, and DNANexus for their support of PSB 2019. We also thank the National Institutes of Health[1] and the International Society for Computational Biology (ISCB) for travel grant support. The research parasite and symbiont awards benefit by support from: GigaScience, Lifebit, Communications Biology, and the Gordon and Betty Moore Foundation.

We are particularly grateful to the onsite PSB staff Al Conde, Paul Murray, Ryan Whaley, Mark Woon, BJ Morrison-McKay, Cynthia Paulazzo, Jackson Miller, Kasey Miller, Heather Sanchez, and Nicholas Murray for their assistance. We also acknowledge the many busy researchers who reviewed the submitted manuscripts on a very tight schedule. The partial list following this preface does not include many who wished to remain anonymous, and of course we apologize to any who may have been left out by mistake.

We look forward to a great meeting once again. Aloha!

Pacific Symposium on Biocomputing Co-Chairs,
October 13, 2018

Russ B. Altman
Departments of Bioengineering, Genetics, Medicine & Biomedical Data Science, Stanford University

A. Keith Dunker
Department of Biochemistry and Molecular Biology, Indiana University School of Medicine

Lawrence Hunter
Department of Pharmacology, University of Colorado Health Sciences Center

Marylyn D. Ritchie
Department of Genetics and Institute for Biomedical Informatics, University of Pennsylvania

Teri E. Klein
Departments of Biomedical Data Science & Medicine, Stanford University

[1] Funding for this conference was made possible (in part) by Grant # 5 R13 LM006766 – 21 from the National Library of Medicine. The views expressed in written conference materials or publications, and by speakers and moderators, does not necessarily reflect the official policies of the Department of Health and Human Services; nor does mention by trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

Thanks to the reviewers…

Finally, we wish to thank the scores of reviewers. PSB aims for every paper in this volume to be reviewed by three independent referees. Since there is a large volume of submitted papers, paper reviews require a great deal of work from many people. We are grateful to all of you listed below and to anyone whose name we may have accidentally omitted or who wished to remain anonymous.

Fadhl Alakwaa
Jessica Cooke Bailey
Anna Basile
Christopher Bauer
Brett Beaulieu-Jones
Asa Ben-Hur
Mary Regina Boland
Will Bush
Tiffany Callahan
Hannah Carter
Hao Chen
Ercument Cicek
James Costello
Dana Crawford
Christian Darabos
Devendra Dhami
Michel Dumontier
Eric Gamazon
Wendong Ge
Mario Giacobini
Dylan Glubb
Casey Greene
Rafael Guerrero
Gamze Gursoy
Jake Hall
Thomas Hampton
Arif Harmanci
Robert Hoehndorf
Yu-Han Hsu
Peizhao Hu
Ting Hu
Mathias Humbert
Haky Im
Ehsan Imani
Xiaoqian Jiang
Yuxiang Jiang
Dokyoon Kim
Younghee Lee
Haiquan Li
Lang Li
Ruowang Li
Zhandong Liu
Ana Hernandez Lopez
Shaoke Lou
Jose Lugo-Martinez
Subha Madhavan
Ahmed Methwally
Jason Miller
Tejaswini Mishra
Noman Mohammed
Jason Moore
Sara Nasser
Sriraam Natarajan
Randal Olson
Alena Orlenko
Patryk Orzechowski
Kymberleigh Pagel
Gaurav Pandey
Vikas Pejaver
Yisu Peng
Thomas Peterson
Rani Powers
Sriram Sankararaman
Alfred Schissler
Andrew Su
Aik-Choon Tan
Jessie Tenenbaum
Gregg Thomas
Ryan Urbanowicz
Rami Vanguri
Olivia Veatch
Shefali Setia Verma
Yogasudha Veturi
Francesca Vitali
Slobodan Vucetic
Justin Wagner
Richard Wang
Shuang Wang
Wenhao Wang
Jonathan Warrell
Martha White
Scott Williams
John Witte
Jiwen Xin
Jingjing Yang
Amrapali Zaveri
Marinka Zitnik

Session Introduction
Pattern Recognition in Biomedical Data: Challenges in putting big data to work

Shefali Setia Verma
University of Pennsylvania
Philadelphia, PA 19104

Anurag Verma
University of Pennsylvania
Philadelphia, PA 19104

Dokyoon Kim
Geisinger
100 North Academy Avenue
Danville, PA 17822

Christian Darabos
Research Computing Services, Dartmouth College,
HB 6129
Hanover, NH 03755

Introduction

Technological advances are leading to an exponential increase in the size of biomedical data. Demand is high for novel computational techniques that can cope with these large datasets and have the potential to support translational research. Methods to analyze biomedical data in order to handle its complexities require sophisticated algorithms for pattern recognition and to handle complexities such as sparesenss and noisiness in these datasets. The availability of high throughput techniques in generating highly resourceful multi-omic biomedical data (genomic, transcriptomic and epigenomic to name a few) gave rise to a whole new set of challenges in identifying patterns. Modern statistical, machine learning, and even artificial intelligence (AI) methods can be used to integrate multiple resources to understand complex phenotypic traits. However, most of these methods pose multiple challenges either in fitting models or in analyzing the resulting models, whether using multiple species or multi-omic datasets for the

same species. This session focuses on innovative ways to address the challenges arising from the quality and quantity of data and also integrating biomedical data from various sources to identify patterns in biomedical datasets[1–3].

While cloud computing aids in analysis performance by improving computing time and storage, it is limited to the software package and there is considerable room for improvement in the cloud-based big-data analysis. Our session also aims at discussing the optimization of tool development for large scale datasets and challenges that are associated with the computational cost as well as resources for pattern recognition. Manuscripts listed in this session can be classified into following 4 categories:

1. ***Identifying patterns in EHR data***:
   Electronic Health Records (EHRs) is a collection of longitudinal health information from an individual's point of care. It includes diagnosis, procedure, laboratory measurement, medication, imaging, and clinical note. Many retrospective case-control studies have already demonstrated meaningful use of EHR data and its potential to improve understanding of disease risk and prevalence in the general populatio[4–7]. However, the data within EHR has not been utilized to its full extent due to several challenges, such as missing data, institutional biases in coding practice, and high throughput electronic phenotyping.

   In the manuscript titled "*Learning Contextual Hierarchical Structure of Medical Concepts to Clarify Phenotypes*", *Beaulieu-Jones et al* present an innovative application of Pointcaré embeddings to model data-driven hierarchy of ICD-9 diagnosis codes. The Pointcaré embeddings approach uses hyperbolic space to learn the embedding from a vector of nodes in a network graph as opposed to traditional Euclidean space-based methods such as Word2Vec[8]or GloVe[9] Since it is shown that the hyperbolic space is more appropriate for hierarchical information[10], so its application of ICD-9 codes shows potential in improving phenotype definitions while keeping the global structure and hierarchy of ICD-9 codes.

   Similarly, as the new methods are showing improvement in electronic phenotyping in EHR data, it is also important to identify patient cohort for a disease more accurately. In manuscript titled "*The Effectiveness of Multitask Learning for Phenotyping with*

Electronic Health Records Data", *Ding et al* investigated the effectiveness of a supervised approach called Multitask Learning (MTL) to define phenotypes using EHR data. Authors demonstrated that MTL approach performed better for complex phenotype definition whereas traditional supervised approaches such as linear models can be preferable for simple phenotype definitions.

Integrating EHR data from various health providers across the country has great potential to predict disease risk across the large population. However, there are various disparities across different health providers such as clinical care bias, population differences, ethical, and privacy policies. In the manuscript "*ODAL: A one-shot distributed algorithm to perform logistic regressions on electronic health records data from multiple clinical sites*", *Duan et al* propose an algorithmic approach to integrate EHR data from multiple health providers in an efficient way, and preserving privacy. They propose a use of a common data model developed by Observational Health Data Sciences and Informatics (ODSHI) and further perform statistical analysis in a distributed manner across multiple sites. Authors address a key issue of data sharing using ODAL by performing large-scale association analysis without explicitly sharing of sensitive data.

2. ***Machine/Deep Learning approaches:***

The current explosion of biomedical big data, including imaging, genomic, and EHR, provide a great opportunity to improve understanding of the genetic architecture of complex diseases and ultimately to improve health care. With the explosion of the biomedical big data, machine learning and deep learning techniques are becoming an integral component of evaluating biomedical data. In particular, deep learning has been extensively used in the field of biomedical informatics, such as healthcare and genomic data analyses as well as text mining.

In the context of healthcare data analysis, the accurate detection of premature ventricular contractions (PVC) in patients is an important task in cardiac care for some patients. *Gordon et al* developed a novel PVC detection algorithm based around a convolutional autoencoder to address the weaknesses, such as the need to use difficult to extract morphological features, domain-specific features, or large number of estimated parameters, and validated their method using the MIT-BIH arrhythmia

database. Although many deep learning methods have been shown with great successes in biomedical informatics, the "black-box" nature of deep learning and the high-reliability requirement of biomedical applications have created new challenges regarding the existence of confounding factors. In the manuscript titled "Removing Confounding Factors Associated Weights in Deep Neural Networks Improves the Prediction Accuracy for Healthcare Applications", *Wang et al* present an efficient method that can remove the influences of confounding factors, such as age or gender, to improve the across-cohort prediction accuracy of deep neural networks.

Deep learning is also applied to many genomic data analyses. Protein domain boundary prediction is usually an early step to understand protein function and structure. Most of the current computational domain boundary prediction methods suffer from low accuracy and limitation in handling multi-domain types, or even cannot be applied on certain targets, such as proteins with the discontinuous domain. *Jiang et al* developed an *ab-initio* protein domain predictor using a stacked bidirectional Long Short-Term Memory Units (LSTM) model in deep learning. Additionally, a deep residual network (deep ResNet) is a type of specialized neural network that helps to handle more sophisticated deep learning tasks and models. *Liu et al* describe the use of a deep ResNet-based model that fuses flanking DNA sequence information with additional SNP annotation information for identifying functional noncoding SNPs in trait-associated regions. As another interesting study, steganography serves to conceal the existence and content of messages in the media using various techniques. Recent advances in next-generation sequencing technologies have facilitated the use of deoxyribonucleic acid (DNA) as a novel covert channel in steganography. *Bae et al* propose a general sequence learning-based DNA steganalysis framework using deep recurrent neural networks (RNNs). The proposed approach learns the intrinsic distribution of coding and non-coding sequences and detects hidden messages by exploiting distribution variations after hiding these messages.

In addition to many applications, deep learning technique is widely used in text mining. Phylogeography research involving virus spread and tree reconstruction relies on accurate geographic locations of infected hosts. Insufficient level of geographic information in nucleotide sequence repositories such as GenBank motivates the use of

natural language processing methods for extracting geographic location names (toponyms) in the scientific article associated with the sequence and disambiguating the locations to their coordinates. *Magge et al* present an extensive study of multiple recurrent neural network architectures for the task of extracting geographic locations and their effective contribution to the disambiguation task using population heuristics. Additionally, in the manuscript titled "Automatic Human-like Mining and Constructing Reliable Genetic Association Database with Deep Reinforcement Learning", *Wang et al* aim to improve the reliability of biomedical text-mining by training the system to directly simulate the human behaviors, such as querying the PubMed, selecting articles from queried results, and reading selected articles for knowledge. They take advantage of the efficiency of biomedical text-mining, the flexibility of deep reinforcement learning, and the massive amount of knowledge collected in UMLS into an integrative artificial intelligent reader that can automatically identify the authentic articles and effectively acquire the knowledge conveyed in the articles.

Although classification has been extensively studied over the past decades, there remain understudied problems when the training data violate the main statistical assumptions relied upon for accurate learning and model characterization. This particularly holds true in the open world setting where observations of a phenomenon generally guarantee its presence, but the absence of such evidence cannot be interpreted as the evidence of its absence. Learning from such data is often referred to as positive-unlabeled learning, a form of semi-supervised learning where all labeled data belong to one (say, positive) class. To improve the best practices in the field, *Ramola et al* study the quality of estimated performance accuracy in positive-unlabeled learning in the biomedical domain.

3. *Identifying patterns in omic data sets:*

Complex traits are often heterogeneous in nature, which means that they are likely not only explained by one data type (for example genomic variations). Thus, integrative methods in combining data from various sources (on same or different samples) is demanding. *Graim et al* present a new method for integrating multiple data types to predict cancer-drug sensitivity. The proposed method PLATYPUS (Progressive LAbel Training bY Predicting Unlabeled Samples) combines prior knowledge with raw input data to make predictions in testing dataset. This method when compared to

ensemble approach on using single dataset yields better prediction even in samples where missingness is observed. *Marty et al* represent an integrative approach for utilizing exome and transcriptome to study the highly heterogeneous Killer Immunoglobulin-like receptor (KIR) region that is known to be associated with cancer phenotypes. Lastly, *Pyman et al* use deep learning methods to classify 26 types of cancer cells from normal tissue cell by analyzing microRNA dataset.

Understanding gene function is an important aspect of interpretation of findings. Rapid advancements have been made in sequencing microbial genome. *Li et al* present a Bayesian approach to analyze transposon mutagenesis with next generation sequencing (TnSeq) data. *Anand et al* represent a method to link non-coding variants to gene functions by using CHIP-Seq data for interpreting association study signals.

Publicly available open source large datasets also provide unique opportunities for pattern recognition. Leveraging these resources are highly important. *Tsui et al* utilized datasets from Sequence Read Archive (SRA) and designed a pipeline to extract allele counts from variety of datasets, such as RNA seq, whole exome sequencing and whole genome sequencing.

4. ***Computational challenges:***

The data-intensive nature of the computational problem in the field of biomedical informatics also warrants the development of software approaches to efficiently use the existing institutional computer infrastructure as well as cloud computing. Additionally, the tools and workflows are changing at a rapid pace as new data types are being generated from new techniques in biology such as sequencing, gene expression data, among others. This raises two key issues: assessment of new software workflows and their reproducibility. There is community effort like Dialogue for Reverse Engineering Assessments and Methods (DREAM) Challenges to compare and benchmark new tools and workflows. In the manuscript "*A Workflow-based Approach to Benchmark Challenges Enhances Reusability, and Reproducibility*", *Srivastava et al* present an approach to improve the reproducibility and interpretability associated with bioinformatics benchmark challenges. To achieve this, the authors used the WINGS system as the model and modified it to allow each step of the submitted algorithms to be analysed.

References

1. Bourne, P. E.; Bonazzi, V.; Dunn, M.; Green, E. D.; Guyer, M.; Komatsoulis, G.; Larkin, J.; Russell, B. The NIH Big Data to Knowledge (BD2K) initiative. *J Am Med Inform Assoc* **2015**, *22*, 1114, doi:10.1093/jamia/ocv136.
2. Ritchie, M. D.; Holzinger, E. R.; Li, R.; Pendergrass, S. A.; Kim, D. Methods of integrating data to uncover genotype-phenotype interactions. *Nat. Rev. Genet.* **2015**, *16*, 85–97, doi:10.1038/nrg3868.
3. Pasaniuc, B.; Price, A. L. Dissecting the genetics of complex traits using summary association statistics. *Nat. Rev. Genet.* **2017**, *18*, 117–127, doi:10.1038/nrg.2016.142.
4. Verma, A.; Verma, S. S.; Pendergrass, S. A.; Crawford, D. C.; Crosslin, D. R.; Kuivaniemi, H.; Bush, W. S.; Bradford, Y.; Kullo, I.; Bielinski, S. J.; Li, R.; Denny, J. C.; Peissig, P.; Hebbring, S.; De Andrade, M.; Ritchie, M. D.; Tromp, G. eMERGE Phenome-Wide Association Study (PheWAS) identifies clinical associations and pleiotropy for stop-gain variants. *BMC Medical Genomics* **2016**, *9*, 32, doi:10.1186/s12920-016-0191-8.
5. Verma, S. S.; Lucas, A. M.; Lavage, D. R.; Leader, J. B.; Metpally, R.; Krishnamurthy, S.; Dewey, F.; Borecki, I.; Lopez, A.; Overton, J.; Penn, J.; Reid, J.; Pendergrass, S. A.; Breitwieser, G.; Ritchie, M. D. IDENTIFYING GENETIC ASSOCIATIONS WITH VARIABILITY IN METABOLIC HEALTH AND BLOOD COUNT LABORATORY VALUES: DIVING INTO THE QUANTITATIVE TRAITS BY LEVERAGING LONGITUDINAL DATA FROM AN EHR. *Pac Symp Biocomput* **2016**, *22*, 533–544.
6. Hoffmann, T. J.; Ehret, G. B.; Nandakumar, P.; Ranatunga, D.; Schaefer, C.; Kwok, P.-Y.; Iribarren, C.; Chakravarti, A.; Risch, N. Genome-wide association analyses using electronic health records identify new loci influencing blood pressure variation. *Nat Genet* **2017**, *49*, 54–64, doi:10.1038/ng.3715.
7. Singh, A.; Nadkarni, G.; Gottesman, O.; Ellis, S. B.; Bottinger, E. P.; Guttag, J. V. Incorporating temporal EHR data in predictive models for risk stratification of renal function deterioration. *J Biomed Inform* **2015**, *53*, 220–228, doi:10.1016/j.jbi.2014.11.005.
8. Mikolov, T. Efficient Estimation of Word Representations in Vector Space., doi:arXiv:1301.3781.
9. Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1532–1543.
10. Krioukov, D.; Papadopoulos, F.; Kitsak, M.; Vahdat, A.; Boguñá, M. Hyperbolic geometry of complex networks. *Physical Review E* **2010**, *82*, doi:10.1103/PhysRevE.82.036106.

Learning Contextual Hierarchical Structure of Medical Concepts with Poincairé Embeddings to Clarify Phenotypes

Brett K. Beaulieu-Jones, Isaac S. Kohane and Andrew L. Beam†

Department of Biomedical Informatics, Harvard Medical School, Boston, MA 02115, USA

†*E-mail: Andrew_Beam@hms.harvard.edu*

dbmi.hms.harvard.edu

Biomedical association studies are increasingly done using clinical concepts, and in particular diagnostic codes from clinical data repositories as phenotypes. Clinical concepts can be represented in a meaningful, vector space using word embedding models. These embeddings allow for comparison between clinical concepts or for straightforward input to machine learning models. Using traditional approaches, good representations require high dimensionality, making downstream tasks such as visualization more difficult. We applied Poincaré embeddings in a 2-dimensional hyperbolic space to a large-scale administrative claims database and show performance comparable to 100-dimensional embeddings in a euclidean space. We then examine disease relationships under different disease contexts to better understand potential phenotypes.

Keywords: Clinical Concept Embeddings, Poincaré, Contextual Disease Relationships, Context-dependent Phenotypes, Deep Learning.

1. Introduction

Word embeddings[1] are a popular way to represent natural language and have seen wide use in machine learning applied to document classification,[?,?] machine translation,[?,?] sentiment analysis,[2] and question answering.[3,4] Clinical concept embeddings extend this approach to model healthcare events,[5–8] and have been particularly useful modeling longitudinal clinical data.[?,9–11] Traditional approaches such as word2vec[1] and GloVe[12] embed entities within a Euclidean space.

However, recent work by Nickel and Kiela on *Poincaré embeddings*[13] claims to provide better embedding representations of hierarchically structured data using a hyperbolic embedding space within the Poincaré ball. This n-dimensional hyperbolic space has a significantly higher capacity than the Euclidean space, which allows it to effectively embed structured trees while preserving distance relationships.[14–17] Moreover, this space allows for embedding of hierarchical, tree-like structures, as Nickel and Kiela[13] observed high fidelity embeddings of ontologies. This has an obvious relevance to medical concepts, given many have an inherent tree structure (e.g. disease nosology) that should be recapitulated in the embedding space.

When clinicians consider a disease, they examine the disease in the context of the patient's

overall environment.[18] For example, renal failure caused by poor blood flow to the kidneys as a result of long-term hypertension would be considered differently from renal failure as the result of a specific infection or immune system disorder like Lupus.[19] Accurate and precise phenotyping is critical to modern clinical studies using the electronic healthcare record (EHR) and other '-omic' associations studies (e.g. genomic, transcriptomic, metabolomic). Misclassified phenotypes have a severe effect on tests of association and require increased sample sizes to maintain constant power.[20–22] Increases in genetic testing and the availability of clinical data repositories (Electronic Health Record, Administrative Claims, large-scale Cohort) have enabled PheWAS association studies to be performed without the need to target and recruit specific populations for each individual study.[23–25] It is important to develop methods that enable researchers to consider a specific disease or phenotype in the context of the overall patient and environment.

We applied Poincaré embeddings to a large-scale administrative claims database to examine how the relationships of different conditions changed in distinct contexts. Our hypothesis was that the increased representational capacity offered by Poincaré embeddings and their ability to naturally model hierarchical data would result in improved embeddings for clinical concepts. We first demonstrate this by showing they can accurately reconstruct the ICD-9 hierarchy on synthetic data. Next we show that they find an improved representation on real data relative to traditional embedding approaches at the same number of dimensions. We conclude with a disease-specific embedding hierarchy within an obese population. Our results could provide a better representation of disease and allow for more accurate machine learning models as well as the fine-tuning of targeted phenotypes for association studies.

2. Methods

To examine the effectiveness of Poincaré embeddings for clinical concept embedding, we: 1.) trained Poincaré embeddings on the ICD-9 hierarchy as validation of parent-child tuples, 2a.) selected and preprocessed chronological member sequences of each diagnosis experienced for a specified cohort (e.g. obese vs. no metabolic disorders diagnosed), 2b.) Learned distributed vector representations for the real data by training a Poincaré embedding model in a two-dimensional space. 3.) Visualized the Poincaré embeddings in a two dimensional space. 4a.) Constructed a distance matrix within the hyperbolic space. 4b.) Analyzed the distance matrix to measure how effectively the embeddings represent clinical groupings (e.g. ICD9 Chapter, Sub-chapter and major codes).

2.1. *Source Code*

The source code used for the analyses in this work are freely available on Github (https://github.com/brettbj/poincareembeddings) under a permissive open source license. The optimized C++ Poincare Embedding implementation by Tatsuya Shirakawa is available under the MIT license (https://github.com/TatsuyaShirakawa/poincare-embedding).

2.2. *Data Source*

These analyses were performed using de-identified insurance administration data including diagnostic billing codes from January 1, 2008 until February 29, 2016 for more than 63 million members. The database does not include any socioeconomic, race or ethnicity data. The Institutional Review Board at Harvard Medical School waived the requirement for approval as it deemed analyses of the de-identified dataset to be non-human subjects research.

The data to rebuild the reference ICD9 hierarchy tree is available in the GitHub repository (https:/github.com/brettbj/poincareembeddings/data/icd9.tsv).

2.3. *Data Selection and Preprocessing*

2.3.1. *Reference ICD9 Example*

We first benchmarked against a known hierarchy, the ICD9 2015-Clinical Modification code ontology. To do this we extracted the ICD9 codes into four levels: 1.) Chapters (e.g. codes 390-459: Diseases of the circulatory system), 2.) Sub-chapters (e.g. codes 401-405: Hypertensive disease), 3.) Major Codes (e.g. code 401: Essential hypertension), and 4.) Detail level codes (e.g. code 401.0: Hypertension, malignant). We assigned relationships between each detail level code and the chapter, sub-chapter and major code it belonged to, each major code to the appropriate sub-chapter and chapter, and each sub-chapter to the chapter it belonged to.

2.3.2. *Real Member Analyses*

We performed cohort analyses by defining different study groups. First we included ten million randomly selected members (without replacement) who were enrolled for at least two years from the database of 63 million members. Next we separated two groups based on obesity diagnoses: 1.) ten million members who do not have a diagnosis for metabolic disorders with ICD9 codes between 270 and 279 2.) 3.38 million members who were diagnosed with obesity ICD9 codes (278.00 and 278.01).

Poincaré embeddings learn distributed vector representations from hierarchical data (e.g. a directed graph or tree). The input to the model is a list of tuples of the form $< A, B >$, which indicates that A and B have some form of unspecified relationship (e.g. *parent of*, *co-occurs with*, etc). In our case, the list of relationships specify that two diagnoses occurred sequentially, within a one year period, and had to occur more than ten total times and in more than 2% of all diagnoses.

2.4. *Poincaré Embeddings*

The key way in which Poincaré embeddings differ from traditional approaches is the distance metric which is used to compare the embeddings for two concepts. This distance metric is given in equation 1:

$$dist((x_1, y_1), (x_2, y_2)) = arccosh(1 + \frac{(x_2 - x_1)^2 + (y_2 - y_1)^2}{2y_1y_2}) \quad (1)$$

Equation 1 shows the distance between two points in the Poincaré ball hyperbolic space.

Training a Poincaré embedding model occurs by maximizing the distance (Equation 1) between unconnected nodes or diagnoses while minimizing the distance between highly connected nodes. This is done using a stochastic Riemannian optimization method, specifically stochastic gradient descent on riemmanian manifolds as seen in Bonnabel.[15]

2.5. *Processing and Evaluating Embeddings*

Once each concept is embedded into a two dimensional space, it is possible to calculate the pair-wise distance between all concepts using Equation 1. To assess how well the embeddings captured the ICD hierarchy on real data, we compared the average distances between concepts in the same ICD9 major code, sub-chapter and chapter against the distances of all other concepts. We then compared the capacity of a two-dimensional Poincairé space with varying size euclidean spaces. To do this, we repeated distance calculations with the clinical concept embeddings trained in a euclidean space on more than 63 million members in 2, 10 and 100 dimensions from Beam et al.[5] To normalize the distance comparisons between hyperbolic and euclidean spaces, we compared the ratio of distances between ICD codes within the same major, sub-chapter and chapter and the other ICD codes outside of the major, sub-chapter, and chapter.

3. Results

3.1. *ICD9 Hierarchy Evaluation*

To evaluate the method with a known ground truth, we embedded the ICD9 hierarchy and then reconstructed it as a tree. Because there are no counts included, stochasticity for all relationships at the same level (Chapter, Sub-chapter, Major, Detail) was expected. Figure 1 shows the reconstructed tree of the predefined ICD9 tree. This served as evidence that Poincairé embeddings can effectively embed a clean ICD9 hierarchy.

3.2. *Poincaré Embeddings on 10 Million Members*

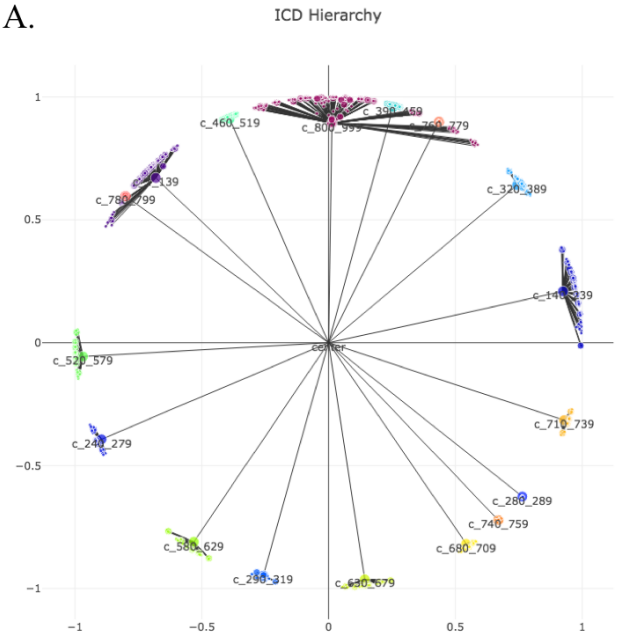
We then trained Poincaré embeddings in a two-dimensional space for 10 million randomly selected members (Table 1).

Table 1 Member Demographics of the Training Data

Demographics	
Male	40.4%
Female	59.6%
Age (2016)	48.66 (22.68)
ICD9 Diagnoses	22.38 (28.70)

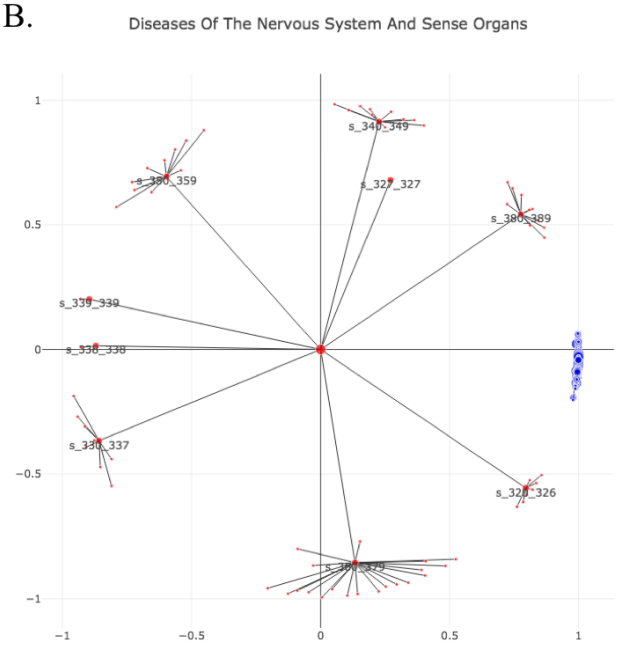
Figure 2A shows the ICD9 concepts (labeled by chapter) embedded in a two-dimensional space. While there were over 223 million total diagnoses, the majority of concepts had less than 100 distinct relations (Figure 2B) and the number of distinct relations was correlated with the distance from the origin ($R^2 = 0.61$) (Figure 2C).

A.



ICD Chapter	Code Range
Infectious and Parasitic Diseases	1-139
Neoplasms	140-239
Endocrine, Nutritional And Metabolic Diseases, And Immunity Disorders	240-279
Diseases Of The Blood And Blood-Forming Organs	280-289
Mental Disorders	290-319
Diseases Of The Nervous System And Sense Organs	320-389
Diseases Of The Circulatory System	390-459
Diseases Of The Respiratory System	460-519
Diseases Of The Digestive System	520-579
Diseases Of The Genitourinary System	580-629
Complications Of Pregnancy, Childbirth, And The Puerperium	630-679
Diseases Of The Skin And Subcutaneous Tissue	680-709
Diseases Of The Musculoskeletal System And Connective Tissue	710-739
Congenital Anomalies	740-759
Certain Conditions Originating In The Perinatal Period	760-779
Symptoms, Signs, And Ill-Defined Conditions	780-799
Injury And Poisoning	800-999

B.



ICD Chapter	Code Range
Diseases of the Nervous System and Sense Organs	1-139
Other	140-239

Fig. 1. ICD Example All codes

Figure 2 shows that the ICD hierarchy is correctly reconstructed using by the Poincaré embeddings in two dimensions. The distances between ICD codes in the same major, sub-chapter and chapter are smaller than the distances across different major codes, sub-chapters and chapters (Table 2). This shows that Poincaré embeddings are representing the data in a way that has similarities with the human-defined ICD9 hierarchy.

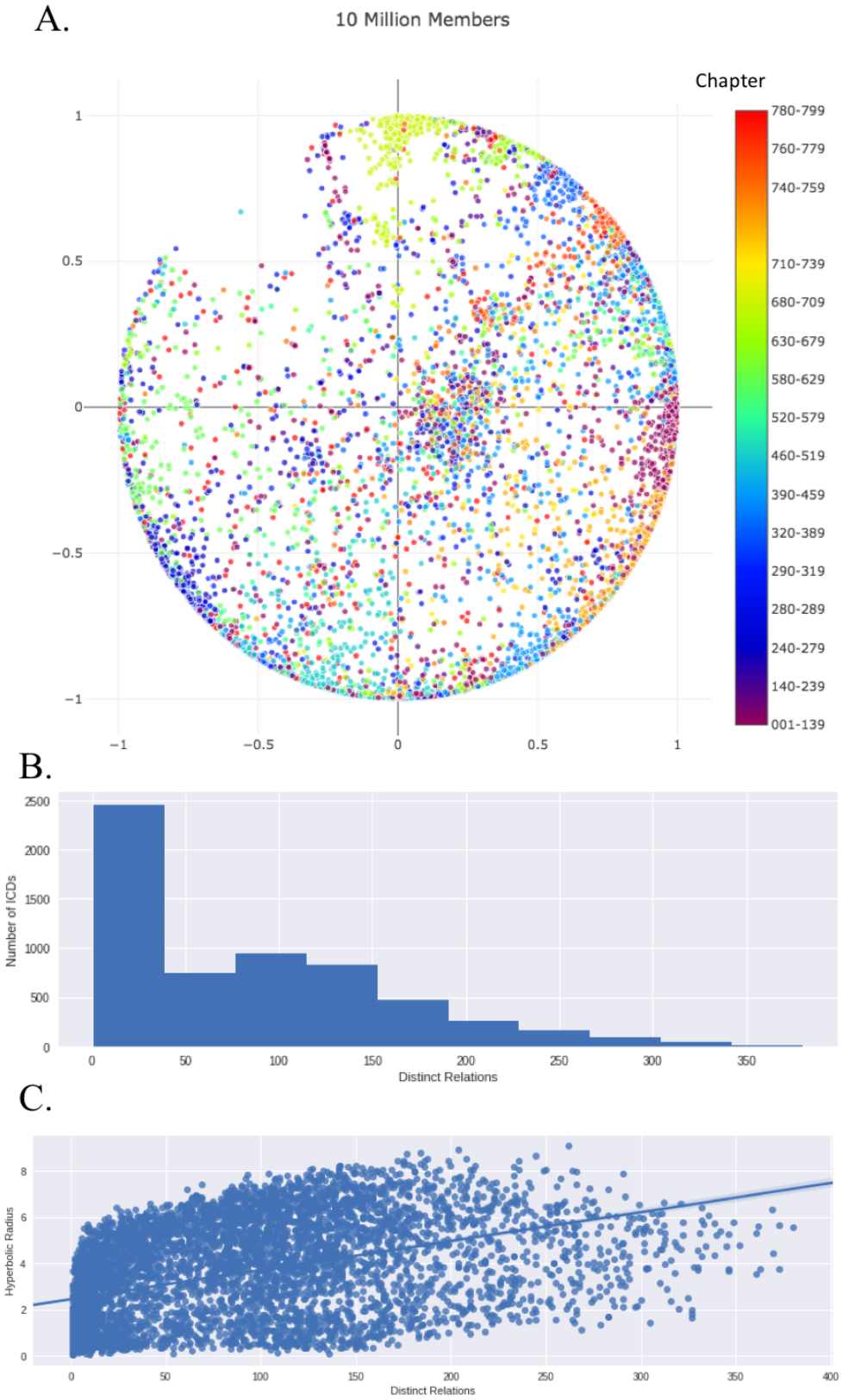


Fig. 2. A.) ICD9 Diagnoses Codes Embedded in a two-dimensional space. B.) Examination of the number of distinct relations for each ICD9 code. C.) Examination of the Correlation between the number of distinct relations and hyperbolic distance.

Table 2. Hyperbolic Distance comparison within Major, Sub-chapter and Chapter

Category	In Category	Outside of Category
Major	3.87 (1.71)	5.89 (1.92)
Sub-chapter	4.47 (1.73)	5.89 (1.92)
Chapter	4.91 (1.81)	5.91 (1.94)

3.3. *Comparison with Euclidean Embeddings*

To evaluate Poincaré embeddings against traditional euclidean embeddings, we compared the 2-dimensional Poincaré embeddings with 2, 10 and 100 dimension embeddings. The Poincaré embeddings were trained on 10 million randomly selected members. Running the preprocessing pipeline required 42 minutes on 16 cores but training the embeddings required only 49 seconds on 16 cores. All euclidean embeddings were trained on more than 63 million members. Table 3 shows the ratios of the mean distances of ICD codes in the same category over ICD codes in all other categories. We show the ratio to allow for comparison between Poincaré and Euclidean distances. As the dimensionality of the euclidean embeddings increased, the ratio of distance in-group vs. out of group decreased, indicating that the higher capacity enabled a better representation. The 2-dimensional Poincaré embeddings compared most closely to the 100-dimensional euclidean embeddings.

Table 3 Distance (ratio) comparison between Poincaré (2-dimensional) and Euclidean (2, 10, & 100-dimensional) within Major, Sub-chapter and Chapter.

Category	Poincaire (2d)	Euclidean (2d)	Euclidean (10d)	Euclidean (100d)
Major	0.657	0.758	0.668	0.649
Sub-chapter	0.759	0.863	0.794	0.774
Chapter	0.831	0.894	0.856	0.830

3.4. *Cohort Specific Embeddings*

Finally, we trained two separate Poincaré embeddings on patients with either: 1.) no prior diagnoses from the sub-chapter of metabolic disorders between ICD code 270 and 279 (N=10,000,000) and 2.) members diagnosed with obesity (ICD codes 278.00, 278.01, N=3,377,267) to first visualize the differences in the context of type 2 diabetes mellitus (Figure 3). Because the Poincaré embedding model was trained in 2-dimensions this was done without any further dimensionality reduction step.

We then examined the diseases in the closest quartile of either cohort to determine which showed the greatest movement from type 2 diabetes (Table 4). Of note, 22 of the top 50 were pain related and there are numerous links in the literature between both obesity (particularly joint and fibromyalgia[26,27]) and type 2 diabetes (particularly neuropathy[28]) with pain.

4. Discussion and Conclusion

Machine learning has great potential to improve the delivery of healthcare to patients, but many methodological challenges remain before this potential can be realized.[29,30] In this work,

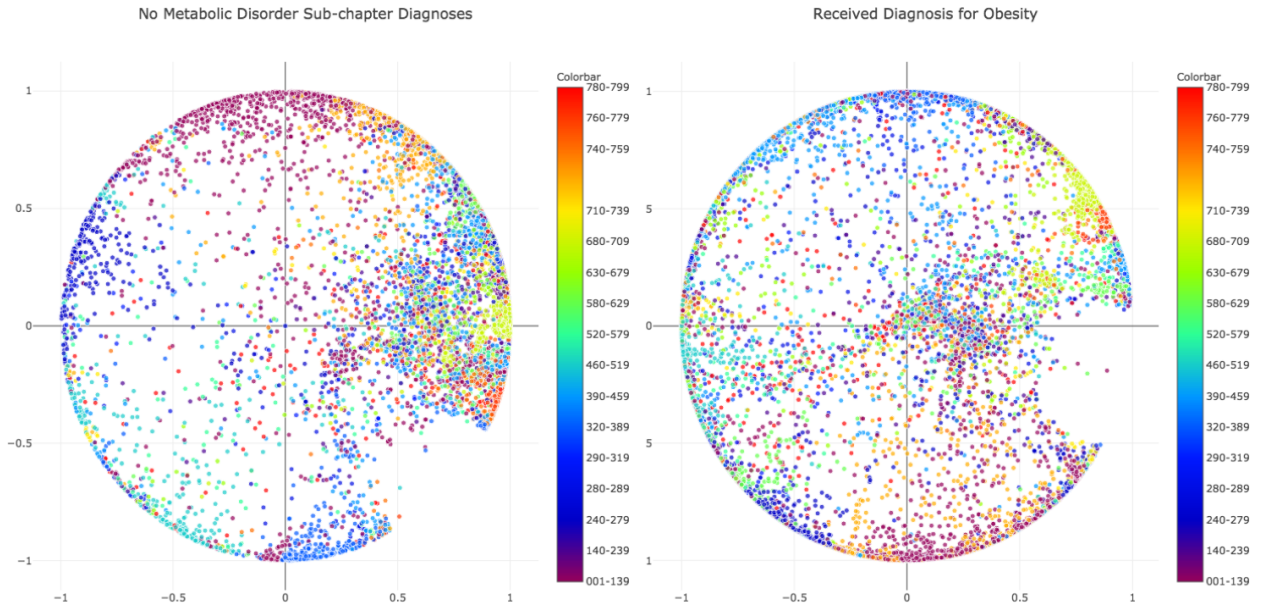


Fig. 3. A.) Poincaré Embeddings trained on 10M members with no metabolic disorder diagnoses (centered on type 2 diabetes). B.) Poincaré Embeddings trained on 3.38M members diagnosed with obesity (centered on type 2 diabetes).

Table 4. ICD9 Codes with the largest changes in distance from Type 2 Diabetes (250.00).

	ICD	Description
1	553.21	Incisional hernia
2	786.09	Other Respiratory Abnormalities
3	599.0	Urinary tract infection
4	285.9	Anemia
5	571	Chronic Liver Disease
6	583.6	Nephritis
7	724.5	Backache, unspecified
8	710.5	Eosinophilia myalgia syndrome
9	796.2	Elevated blood pressure w/o hypertension
10	719.46	Pain in Leg

we showed the increased capacity and hierarchical positioning of Poincaré embedding models can be useful to learn representations of disease diagnosis codes. Two-dimensional Poincaré embeddings were on par with 100-dimension euclidean embeddings when compared to the human-defined ICD hierarchy. Importantly the extra capacity of Poincaré embeddings may directly allow for visualization in a two-dimensional space, while traditional euclidean embedding techniques require an additional dimensionality reduction step (PCA, t-SNE, UMAP). Many of these techniques are non-deterministic and may not preserve global structure.

An important limitation of our current method is that the pre-processing step constructs binary relations between concepts whenever they occur with a specified threshold (more than

10 occurrences and 2% of cases). It is likely that additional information could be learned by encoding the actual frequency between concepts. In addition, it could be useful to evaluate additional distance matrices that have worked well for hierarchical problems in other domains, such as pg-gram and Edit distance.[31]

There are significant opportunities to expand on and apply these techniques to biomedical domains in order to examine and consider phenotypic context when performing associations. We are especially interested in the ability to contextualize a phenotype for association studies by considering the way ICD code relationships change given comorbidities. For example, start by measuring the way Poincaré embeddings change given a comorbidity (e.g. type 2 diabetes given metabolic disorder). If there are significant changes, it may be helpful to design association studies to separate endpoints, for example diabetes with no prior metabolic disorders and diabetes with prior metabolic disorders. In this case, the disease etiology may be distinct, and therefore we would expect the potential for different genetic drivers.

5. Acknowledgments

The authors thank Tatsuya Shirakawa for developing and open-sourcing an efficient implementation of the Poincaré Embedding Model. This work was supported in part by NLM grant 4 T15 LM007092-25.

References

1. B. T. Mikolov, K. Chen, G. Corrado and J. Dean, *arXiv:1301.3781* (2013).
2. C. R. Association for Computational Linguistics. Meeting (45th : 2007 : Prague, R. E. Association for Computational Linguistics., P. T. Pham, D. Huang, A. Y. Ng and C. Potts, *ACL 2007 : proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic.* (Association for Computational Linguistics, 2007).
3. J. Zhou and O. G. Troyanskaya, *Nature Methods* **12**, 931 (2015).
4. A. Bordes, J. Weston and N. Usunier, Open Question Answering with Weakly Supervised Embedding Models (Springer, Berlin, Heidelberg, 2014) pp. 165–180.
5. A. L. Beam, B. Kompa, I. Fried, N. P. Palmer, X. Shi, T. Cai and I. S. Kohane, *arXiv preprint arXiv:1804.01486* (2018).
6. Y. Choi, C. Y.-I. Chiu and D. Sontag, *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science* **2016**, 41 (2016).
7. T. Ching, D. Himmelstein, B. Beaulieu-Jones, A. Kalinin, B. Do, G. Way, E. Ferrero, P.-M. Agapow, M. Zietz, M. Hoffman, W. Xie, G. Rosen, B. Lengerich, J. Israeli, J. Lanchantin, S. Woloszynek, A. Carpenter, A. Shrikumar, J. Xu, E. Cofer, C. Lavender, S. Turaga, A. Alexandari, Z. Lu, D. Harris, D. Decaprio, Y. Qi, A. Kundaje, Y. Peng, L. Wiley, M. Segler, S. Boca, S. Swamidass, A. Huang, A. Gitter and C. Greene, *Journal of the Royal Society Interface* **15** (2018).
8. B. Beaulieu-Jones, *Machine learning for structured clinical data* 2018.
9. E. Choi, M. Taha Bahadori, A. Schuetz and W. F. Stewart, *Doctor AI: Predicting Clinical Events via Recurrent Neural Networks*, tech. rep.
10. Z. C. Lipton, D. C. Kale, C. Elkan and R. Wetzel (11 2015).
11. A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, P. Sundberg, H. Yee, K. Zhang, Y. Zhang, G. Flores, G. E. Duggan, J. Irvine, Q. Le, K. Litsch, A. Mossin, J. Tansuwan, D. Wang, J. Wexler, J. Wilson, D. Ludwig, S. L. Volchenboum,

K. Chou, M. Pearson, S. Madabushi, N. H. Shah, A. J. Butte, M. D. Howell, C. Cui, G. S. Corrado and J. Dean, *npj Digital Medicine* **1**, p. 18 (12 2018).
12. J. Pennington, R. Socher, C. M. P. o. t. 2014 and u. 2014, *aclweb.orgSign in* .
13. M. Nickel and D. Kiela, *Poincaré Embeddings for Learning Hierarchical Representations*, tech. rep.
14. M. Gromov., Hyperbolic groups., in *Essays in group theory, Springer.*, 1987 p. pages 75–263.
15. S. Bonnabel, *Stochastic gradient descent on Riemannian manifolds*, tech. rep.
16. A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston and O. Yakhnenko, *Translating Embeddings for Modeling Multi-relational Data*, tech. rep.
17. D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat and M. Boguna (6 2010).
18. B. K. B. Beaulieu-Jones and C. S. Greene, *Journal of Biomedical Informatics* **64**, 168 (2016).
19. M. M. Salem, *Seminars in nephrology* **22**, 17 (1 2002).
20. S. Smith, E. H. Hay, N. Farhat and R. Rekaya, *BMC genetics* **14**, p. 124 (12 2013).
21. S. Buyske, G. Yang, T. C. Matise and D. Gordon, *Human Heredity* **67**, 287 (2009).
22. R. Rekaya, S. Smith, E. H. Hay, N. Farhat and S. E. Aggrey, *The application of clinical genetics* **9**, 169 (2016).
23. A. Verma, A. Lucas, S. S. Verma, Y. Zhang, N. Josyula, A. Khan, D. N. Hartzel, D. R. Lavage, J. Leader, M. D. Ritchie and S. A. Pendergrass, *American journal of human genetics* **102**, 592 (4 2018).
24. J. C. Denny, M. D. Ritchie, M. A. Basford, J. M. Pulley, L. Bastarache, K. Brown-Gentry, D. Wang, D. R. Masys, D. M. Roden and D. C. Crawford, *Bioinformatics* **26**, 1205 (2010).
25. S. A. Pendergrass, K. Brown-Gentry, S. Dudek, A. Frase, E. S. Torstenson, R. Goodloe, J. L. Ambite, C. L. Avery, S. Buyske, P. Bůžková, E. Deelman, M. D. Fesinmeyer, C. A. Haiman, G. Heiss, L. A. Hindorff, C. N. Hsu, R. D. Jackson, C. Kooperberg, L. Le Marchand, Y. Lin, T. C. Matise, K. R. Monroe, L. Moreland, S. L. Park, A. Reiner, R. Wallace, L. R. Wilkens, D. C. Crawford and M. D. Ritchie, *PLoS Genetics* **9** (2013).
26. A. Okifuji and B. D. Hare, *Journal of pain research* **8**, 399 (2015).
27. D. S. McVinnie, *British journal of pain* **7**, 163 (11 2013).
28. M. J. Young, A. J. M. Boulton, A. F. Macleod, D. R. R. Williams and P. H. Sonksen, *Diabetologia* **36**, 150 (2 1993).
29. A. L. Beam and I. S. Kohane, *JAMA* **319**, p. 1317 (4 2018).
30. M. Ghassemi, T. Naumann, P. Schulam, A. L. Beam and R. Ranganath (6 2018).
31. D. Hassan, U. Aickelin and C. Wagner, *Comparison of Distance Metrics for Hierarchical Data in Medical Databases*, tech. rep.

The Effectiveness of Multitask Learning for Phenotyping with Electronic Health Records Data

Daisy Yi Ding[1], Chloé Simpson[1], Stephen Pfohl[1], Dave C. Kale[2], Kenneth Jung[1], Nigam H. Shah[1]
[1]*Stanford Center for Biomedical Informatics Research, Stanford University, Stanford, CA;*
[2]*USC Information Sciences Institute, University of Southern California, Marina del Rey, CA*

Electronic phenotyping is the task of ascertaining whether an individual has a medical condition of interest by analyzing their medical record and is foundational in clinical informatics. Increasingly, electronic phenotyping is performed via supervised learning. We investigate the effectiveness of multitask learning for phenotyping using electronic health records (EHR) data. Multitask learning aims to improve model performance on a target task by jointly learning additional auxiliary tasks and has been used in disparate areas of machine learning. However, its utility when applied to EHR data has not been established, and prior work suggests that its benefits are inconsistent. We present experiments that elucidate when multitask learning with neural nets improves performance for phenotyping using EHR data relative to neural nets trained for a single phenotype and to well-tuned baselines. We find that multitask neural nets consistently outperform single-task neural nets for rare phenotypes but underperform for relatively more common phenotypes. The effect size increases as more auxiliary tasks are added. Moreover, multitask learning reduces the sensitivity of neural nets to hyperparameter settings for rare phenotypes. Last, we quantify phenotype complexity and find that neural nets trained with or without multitask learning do not improve on simple baselines unless the phenotypes are sufficiently complex.

Keywords: Electronic Health Records; Electronic phenotyping algorithms; Deep learning; Multi-task learning.

1. Introduction

The goal of electronic phenotyping is to identify patients with (or without) a specific disease or medical condition using their electronic medical records. Identifying sets of such patients (i.e. a patient cohort) is the first step in a wide range of applications such as comparative effectiveness studies,[1,2] clinical decision support,[3,4] and translational research.[5] Increasingly, such phenotyping is done via supervised machine learning methods.[6–8]

Multitask learning (MTL) is a widely used technique in machine learning that seeks to improve performance on a *target task* by jointly modeling the target task and additional *auxiliary tasks*.[9] MTL has been used to good effect in a wide variety of domains including computer vision,[10] natural language processing,[11,12] speech recognition,[13] and even drug development.[14,15] However, its effectiveness using EHR data is less well established, with prior work providing contradictory evidence regarding its utility.[16,17]

In this work, we investigate the effectiveness of MTL for phenotyping using EHR. Our pre-

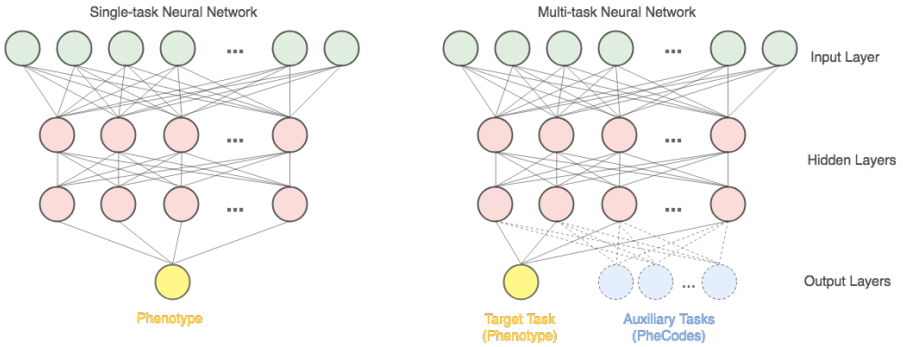


Fig. 1. The architecture of a multitask neural net for electronic phenotyping is shown on the right: the target task (shown in yellow) and the auxiliary tasks (shown in blue) share hidden layers and have distinct output layers; for comparison, we show the corresponding single-task neural net on the left with a single output layer for the target phenotype.

liminary studies recapitulated the inconsistent benefits found in prior work.[16,17] We thus aimed to elucidate the properties of the phenotypes for which MTL helps versus harms performance.

In this paper, we present a systematic exploration of the factors that determine whether or not MTL improves the performance of neural nets for phenotyping with EHR data. Our experiments suggest the following conclusions:

- MTL helps performance for low prevalence (i.e. rare) phenotypes, but harms performance for relatively high prevalence phenotypes. Consistent with some prior work, there is a dose-response relationship with the number of auxiliary tasks, with the magnitude of the benefit or harm generally increasing as auxiliary tasks are added.
- MTL reduces the sensitivity of neural nets to hyperparameter settings. This is of practical importance when one has a limited computational budget for model development.
- Neural nets trained with or without MTL do not improve on simple baselines unless phenotypes are sufficiently complex. However, learning more complex models can be problematic with complex but low prevalence phenotypes. We explore this phenomenon by quantifying phenotype complexity using information theoretic metrics.

2. Background

2.1. *Multitask nets*

Multitask Learning MTL seeks to improve performance on a given target task by jointly learning additional auxiliary tasks. For instance, if the target task is whether or not a patient has type 2 diabetes, one might jointly learn auxiliary tasks such as whether or not the patient has other diseases such as congestive heart failure or emphysema. MTL is most frequently embodied as a neural net in which the earliest layers of the network are shared among the target and auxiliary tasks, with separate outputs for each task (see Figure 1). MTL was originally proposed to improve performance on risk stratification of pneumonia patients by leveraging information in lab values as auxiliary tasks.[9] It has since been used extensively for health care problems such as predicting illness severity[18] and mortality,[17] and disease risk and progression.[19–23] However, the reported benefits of MTL are inconsistent across problems.

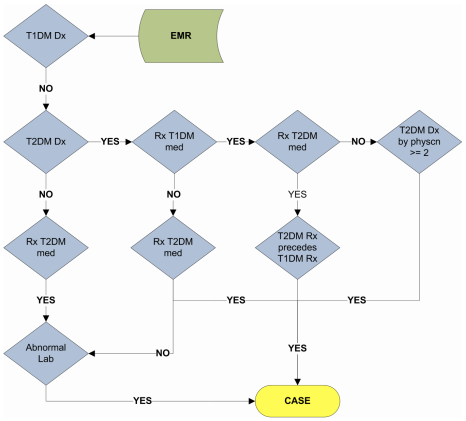


Fig. 2. Rule-based definitions for *Type 2 Diabetes Mellitus* from PheKB.[34]

For example, Che et. al showed that MTL improved performance on identifying physiological markers in clinical time series data,[16] while Nori et. al concluded that MTL failed to improve performance on predicting mortality in an acute care setting.[17] Our aim in this study is to clarify when one might expect MTL to help performance on problems using EHR data. We focus specifically on the foundational problem of phenotyping, which we discuss next.

Electronic Phenotyping In this study, *phenotyping* is simply identifying whether or not a patient has a given disease or disorder. The gold standard for phenotyping remains manual chart review by trained clinicians, which is time-consuming and expensive.[24–26]

This has spurred work on *electronic phenotyping*, which aims to solve the same problem using automated means and EHR data as input. The earliest electronic phenotyping algorithms were rule-based decision criteria created by domain experts.[24–28] Figure 2 shows an example of a rule-based algorithm for type 2 diabetes mellitus. In this approach, identifying patients with the phenotype can be automated once the algorithm is specified, but the latter process is still time consuming and expensive.

More recent work has focused on using statistical learning[6,29–33] to automate the process of specifying the algorithm itself using the methods of machine learning (i.e. models such as logistic regression, random forests, and neural nets). MTL is a particular method for doing this better. Our goal in this work is not to maximize performance for some phenotype but rather to gain insight into when MTL helps versus harms in this approach to phenotyping.

3. Methods

3.1. *Dataset Construction and Design*

Dataset Our data comprises de-identified patient data spanning 2010 through 2016 for 1,221,401 patients from the Stanford Translational Research Integrated Database Environment (STRIDE) database.[35] Each patient's data includes timestamped diagnosis (ICD-9), procedure (CPT), drug (RxNorm) codes, along with demographic information (age, gender, race, and ethnicity). We use a simple multi-hot feature representation whereby each ICD-9, CPT, and RxNorm code is mapped to a binary indicator variable for whether the code occurs in the patient's medical history. We similarly encode gender, race, ethnicity, and each integer

value of age. This process results in a sparse representation of 29,102 features.

Target Task Phenotypes Phenotyping with statistical classifiers is typically framed as a binary classification task, which requires data labeled with whether or not the patient has the phenotype. For this study, we derive the phenotypes using rule-based definitions from PheKB,[36] a compendium of phenotype definitions developed to support genome-wide association studies. We focus on 4 phenotypes, chosen to span a range of prevalences. They are: type 2 diabetes mellitus (T2DM), atrial fibrillation (AF), abdominal aneurysm (AA), and angioedema (AE). The respective prevalences of these phenotypes in our data are 2.95%[a], 2.89%, 0.12%, and 0.08%. We use these rule-based definitions to derive the phenotypes because they are easy to implement, scalable and transparent – later we describe how we take advantage of the rule-based definitions to gain insight into the effectiveness of MTL relative to baselines.

Auxiliary Tasks Our auxiliary tasks are to classify *phecodes*, manually curated groupings of ICD-9 codes originally used to facilitate phenome-wide association studies.[37] We randomly select phecodes with prevalence between 0.08% and 2.95%, i.e. the lowest and the highest target phenotype prevalences, as auxiliary tasks. We conduct binary classification on each phecode and experiment with 5, 10, and 20 randomly selected phecodes as auxiliary tasks.

3.2. *Experimental Design*

We aim to investigate whether and under what circumstances MTL improves performance upon baselines. Recent work suggests that we need to be careful in order to draw robust conclusions on the relative merits of machine learning, especially neural net based methods.[38–41]

First, one typically randomly partitions data into training, validation and test sets. We fit models to the training set, select or tune models using the validation set, and estimate performance on new data using the test set. All three steps use finite samples and are thus subject to noise due to sampling. This is especially true when data exhibit extreme class imbalance, as is the case with our phenotypes. Second, the performance of even simple feed-forward neural nets is known to be sensitive to hyperparameters such as the number of hidden layers and their sizes. Finally, fitting neural nets is inherently stochastic due to random initialization of model parameters and training by some variation of stochastic gradient descent. This, combined with the highly non-convex nature of neural nets, implies that different training runs of a neural net with fixed hyperparameters and dataset splits can still result in widely varying performance.[42]

We thus designed our experiments to mitigate noise due to these factors. First, for each phenotype, we perform ten random splits of the data into training (80%), validation (10%), and test sets (10%). We use stratified sampling to fix the prevalence of the targets to the overall sample prevalence in each of the training, validation and test sets. Second, for each of these splits, we perform a grid search over these hyperparameters for the MTNN and STNN models: we vary the number of hidden layers (1 or 2), their size (128, 256, 512, 1024, and 2048), and the initial learning rate for the algorithm (1e-4 and 5e-5). Moreover, we performed experiments

[a]The prevalence is low compared to the population prevalence of approximately 9% because the rule-based definitions from PheKB are tuned for high precision at the cost of lower recall.

varying the number of auxiliary tasks (in the form of 5, 10, and 20 nested, randomly selected phecodes) for MTNNs by conducting the above grid search for each scenario. For each split, we also fit an L1 regularized logistic regression model, tuned on the validation set. We use the area under the Precision-Recall curve (AUPRC) as our evaluation metric since it can be more informative than the area under the receiver operator characteristic curve (AUROC) in problems with extreme class imbalance.[43]

Phenotype Complexity Our experiments suggested that the complexity of the phenotype is important in whether MTNNs and STNNs outperform well-tuned logistic regression. We quantified the phenotype complexity with regard to a subset of the features upon which the classifiers are built[b]. If we had access to an oracle that told us which features of the patient representation are important in determining a patient's phenotype, we could characterize the complexity of the phenotype with regard to the observed combinations of these features in the positive cases. We could also compare the distributions of the positive and negative cases to examine how difficult it is to discriminate positive and negative cases given the relevant features.

Our phenotypes are derived from the rule-based definitions, which we use as such an oracle: for each phenotype, we extract the features involved in its rule-based definitions (the *oracle features*) and count occurrences of each distinct combination of these features observed in the positive and negative cases. Each unique combination is represented as a binary string with each digit indicating the presence or absence of an oracle feature. Since some of the phenotype definitions involve very many combinations, we hash the combinations into a lower-dimensional space, i.e. a fixed number buckets. Specifically, we use a hash function to map the combinations (the variable-length binary strings) to a fixed number of hash codes (the buckets). We obtain the counts in each bucket for the positive and negative cases and analyze the resulting histograms using two information theoretic metrics.

Let $\mathbf{x}_i$ be the vector of oracle features for bucket i. We summarize the phenotype complexity of positive cases by treating the histogram as a discrete probability distribution and calculate its information entropy,[44] defined as:

$$H(\mathbf{X}) = \mathbb{E}_{\mathbf{x}\sim P}\left[\log(\mathbf{x})\right] = \sum_{i=1}^{n} p(\mathbf{x}_i)\log(\mathbf{x}_i),$$

where n is the number of buckets. This metric summarizes the diversity of positive cases with respect to the oracle features and is higher for more complex phenotypes.

We compare the distributions of the positive and negative cases using the Kullback-Leibler (KL) divergence.[45] For discrete probability distributions P^+ and P^-, the KL divergence from P^- to P^+ is defined as:

$$D_{KL}(P^+ \parallel P^-) = \sum_{i=1}^{n} P^+(\mathbf{x}_i)\frac{P^-(\mathbf{x}_i)}{P^+(\mathbf{x}_i)},$$

where n is the number of buckets[c]. $P^+(\mathbf{x}_i)$ and $P^-(\mathbf{x}_i)$ are the normalized frequencies of bucket

[b]There is no direct way to quantify the complexity of the rule-based definitions shown in Figure 2.
[c]KL divergence does not admit zero probabilities so we use Laplace smoothing on the distributions to deal with combinations that do not have mutual support.

i for cases and controls respectively. KL divergence measures the dissimilarity between the case and control distributions and is lower for the phenotypes that are harder to discriminate.[d]

Neural Net Details All neural nets used ReLU activations[46] for the hidden layers and Xavier initialization[47] and were trained using Adam[48] with standard parameters ($\beta_1 = 0.9$ and $\beta_2 = 0.99$) for 6 epochs[e]. We controlled overfitting with batch normalization and early stopping on the validation set.

4. Experiments and Results

In this section, we present results that provide insight into the following questions:

- When does MTL improve performance relative to single-task models for phenotyping?
- How do the effects of MTL change with the number of phecodes as auxiliary tasks?
- How do the neural net methods compare with strong baseline methods, and what are the characteristics of the tasks for which they provide some benefit?

4.1. *When Does Multitask Learning Improve Performance?*

We investigate the performance of MTNNs over a range of hyperparameter settings and over multiple random splits of the data. MTNN performance is compared to the performance of STNNs over the same hyperparameter settings and data splits. Figure 3 shows the optimal MTNN and STNN performance achieved on each split for the four phenotypes. We find that MTNNs consistently outperform STNNs for the low prevalence phenotypes, i.e. angioedema and abdominal aneurysm. In contrast, MTL harms performance for the relatively high-prevalence phenotypes, i.e. T2DM and atrial fibrillation. The left plot in Figure 4 shows the pairwise differences between MTNN and STNN optimal performance across the splits.

Moreover, the performance of STNNs is very sensitive to hyperparameter settings for the low prevalence phenotypes, as illustrated by the large spread in AUPRC values (see Figure 3). In contrast, MTNNs are more robust to hyperparameter settings for these phenotypes. In practice, tuning neural nets is time-consuming and finding an ideal model demands extensive computation. MTL may increase our chance of finding a reasonable model, which is of practical value when one has a limited computational budget on model space exploration.

4.2. *Relationship Between Performance and Number of Tasks*

We investigate how MTL is influenced by the number of auxiliary tasks as defined in the form of phecodes. We trained MTNNs with nested sets of 5, 10, and 20 randomly selected phecodes (i.e. the 5-phecode set is a subset of the 10-phecode set, and so on), and reported the performance with the optimal hyperparameter setting for each split. The right plot in Figure 4 shows pairwise differences in AUPRC values between MTNNs and STNNs. For the low prevalence phenotypes, more phecodes increases performance gains. Similarly, more phecodes

[d]Please refer to https://arxiv.org/abs/1808.03331 for a more detailed description of our method.
[e]We found 6 epochs was sufficient for all models to converge.

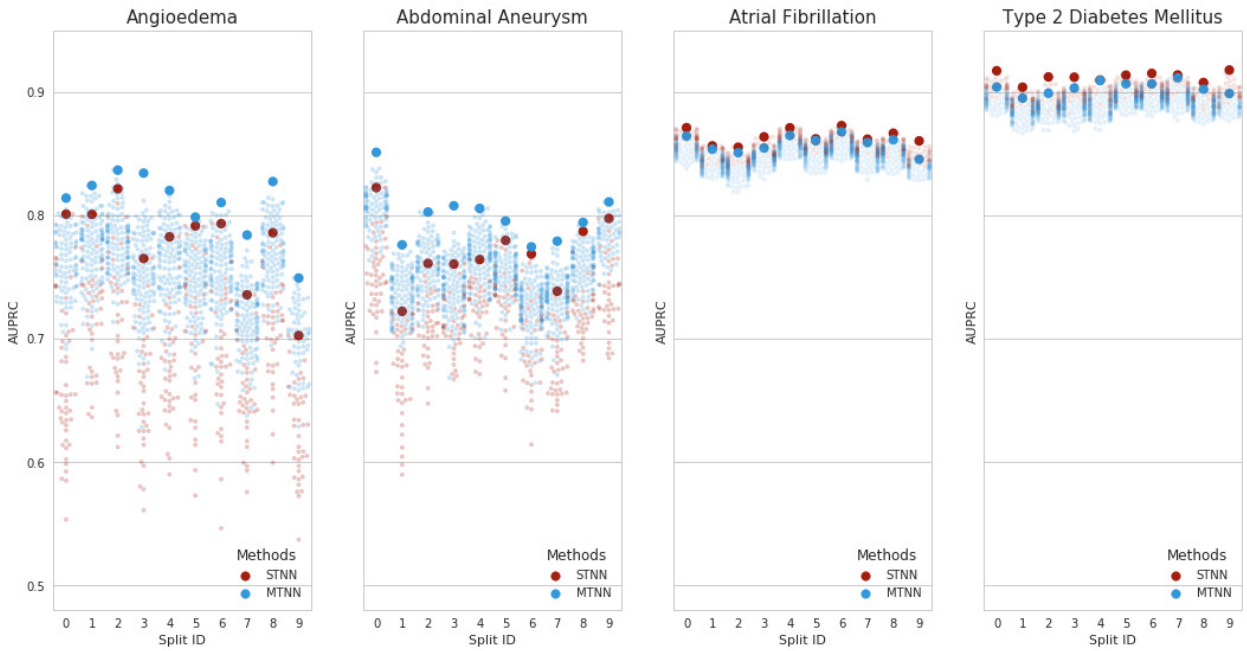


Fig. 3. MTNN and STNN performance for Angioedema, Abdominal Aneurysm, Atrial Fibrillation, and Type 2 Diabetes Mellitus with various hyperparameter settings across the ten splits; the best case MTNN and STNN performance is emphasized by the solid dots: the blue and red dots correspond to MTNNs and STNNs respectively.

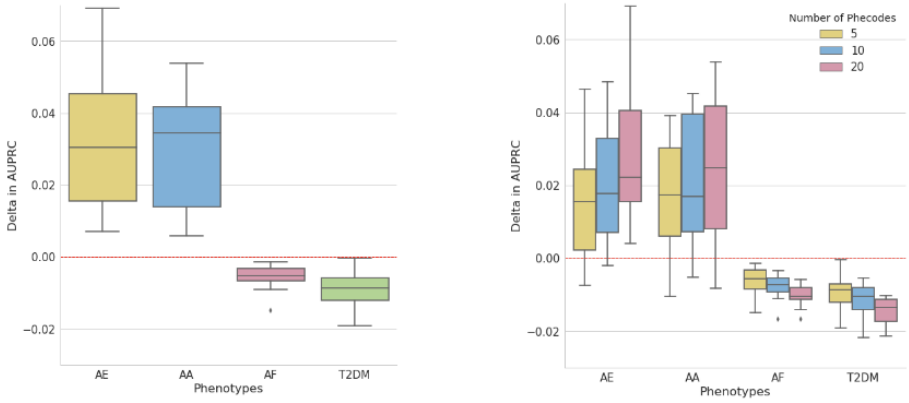


Fig. 4. The left plot shows the pairwise differences in AUPRC values of the optimal MTNNs and STNNs for Angioedema, Abdominal Aneurysm, Atrial Fibrillation, and Type 2 Diabetes Mellitus across the ten splits. The right plot shows the pairwise differences in AUPRC values of the optimal STNNs and MTNNs with different number of phecodes as auxillary tasks.

for high prevalence phenotypes leads to more severe negative effects, though the scale of the negative effects is smaller than the positive effects for low prevalence phenotypes[f].

[f]This dose-response relationship with the number of auxiliary tasks recapitulates the findings of Ramsundar et al,[14] but we find the relationship holds for both the benefit and harm of MTL.

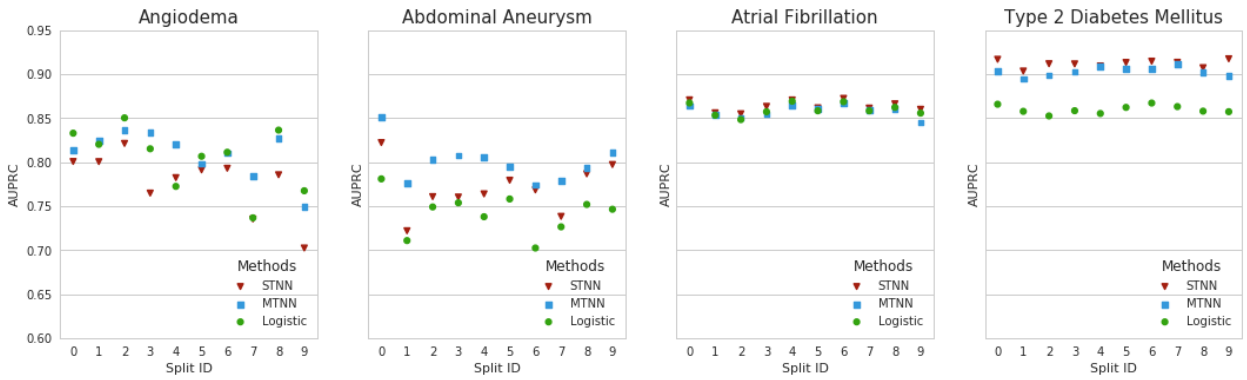


Fig. 5. MTNN, STNN, and LR optimal performance for Angioedema, Abdominal Aneurysm, Atrial Fibrillation, and Type 2 Diabetes Mellitus across splits: the blue squares, the red triangles, and the green dots correspond to MTNN, STNN, and LR respectively.

4.3. *Comparison with Logistic Regression Baseline*

In discussing the merits of MTL, it is important to also compare the performance against simpler baseline methods in addition to single-task neural nets. We compare the performance of the neural nets with L1 regularized logistic regression (LR), a consistently strong baseline for EHR data[49,50] (see Figure 5). LR is consistently outperformed by the neural nets for abdominal aneurysm and type 2 diabetes mellitus, which are low and high prevalence respectively. For angioedema, a low prevalence phenotype, performance relative to LR is inconsistent across the splits, although MTNNs consistently beat STNNs. And for atrial fibrillation, a high prevalence phenotype, MTNNs and STNNs provide little or no benefit over LR. Prevalence alone is insufficient to account for the relative performance between both MTNN and STNN and LR.

4.4. *Interaction between Phenotype Prevalence and Complexity*

Our comparison of MTNNs and STNNs versus LR suggests that phenotype prevalence alone cannot explain when neural nets outperform simpler linear models. We hypothesized that phenotype complexity also plays a role since neural nets with or without MTL can automatically model non-linearities and interactions, while LR must have non-linearities and interactions explicitly encoded in features. We leveraged the rule-based phenotype definitions to explore this hypothesis and found evidence of an interaction between phenotype prevalence and complexity.

Phenotype Complexity For each phenotype, we generated histograms of the observed combinations of the oracle features for the positive and negative cases (see Figure 6) and calculated the information entropy of the positive cases and the KL divergence between the positive and negative cases (see Table 1) as described in Methods 3.2.

We find that atrial fibrillation, a high-prevalence phenotype, has low entropy and high KL divergence. With respect to the oracle features, all the positive cases are similar to each other, while the positive and negative cases are very dissimilar to each other. A relatively simple model should be able to capture this, explaining the observation that LR achieves comparable

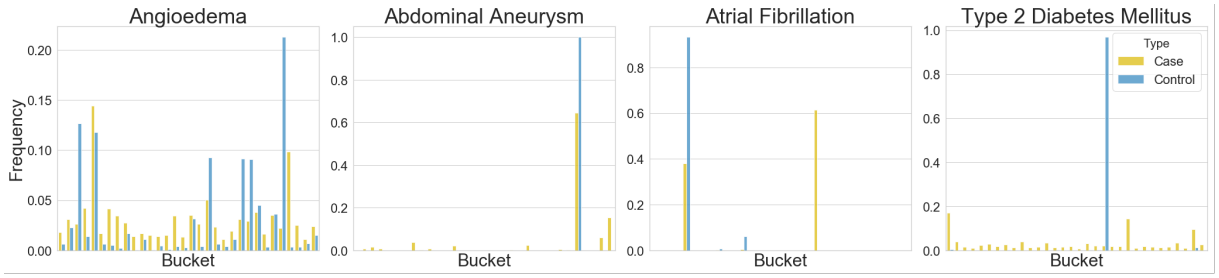


Fig. 6. Distributions of the combinations of the oracle features involved in the rule-based definitions for Angioedema, Abdominal Aneurysm, Atrial Fibrillation, and Type 2 Diabetes Mellitus. The yellow and blue bars correspond to the positive and negative cases respectively. The x-axes represent the buckets of unique combinations of the oracle features: in our study, we use 32 buckets. Note that the choice of 32 buckets was arbitrary and not tuned in any way.

Table 1. Phenotype Complexity

Phenotype	Prevalence	Entropy	KL Divergence
Angioedema	0.08 %	3.233	0.930
Abdominal Aneurysm	0.12%	1.396	2.414
Atrial Fibrillation	2.89%	0.709	5.383
Type 2 Diabetes Mellitus	2.95 %	3.012	3.806

performance to MTNNs and STNNs for this phenotype.

Abdominal aneurysm, a low prevalence phenotype, and T2DM, a high prevalence phenotype, have higher information entropy and lower KL divergence values than atrial fibrillation. Thus, the positive cases are more diverse and discrimination is more difficult than atrial fibrillation with respect to each phenotype's oracle features. For these phenotypes, both MTNNs and STNNs outperform LR – we benefit from more expressive models. However, whether MTNNs beat STNNs depends on prevalence.

Finally, angioedema has the highest entropy and lowest KL divergence – it is both the most complex and hardest to discriminate of the four phenotypes. Complex phenotypes should benefit from more expressive models. However, we observe that while MTNNs consistently outperform STNNs, their performance relative to LR is inconsistent across splits. One possible explanation for this behavior is that relative performance is sensitive to the assignment of patients to training, validation and test sets: with such diverse cases and common support with respect to the oracle features, it is much more likely for the test set to contain patients unlike any seen in the training set.

5. Limitations

We have set out to investigate MTL and its effectiveness for electronic phenotyping. However, our work has important limitations. First, we randomly select phecodes for auxiliary tasks, but it has been argued that auxiliary tasks should be directly related to the target task.[51] It is possible that better auxiliary tasks would improve the benefit of MTL. Specifically, more related phecodes might mitigate or eliminate the performance degradation observed for the

high-prevalence phenotypes or inconsistent relative performance between MTNN and LR for angioedema. However, the notion of task relatedness is underspecified so it is problematic to compute in order to select auxiliary tasks. Indeed, in preliminary work we explored various formulations of relatedness to select auxiliary tasks but found that none performed better than random selection. One could ask domain experts to manually construct or pick auxiliary tasks for specific phenotypes, but this is beyond the scope of this work. Moreover, it has also been shown that the task relatedness is unnecessary for MTL to provide benefits.[52] However, we acknowledge that it is an interesting line of inquiry for future work to further explore how to improve multitask learning for electronic phenotyping. Second, to address the unavailability of large-scale ground truth phenotypes, we use rule-based definitions because they are transparent and available, but we recognize that the phenomenon we observe may be artifacts of the rule-based definitions. We also acknowledge the possibility that the observed phenomenon might not generalize to other phenotypes; we focused on four phenotypes to conduct an in-depth examination, sacrificing breadth. Finally, the rule-based phenotype definitions contain predicates encoding temporal relationships, e.g., a drug code followed by a diagnosis code. Our simple multi-hot feature representation does not encode temporal information. As a result, there is an upper bound on the performance of any statistical classifier using this feature representation.

6. Conclusion

We have investigated the effectiveness of multitask learning on electronic phenotyping with EHR data, aiming to elucidate the properties of situations for which MTL improves or harms performance. We trained multitask neural networks to classify a target phenotype jointly with auxiliary tasks drawn from phecodes. We found that MTL provided consistent performance improvements over single-task neural networks on extremely rare phenotypes. However, for relatively higher prevalence phenotypes, MTL actually reduced performance. In both cases, the effect scaled with the number of auxiliary tasks as defined in the form of phecodes. Moreover, we found that MTL improved the robustness of neural networks to hyperparameter settings for the extremely rare phenotypes, which is of practical value in situations when one has a limited computational budget for model exploration. Finally, we analyzed phenotype complexity to shed light on the relative performance of both MTNN and STNN versus well-tuned L1 regularized logistic regression baselines and found evidence of an interaction between phenotype prevalence and complexity. We showed that simple linear models are sufficient for non-complex phenotyping tasks. More expressive models can substantially improve performance for more complex phenotypes, but only if the data support learning them well, which may be problematic for rare phenotypes.

Acknowledgments

This work was supported by NLM R01-LM011369-05 and a grant supporting the Observational Health Data Science and Informatics (OHDSI) by Janssen Research and Development LLC. Internal funding by the School of Medicine at Stanford also supported part of this work. We gratefully acknowledge Jason Fries for many helpful discussions about this work.

References

1. D. C. Crawford, D. R. Crosslin, G. Tromp, I. J. Kullo, H. Kuivaniemi, M. G. Hayes, J. C. Denny, W. S. Bush, J. L. Haines, D. M. Roden *et al.*, *Frontiers in Genetics* **5**, p. 184 (2014).
2. F. J. Manion, M. R. Harris, A. G. Buyuktur, P. M. Clark, L. C. An and D. A. Hanauer, *Current Oncology Reports* **14**, 494 (2012).
3. C. A. Longhurst, R. A. Harrington and N. H. Shah, *Health Affairs* **33**, 1229 (2014).
4. W.-Q. Wei and J. C. Denny, *Genome Medicine* **7**, p. 41 (2015).
5. N. H. Shah, *Nature Biotechnology* **31**, p. 1095 (2013).
6. V. Agarwal, T. Podchiyska, J. M. Banda, V. Goel, T. I. Leung, E. P. Minty, T. E. Sweeney, E. Gyang and N. H. Shah, *Journal of the American Medical Informatics Association* **23**, 1166 (2016).
7. Y. Halpern, S. Horng and D. Sontag, *Proceedings of the 1st Machine Learning in Health Care (MLHC)* , p. 209 (2016).
8. J. Banda, Y. Halpern, D. Sontag and N. Shah, *AMIA Summits on Translational Science Proceedings* , p. 48 (2017).
9. R. Caruana, S. Baluja and T. M. Mitchell, *Advances in Neural Information Processing Systems* , 959 (1995).
10. R. Girshick, J. Donahue, T. Darrell and J. Malik, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* , 580 (2014).
11. B. Plank, A. Søgaard and Y. Goldberg, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (2016).
12. P. Liu, X. Qiu and X. Huang, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* , 1 (2017).
13. S. Toshniwal, H. Tang, L. Lu and K. Livescu, *18th Annual Conference of the International Speech Communication Association* , 3532 (2017).
14. B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding and V. Pande, *arXiv preprint arXiv:1502.02072* (2015).
15. P. Zhang, F. Wang and J. Hu, *AMIA Annual Symposium Proceedings* **2014**, p. 1258 (2014).
16. Z. Che, D. Kale, W. Li, M. T. Bahadori and Y. Liu, *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* , 507 (2015).
17. N. Nori, H. Kashima, K. Yamashita, H. Ikai and Y. Imanaka, *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* , 855 (2015).
18. M. Ghassemi, M. A. F. Pimentel, T. Naumann, T. Brennan, D. A. Clifton, P. Szolovits and M. Feng, *Proceedings of the 29th Conference on Artificial Intelligence* , 446 (2015).
19. C. Ngufor, S. Upadhyaya, D. Murphree, D. Kor and J. Pathak, *IEEE International Conference on Data Science and Advanced Analytics* , 1 (2015).
20. X. Wang, F. Wang, J. Hu and R. Sorrentino, *AMIA Annual Symposium Proceedings* **2014**, p. 1180 (2014).
21. N. Razavian, J. Marcus and D. Sontag, *Machine Learning for Healthcare Conference* , 73 (2016).
22. J. Zhou, L. Yuan, J. Liu and J. Ye, *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* , 814 (2011).
23. Z. C. Lipton, D. C. Kale, C. Elkan and R. Wetzel, *arXiv preprint arXiv:1511.03677* (2015).
24. K. M. Newton, P. L. Peissig, A. N. Kho, S. J. Bielinski, R. L. Berg, V. Choudhary, M. Basford, C. G. Chute, I. J. Kullo, R. Li *et al.*, *Journal of the American Medical Informatics Association* **20**, e147 (2013).
25. C. L. Overby, J. Pathak, O. Gottesman, K. Haerian, A. Perotte, S. Murphy, K. Bruce, S. Johnson, J. Talwalkar, Y. Shen *et al.*, *Journal of the American Medical Informatics Association* **20**, e243 (2013).

26. H. Mo, W. K. Thompson, L. V. Rasmussen, J. A. Pacheco, G. Jiang, R. Kiefer, Q. Zhu, J. Xu, E. Montague, D. S. Carrell *et al.*, *Journal of the American Medical Informatics Association* **22**, 1220 (2015).
27. A. N. Kho, J. A. Pacheco, P. L. Peissig, L. Rasmussen, K. M. Newton, N. Weston, P. K. Crane, J. Pathak, C. G. Chute, S. J. Bielinski *et al.*, *Science Translational Medicine* **3**, 79re1 (2011).
28. M. Conway, R. L. Berg, D. Carrell, J. C. Denny, A. N. Kho, I. J. Kullo, J. G. Linneman, J. A. Pacheco, P. Peissig, L. Rasmussen *et al.*, *AMIA Annual Symposium Proceedings* **2011**, p. 274 (2011).
29. Y. Huang, P. McCullagh, N. Black and R. Harper, *Artificial Intelligence in Medicine* **41**, 251 (2007).
30. Y. Chen, J. Ghosh, C. A. Bejan, C. A. Gunter, S. Gupta, A. Kho, D. Liebovitz, J. Sun, J. Denny and B. Malin, *Journal of Biomedical Informatics* **55**, 82 (2015).
31. J. Zhou, F. Wang, J. Hu and J. Ye, *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* , 135 (2014).
32. J. C. Ho, J. Ghosh and J. Sun, *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* , 115 (2014).
33. Y. Halpern, S. Horng, Y. Choi and D. Sontag, *Journal of the American Medical Informatics Association* **23**, 731 (2016).
34. Type 2 diabetes mellitus. `https://phekb.org/phenotype/18`, Accessed: 2018-07-23.
35. H. J. Lowe, T. A. Ferris, P. M. Hernandez and S. C. Weber, *American Medical Informatics Association Annual Symposium* (2009).
36. J. C. Kirby, P. Speltz, L. V. Rasmussen, M. Basford, O. Gottesman, P. L. Peissig, J. A. Pacheco, G. Tromp, J. Pathak, D. S. Carrell *et al.*, *Journal of the American Medical Informatics Association* **23**, 1046 (2016).
37. W.-Q. Wei, L. A. Bastarache, R. J. Carroll, J. E. Marlo, T. J. Osterman, E. R. Gamazon, N. J. Cox, D. M. Roden and J. C. Denny, *PloS One* **12**, p. e0175508 (2017).
38. Y. Li, N. Du and S. Bengio, *arXiv preprint arXiv:1708.00065* (2017).
39. G. Melis, C. Dyer and P. Blunsom, *arXiv preprint arXiv:1707.05589* (2017).
40. M. Lucic, K. Kurach, M. Michalski, S. Gelly and O. Bousquet, *arXiv preprint arXiv:1711.10337* (2017).
41. A. Oliver, A. Odena, C. Raffel, E. D. Cubuk and I. J. Goodfellow, *arXiv preprint arXiv:1804.09170* , 1 (2018).
42. N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy and P. T. P. Tang, *arXiv preprint arXiv:1609.04836* (2016).
43. T. Saito and M. Rehmsmeier, *PloS One* **10**, p. e0118432 (2015).
44. C. E. Shannon, *ACM SIGMOBILE Mobile Computing and Communications Review* **5**, 3 (2001).
45. S. Kullback and R. A. Leibler, *The Annals of Mathematical Statistics* **22**, 79 (1951).
46. V. Nair and G. E. Hinton, 807 (2010).
47. X. Glorot and Y. Bengio, *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics* , 249 (2010).
48. D. P. Kingma and J. Ba, *arXiv preprint arXiv:1412.6980* (2014).
49. A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun *et al.*, *Digital Medicine* **1**, p. 18 (2018).
50. N. Razavian, S. Blecker, A. M. Schmidt, A. Smith-McLallen, S. Nigam and D. Sontag, *Big Data* **3**, 277 (2015).
51. R. Caruana, *Machine learning* **28**, 41 (1997).
52. B. Romera-Paredes, A. Argyriou, N. Berthouze and M. Pontil, *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics* , 951 (2012).

ODAL: A one-shot distributed algorithm to perform logistic regressions on electronic health records data from multiple clinical sites*

Rui Duan[#], Mary Regina Boland[#], Jason H. Moore and Yong Chen[†]
Department of Biostatistics, Epidemiology & Informatics, University of Pennsylvania,
423 Guardian Drive, PA, 19104, USA
[†] *Corresponding Email: ychen123@upenn.edu*
[#] *Co-first author*

Electronic Health Records (EHR) contain extensive information on various health outcomes and risk factors, and therefore have been broadly used in healthcare research. Integrating EHR data from multiple clinical sites can accelerate knowledge discovery and risk prediction by providing a larger sample size in a more general population which potentially reduces clinical bias and improves estimation and prediction accuracy. To overcome the barrier of patient-level data sharing, distributed algorithms are developed to conduct statistical analyses across multiple sites through sharing only aggregated information. The current distributed algorithm often requires iterative information evaluation and transferring across sites, which can potentially lead to a high communication cost in practical settings. In this study, we propose a privacy-preserving and communication-efficient distributed algorithm for logistic regression without requiring iterative communications across sites. Our simulation study showed our algorithm reached comparative accuracy comparing to the oracle estimator where data are pooled together. We applied our algorithm to an EHR data from the University of Pennsylvania health system to evaluate the risks of fetal loss due to various medication exposures.

Keywords: birth outcomes; distributed computing; meta-analysis; multi-site analysis; pregnancy; prenatal; surrogate likelihood.

1. Introduction

1.1. *Integrate evidence from multiple clinical sites*

Electronic Health Records (EHR) contain information collected routinely as a part of clinical care. These data include diagnoses, medications, procedures, imaging and clinical notes. Since 2009, the use of EHR has grown tremendously across the nation. This allows for meaningful use of data recorded there [1, 2]. Institutional data integration is a major trend in EHR-based research [3, 4]. Integrating data from different institutions or clinical sites allows us to obtain more meaningful sample size and potentially accelerates knowledge discoveries in a more general population. In

* This work is supported in part by the University of Pennsylvania, and National Institutes of Health grants AI116794, DK112217, ES013508, HL134015, LM010098, LM011360, LM012601, TR001263, 1R01LM012607, 1R01AI130460, and the Commonwealth Universal Research Enhancement Program grant from the Pennsylvania Department of Health.

particular, for studying relatively rare events or conditions, such as complications from invasive procedures, adverse events associated with new medications, association of disease with a rare gene variant, and many others, integrating EHR data from different clinical sites is critical for obtaining more accurate, generalizable and reproducible results [5]. Moreover, because of the healthcare process biases endemic in EHR, it is necessary to validate findings across multiple sites. This allows for assessment of clinical practice bias (e.g., one drug is prescribed more frequently at a particular hospital), race/ethnic disparities in populations that results in differences in the exposure and/or the outcome at a given site and other types of biases that may be due to the specific research database housed at a given institution [6].

To address these issues, the Observational Health Data Sciences and Informatics (OHDSI) consortium was formed (https://ohdsi.org/) for the primary purpose of developing open source tools that would be shareable across multiple sites. They also developed a Common Data Model [7] to enable each site to map their local data to a common shareable framework. This allows for a single script to be run across multiple sites without alteration. This simultaneously minimizes the probability of a database translation error (when a script is translated from one database structure to another to extract the same type of result) while speeding up the time to results.

Many studies have been conducted that have successfully utilized the OHDSI consortium, including a treatment pathways study [8], a birth season – disease risk study [9, 10] and several pharmacovigilance studies [11]. Using multiple sites allows researchers to study geographic variation [8, 10], which can be caused by regional changes in pollution and other exposures [10].

1.2. *Distributed Computing*

One barrier of institutional data sharing is regularity and government challenge on privacy protection [12]. In general, patient-level information with regards to important outcomes such as presence/absence of a medical condition or important confounders such as comorbidities, race/ethnicity, and age are not possible to share across institutions. As a consequence, current multi-site studies that rely on consortia, such as the OHDSI consortium [8, 10] or the eMERGE network (Electronic Medical Records and Genomics), can only utilize summary statistics that are shared across institutions. This necessitates the use of meta-analysis methods to aggregate signals from across the network [10].

As of 2018, the OHDSI consortium runs each script locally at a given institution and returns results, typically summary statistics (p-values, effect estimates) to the primary investigator for a given protocol. The Shared Health Research Information Network (SHRINE) has constructed a federated query network whereby analyses are run through the network and results are returned to the investigator [13]. If patient-level information were shareable in a privacy-preserving manner, it would enable more sophisticated patient-level statistical modeling and analyses [14].

Distributed Computing is a strategy where a computational goal is achieved by distributively computing its components from multiple sites. With data from multiple clinical sites, statistical analyses can be performed distributively without sharing patient-level information. For example, motivated by the pSCANNER project (patient-centered Scalable National Network for Effectiveness Research), a distributed algorithm for conducting logistic regression, termed as

GLORE (Grid Binary LOgistic Regression), was developed and deployed to pSCANNER consortium [12, 15]. Another example was the WebDISCO (a web service for distributed Cox model learning) method for fitting the Cox proportional hazard model [16] on EHR data from multiple clinical sites without sharing individual patient-level data [17]. These methods proved the utility and plausibility of a distributed privacy-preserving computing approach for obtaining results from multiple sites while still adjusting for patient-level covariates [15].

Despite their usefulness and promise, as acknowledged by the investigators, the aforementioned methods [12, 17] require iteratively transferring information across sites, which is time-consuming and labor-intensive in practice. Such practical limitation could be one of the barriers to adapt distributed algorithms in research consortia. This limitation motivated researchers to develop non-iterative distributed algorithms [18, 19]. A recently published paper by Jordan et al. proposed an innovative one-shot distributive computing framework, where the main idea is to construct a surrogate likelihood function through the use of patient-level data from a local site and aggregated information from other sites [20]. This idea was also proposed in distributed analysis for high-dimensional regression with sparsity [21]. In this study, we exercise the surrogate likelihood idea in logistic regression and develop a One-shot Distributed Algorithm to perform Logistic regressions (termed as ODAL). A major advantage of the proposed method, inherited from the merits of the surrogate likelihood [20], is that it only requires synthesizing summary statistics from multiple clinical sites *once*. Compared to algorithms that require iterative communication across sites, it is more practical to be deployed in research consortia.

2. Material and Method

In this section, we first present our motivating problem, then introduce our proposed method, and describe the design of simulation studies for evaluating the performance of our method.

2.1. *Clinical Cohort and Motivating Problem*

We extract females treated at one of the hospitals and/or clinics that comprise the University of

Table 1. Demographics of Pregnancies Treated at UPenn Health System

Demographics	Normal Pregnancy (N=30,810)	Fetal Loss (M=4,763)	P-value
Race			
White *	13911 (45.2%)	2291 (48.1%)	
African American	12918 (41.9%)	1871 (39.3%)	
Other	1916 (6.2%)	274 (5.8%)	
Asian	2065 (6.7%)	327 (6.9%)	
Age	29.40	32.15	<0.001
Weight (pounds)	126.26	115.43	<0.001
Body Mass Index	19.06	16.61	<0.001

* For race, we only used a binary variable for white versus non-white

Pennsylvania health system (abbreviated as UPenn). UPenn clinics are located in the entire Philadelphia Metropolitan area, which includes Delaware and Southern New Jersey. A pregnancy is defined as 'normal' if the woman was coded with any of the Z34 ICD-10 codes or a V22 ICD-9 code. A pregnancy is labeled as ending in fetal loss if any ICD-9 code is used within 630 through

639 or O00-O08 in the ICD-10 system. A similar fetal loss definition was used previously [22]. We only include patients who were prescribed or listed as taking at least 1 of the top 100 prescribed medications within 1 year prior to the first diagnosis of either a fetal loss or a normal pregnancy. The demographics of our cohorts are given in Table 1. P-values for differences between the fetal loss cohort and the normal pregnancy cohort are determined using a t-test. The variable race is dichotomized as white versus non-white in our models. The weight and BMI variables are averages across an individual's entire medical record. The statistics reported in Table 1 are excluding those with 0 weight or 0 BMI (i.e., indicating that no entries are available for those parameters). However, because the average weight and BMI is computed across the individual's entire record the value is smaller for those with longer records containing null entries.

Our proof-of-concept study involves predicting pregnancy outcome: fetal loss versus normal pregnancy. We include 4 relevant demographic covariates: age, race, Body Mass Index (BMI), and weight. We include our 'exposure' term of interest – namely the medication exposure. We ran our algorithm for each of the top 100 medications (ranked by drug prevalence) prescribed within 1 year prior to the pregnancy outcome while adjusting for the 4 demographic confounders. For purposes of this study, we randomly assign each pregnancy id to one of ten clinic IDs to ensure that an equal proportion of data is assigned to each of the ten clinics (approximately 3,557 pregnancies per clinic).

2.2. *Algorithm*

In this subsection, we introduce the distributed algorithm ODAL. First, we introduce the needed notations. We denote Y to be a binary outcome and z to be a $(p-1)$-dimensional vector, which contains the exposure of interest and potential confounders to be adjusted in a regression model. Let $x = (1, z)$. Suppose we have N observations from K different sites. Without loss of generality, we assume that each site contains n observations, noting that the algorithm also applies to sites with unequal sample sizes. Let (x_{ij}, Y_{ij}) denotes the i-th observation in the j-th site. Under the assumption of a logistic regression model, the log likelihood function for the combined data can be written as

$$L(\beta) = \frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{n}[Y_{ij}x_{ij}^{T}\beta - log\{(1 + exp(x_{ij}^{T}\beta)\}],$$

where β is a p-diemsional vector including the regression intercept and coefficients. Since the individual patient-level information is not allowed to be transferred across sites, we cannot obtain L(β) directly. To tackle this challenge, we apply Taylor expansion on the log likelihood function (1) around an initial value $\bar{\beta}$, and obtain

$$L(\beta) = L(\bar{\beta}) + \nabla L(\bar{\beta})(\beta - \bar{\beta}) + \sum_{j=2}^{\infty}\frac{1}{j!}\nabla^{j}L(\bar{\beta})\,(\beta - \bar{\beta})^{\otimes j}.$$

Suppose we have full access to the data stored in a local site (without loss of generality, assume it is the site 1). The log-likelihood at the local site can be written as

$$L_1(\beta) = \frac{1}{n}\sum_{i=1}^{n}[Y_{i1}x_{i1}^{T}\beta - log\{(1 + exp(x_{i1}^{T}\beta)\}]. \quad (2)$$

Similarly, we can expand the local log likelihood function $L_1(\beta)$ around an initial value $\bar{\beta}$,

$$L_1(\beta) = L_1(\bar{\beta}) + \nabla L_1(\bar{\beta})(\beta - \bar{\beta}) + \sum_{j=2}^{\infty}\frac{1}{j!}\nabla^{j}L_1(\bar{\beta})\,(\beta - \bar{\beta})^{\otimes j}.$$

Using the idea from Jordan et al. [20], higher order terms of the local likelihood $L_1(\beta)$ in (2) can be used to approximate the higher order terms of the combined likelihood $L(\beta)$ in (1), resulting in the following *surrogate likelihood* function after dropping some constant terms,

$$\tilde{L}(\beta) = L_1(\beta) + \left\{\frac{1}{K}\sum_{k=1}^{K} \nabla L_k(\bar{\beta}) - \nabla L_1(\bar{\beta})\right\}\beta, \quad (3)$$

where $\nabla L_k(\beta) = \frac{1}{n}\sum_{i=1}^{n}[Y_{ik} - exp(x_{ik}^T\beta)/\{1 - exp(x_{ik}^T\beta)\}]x_{ik}$.

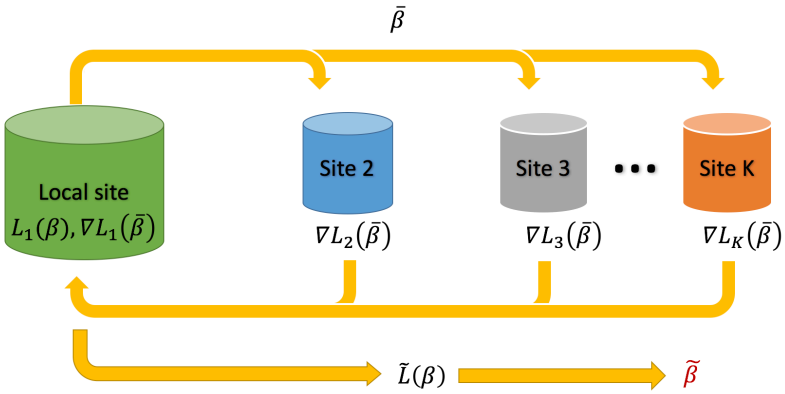


Figure 1. Schematic illustration of ODAL. Using data from the local site (i.e., site 1), the local estimator $\bar{\beta}$ is calculated and transferred to other sites. The intermediate term $\nabla L_j(\bar{\beta})$ is then evaluated at each site *j (j=2, ..., K)* and transferred back to the local site. Combined with $\nabla L_1(\bar{\beta})$ and $L_1(\beta)$, we construct the surrogate function $\tilde{L}(\beta)$ in the local site and obtain the ODAL estimator $\tilde{\beta}$ by maximizing $\tilde{L}(\beta)$.

There are several notable features of the above surrogate likelihood. First, the terms $L_1(\beta)$ and $\nabla L_1(\bar{\beta})$ can be calculated using data from the local site. Secondly, the term $\nabla L_k(\bar{\beta})$ can be computed from site k and transferred to the local site. Note that each $\nabla L_k(\bar{\beta})$ is with dimension p and contains only aggregated information. Therefore, the information transferring maintains low communication cost and is privacy preserving. The ODAL estimator is then obtained locally by minimizing the surrogate likelihood function in equation (3), i.e.

$$\tilde{\beta} = \arg\max_{\beta} \tilde{L}(\beta).$$

Regarding the initial value $\bar{\beta}$, a nature choice of $\bar{\beta}$ is the maximum likelihood estimator of the local likelihood $L_1(\beta)$. A detailed algorithm is outlined below.

Algorithm: ODAL

1. Initial value: obtain $\bar{\beta} = \arg\max_{\beta} L_1(\beta)$ using data in the local site (i.e., site 1), where $L_1(\beta)$ is the log likelihood of logistic regression defined in equation (2)
2. Initial communication: transfer $\bar{\beta}$ to the other sites (i.e., sites 2, 3, …, *K*)
3. For j = 2 to *K*,
4. **do** compute $\nabla L_j(\bar{\beta})$, where $L_j(\beta)$ is defined similarly as in equation (2)
5. transfer $\nabla L_j(\bar{\beta})$ to the local site
6. **end**
7. Compute the surrogate likelihood $\tilde{L}(\beta)$ defined in equation (3)

8. Obtain $\tilde{\beta} = \arg\max_{\beta} \tilde{L}(\beta)$
9. **return** $\tilde{\beta}$

2.3. *Simulation Design*

To evaluate the empirical performance of the ODAL method, we consider a setting where a binary outcome is associated with two continuous risk factors and two binary risk factors. We generate the two continuous variables from a standard normal distribution $N(0,1)$ and a uniform distribution $U(0,1)$ respectively. The two binary variables are generated from Bernoulli distributions with probability 0.1 and 0.5 respectively. Slightly different from the previous notation, we let x denote the vector of all the risk factors. The outcome Y is generated from Bernoulli distribution, with the conditional probability satisfying the logistic regression model,

$$logit(Pr(Y = 1|x)) = \alpha + x^T\beta,$$

where $logit(p) = log\{p/(1-p)\}$, β is the vector of coefficients and α is the regression intercept.

To mimic a distributed research network, we generate a total number of N subjects and randomly divide them into K sub-datasets. The local dataset is set to be the first sub-dataset and the number of subjects of the local dataset is n. We design the simulation study to investigate the relative accuracy of the ODAL compared to the following two methods:

(i) the pooled estimator: the individual patient-level data pooled from all clinical sites are used, which can serve as a gold standard for the best possible accuracy; i.e., the estimate that maximizes the log likelihood in equation (1);

(ii) (ii) the local estimator: only individual patient-level data from the local site are used; i.e., the estimate that maximizes the log likelihood in equation (2).

We use mean square error (MSE) to summarize the performance of the three estimators and consider the following four scenarios:

A. We randomly generate data for N patients, and evenly divide them into 10 sites. We increase N from 1000 to 10000. This reflects a setting where a network, such as PEDSnet (the National Pediatric Learning Health System), contains a fixed number of pediatric hospitals, but the number of patients increases over time and is updated quarterly [23].
B. We randomly generate data from K sites each has 1000 patients, and increase K from 2 to 100. This is a setting where a consortium involves a growing number of clinical sites, and the number of patients per site is relatively stable. For example, the Hospital Compare dataset (https://www.medicare.gov/hospitalcompare/search.html) contains results from Meaningful Use measures (e.g., 30-day readmissions) for increasing number of hospitals reporting those measures, however the average hospital size remained relatively constant.
C. We randomly generate data for 10000 patients, and evenly divide them into K sites. We increase K from 2 to 100. This setting is included to investigate the relative performance of the ODAL for a small versus large number of clinical sites, while holding the total number of patients fixed. Depending on how the data are stored in each hospital, the investigators may choose to perform a distributive analysis on the hospital-level or on the clinic-level.

D. We randomly generate data for 10000 patients divide them into 10 sites. The local site has sample size n, and other 9 sites evenly split the rest of data. We increase n from 100 to 9100. This setting is to investigate the performance of the ODAL when the relative size of the local site, compared to the total number of patients, increases from a small percentage to a large proportion. For example, OHDSI contains many sites of varying sizes from 0.5 million patients to hundreds of millions of patients. Depending on where an investigator is located, the 'local' dataset will vary with regards to the proportion of the dataset as a whole.

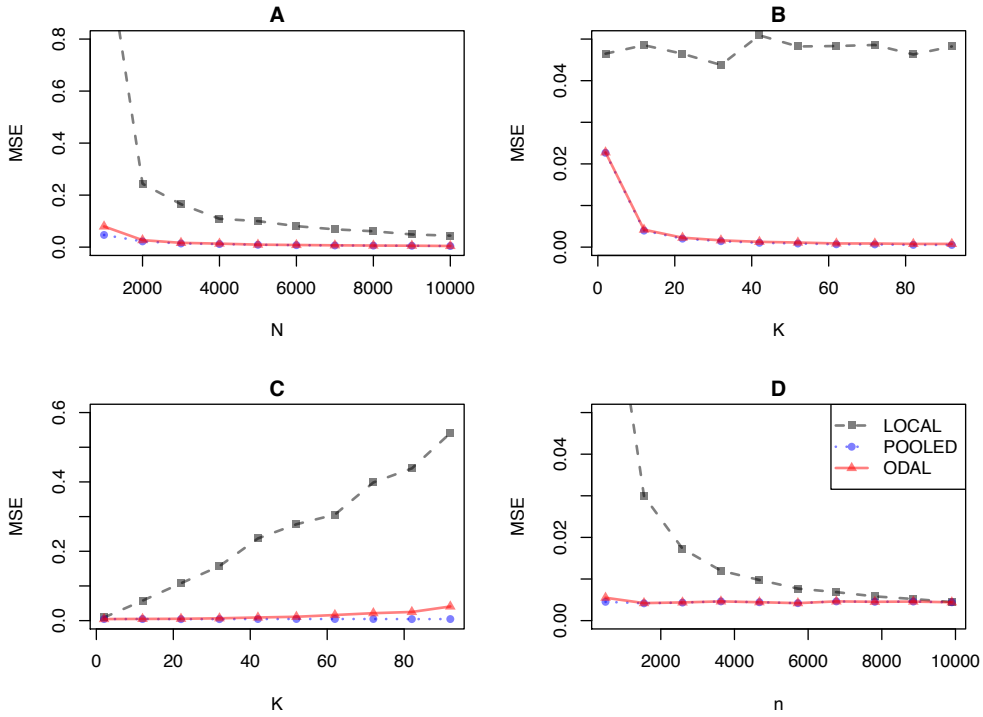


Figure 2. Mean square errors (MSE) of ODAL, the pooled and the local estimators under settings A, B, C and D. In setting A (upper left panel), we evenly divide N subjects in to 10 sub-datasets, and increase N from 1000 to 10000. In setting B (upper right panel), each site contains 1000 subjects and the number of sites K is then increased from 2 to 100. In setting C (lower left panel), we generate 10000 subjects, and evenly divide them into K sub-datasets, where K increases from 2 to 100. In setting D (lower right panel), we generate 10000 subjects, and divide them into 10 sub-datasets, where the local dataset has n subjects and the other 9 sub-datasets has the equal number of subjects. We increase n from 100 to 9900.

3. Results

3.1. *Simulation Results*

Figure 2 presents the mean square errors of the ODAL, the pooled and the local estimators under four different scenarios. Overall, it shows that in all considered scenarios, the ODAL provides estimates with comparable accuracy as the best possible pooled estimates. In Setting A, where number of sites is fixed and each site has relatively the same number of subjects, ODAL can reach almost the same accuracy as the pooled estimator when total sample size is relatively large. When total sample size is limited, ODAL can still provide much more accurate estimation than the local

estimator (MSE of the local estimator is 15 times higher than MSE of ODAL when $N = 1000$). *This suggests that by borrowing simple gradient information* $\nabla L_k(\bar{\beta})$ *from other sites, ODAL gained substantial statistical efficiency compared to the estimate using the data at the local site alone.* Setting B shows that by borrowing information from more sites, the accuracy of estimation increases. In addition, the ODAL and the pooled estimators provide estimates with negligible difference in accuracy.

Setting C shows that by dividing a fixed number of subjects into increasing number of sites, as expected, the performance of the pooled estimator stays the same. ODAL performs as good as the pooled estimator when the number of sites is relatively small. With increasing number of sites, ODAL has slightly increased amount of error, but is much more accurate compared to the local estimator (MSE of the local estimator is 13 times of the MSE of the ODAL estimate when $K = 100$). The results from Setting C suggest that ODAL can guarantee reasonable accuracy even when the number of sites are moderately large. Such investigation also provides quantitative guidance on choosing between performing the distributed analysis at the clinic level (relatively large number of sites) or the hospital level (relatively small number of sites). Setting D considers the influence of number of subjects contained in the local sites on the accuracy of each methods. As expected, the local estimator performs worse with smaller number of subjects in the local site. The change of local sample size does not influence the performance of the pooled estimator since the total sample size is fixed. Compared to the pooled estimator, the ODAL performs almost the same where the ratio of MSE decreases from 1.22 to 1.00 with the increase of local sample size.

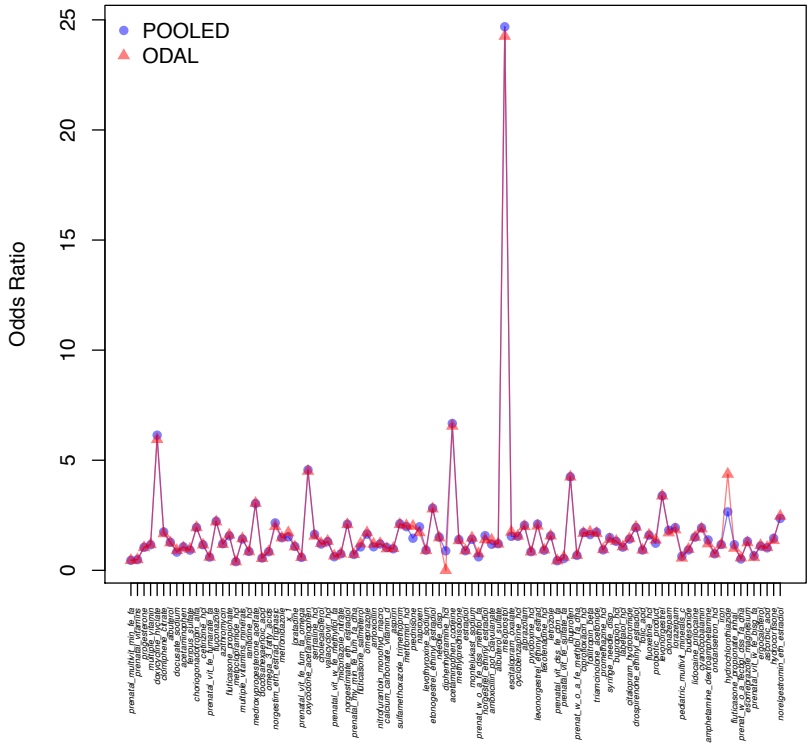


Figure 3: Odds ratio estimates from the ODAL method (red triangles) and the pooled data (blue circles) for 100 medications and their associations with fetal loss. The 100 medications from left to right are sorted by their prevalence in the population.

The distributed algorithm GLORE, which leads to exactly the same estimate as the pooled estimator, requires cross-site iterations until a convergence is reached. In our simulation, the number of iterations to obtain the pooled estimates ranges from 6 to 10 using the glm() function in R 3.4.1. In the case with more covariates involved, it may require larger number of iterations to achieve convergence, which creates a substantial burden in communication across clinical sites.

3.2. *Fetal Loss Prediction via ODAL*

We apply ODAL to the EHR data described in Section 2.1 to evaluate the risks of fetal loss due to various medication exposures. We include the top 100 medications prescribed within 1 year prior to a normal pregnancy or fetal loss outcome. We randomly assign each of our pregnancies to 1 of 10 clinics to test the performance of ODAL. We include one medication at a time adjusting for maternal age, race/ethnicity (collapsed to a binary variable of White versus non-White), weight and BMI. Figure 3 compares the estimates from ODAL to the pooled estimator. The average relative difference in the odds ratios between ODAL and the pooled estimator is 0.0046 across all 100 medications. This indicates that the result from ODAL is very close to the result that would be achieved if all individual-level data are pooled together for the analysis.

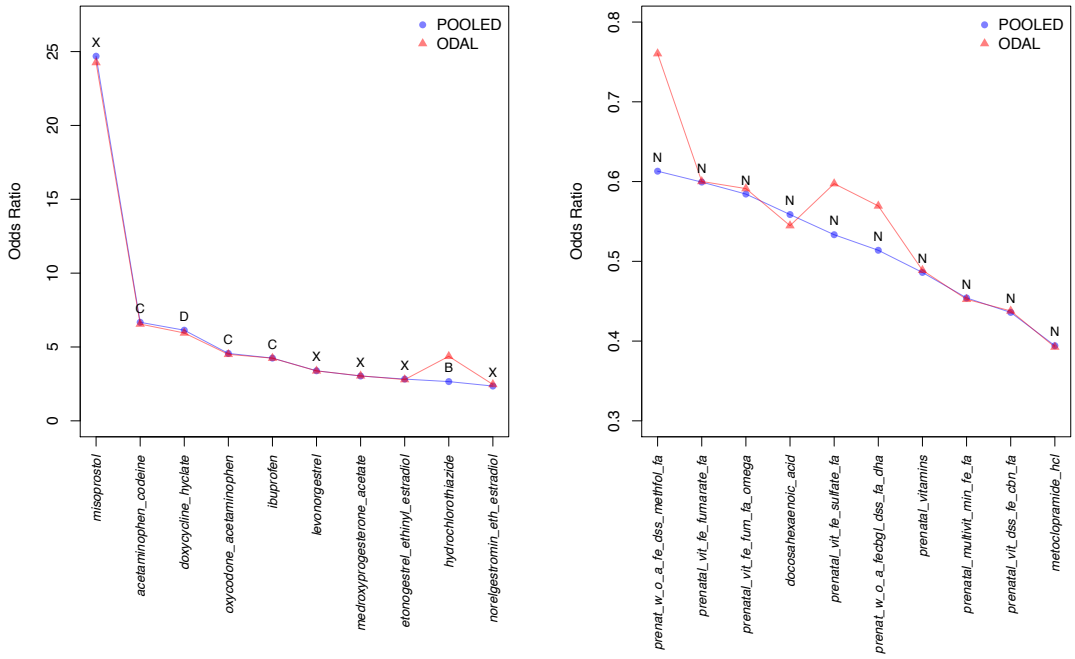


Figure 4: Odds ratio estimates from ODAL and the pooled estimator for the top 10 medications positively associated (left panel) and negatively associated (right panel) with fetal loss. On the left panel, the ten medications are misoprostol, acetaminophen codeine, doxycycline hyclate, oxycodone acetaminophen, ibuprofen, levonorgestrel, medroxyprogesterone acetate, etonogestrel ethinyl estradiol, hydrochlorothiazide and norelgestromin eth estradiol. On the right panel, the ten acronyms are referring to prenatal vitamins (without vit. A) with DHA, iron, folic acid and docusate sodium; Prenatal vitamins with Iron fumarate, Folic Acid; Prenatal vitamin with Folic Acid and DHA; DHA; Prenatal vitamins with Iron, Sulfate, and Folic Acid; Prenatal vitamin (without vit. A) with DHA, Folic Acid, Extra Iron and Docusate sodium; Prenatal vitamins; Prenatal multi-vitamin with Folic Acid and minimum Iron; Prenatal vitamins with Iron, Docusate sodium, and Folic Acid; Metoclopramide hcl. The letter on each medication shows the

FDA assigned pregnancy category, where A, B and C means of no or unknown risk, D and X means of risk. N means the medication is not assigned a category. Detailed interpretations can be found at https://chemm.nlm.nih.gov/pregnancycategories.htm.

Figure 4 presents the top 10 medications that are positively associated (left panel) and bottom 10 medications that are negatively associated (right panel) with fetal loss. To compare our findings with existing knowledge in the literature, we use information on the pregnancy safety of the drug using the Food and Drug Administration (FDA)'s A-X category system. This information is readily obtainable from drugs.com, a freely available online resource, for drugs and their various therapeutic uses and effects (https://www.drugs.com/). Each drug's FDA category is shown above in Figure 4. Drugs in category A are drugs where no fetal risk has been observed in controlled human studies, category B drugs are drugs with no evidence of fetal risk in animal models but well-controlled human studies are lacking, category C drugs are drugs where fetal risk has been shown in animal models but the effects are unknown in humans while category D and X are drugs with known evidence of some fetal risk in humans and animals [24]. Of the top 10 drugs associated with fetal loss Figure 4, six are either category D or X with known evidence of fetal risk in the literature. Three drugs are category C pain relievers, two are drug combos of Tylenol (acetaminophen) with an opioid (codeine or oxycodone) while ibuprofen is an over-the-counter pain reliever. The only category A or B drug in the top 10 is hydrochlorothiazide (a diuretic that treats hypertension), a category B drug. However, hydrochlorothiazide is considered a category D drug, and contra-indicated in pregnancy, is used to treat pregnancy-related hypertension. Therefore, there is likely a dosage that is fetal toxic. In the ten medications that are negatively associated with fetal loss, we identify 8 types of prenatal vitamins with folic acid, docosahexaenoic acid (DHA) and metoclopramide hcl. These findings are consistent with the literature on the importance of prenatal vitamins to prevent early term miscarriages and fetal loss. For example, it has been suggested by many studies that folic acid has positive impacts on preventing early pregnancy loss [25]. In summary, the ODAL method leads to estimates that are highly consistent with the pooled estimates, and the identified associations are also consistent with our current understanding of these medications.

4. Discussion

The integration of EHR data from multiple healthcare databases increases statistical sample size and heterogeneity of exposure, as well as reduces clinical bias and improves the power of statistical analyses. The rise of large healthcare networks, such as ODHSI, pSCANNER, SHRINE and PEDSnet provide platforms for data integration and evidence synthesis [23]. To avoid sharing individual-level information, distributed algorithms have been developed which can conduct population-level analyses in a privacy-preserving manner. In this paper, we propose a novel privacy-preserving and communication-efficient distributed algorithm to study binary outcomes with a set of risk factors using logistic regression. As demonstrated by our simulation study and the application to fetal loss data analysis, our algorithm provides a close approximation to the pooled estimator where all patient-level information is pooled together.

The communication efficiency of our algorithm comes from two aspects. First, in contrast to the existing iterative algorithms such as GLORE and WebDISCO, our algorithm does not require

iterative communication across sites. This is crucial especially in the healthcare field where data and information exchange often require large amount of administrative work and technical support. On the other hand, the intermediate result that need to be transferred in the ODAL method is only the first gradient of the likelihood function evaluated at an initial value, which is a vector with dimension equal to the number of parameters p. In contrast, for algorithms such as GLORE [12], in each iteration, the value of the second gradient of the likelihood function need to be transferred, which is a $p \times p$ matrix. When studying a large amount of risk factors, for example large number of potential confounders or genetic variations, the dimension of the matrices can be big which might cause high communication cost for transferring the data.

On the other hand, ODAL requires access of individual patient-level data for one clinical site, in order to construct the surrogate likelihood function. In situations where individual patient-level data are inaccessible in any site, GLORE is preferred.

The OHDSI consortium consists of many partner institutions where patient-level data sharing is not permissible as this often conflicts with regional legislation. In this instance an individual researcher may have patient-level data available at their given site, but then would deploy their algorithms at other sites without having access to the patient-level data. For these situations ODAL is ideal because aggregated information from other sites is only borrowed once without having access to the patient-level data in those countries and regions where that is impermissible.

Deploying ODAL within OHDSI and other large consortia would enable us to further validate our findings with regards to medications taken within 1-year prior to normal pregnancy or fetal loss diagnoses. Validation of these results and also larger scale assessment of medications that potentially increase the risk of fetal loss is still much needed. Algorithms have been developed to assess the fetal effect of category C medications [22], but these can often be limited by confounding and other local institution-specific biases. Use of ODAL across a large international consortium such as OHDSI would propel adequate assessment of each drug's fetal toxicity even for those where the effects remain unknown (i.e., category C medications).

In the future, we are planning to extend our method to other types of outcomes, such as continuous, categorical, and time-to-event data. Furthermore, we are developing open-source software packages for directly implementing ODAL on distributed networks. We believe that our algorithm can be a good complement to the existing distributed algorithms.

References

1. Blumenthal, D. and M. Tavenner, *The "meaningful use" regulation for electronic health records.* N Engl J Med, 2010. **363**(6): p. 501-4.
2. Andreu-Perez, J., et al., *Big data for health.* IEEE J Biomed Health Inform, 2015. **19**(4): p. 1193-1208.
3. Smith, B., et al., *The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration.* Nat Biotechnol, 2007. **25**(11): p. 1251-5.
4. Goble, C. and R. Stevens, *State of the nation in data integration for bioinformatics.* J Biomed Inform, 2008. **41**(5): p. 687-93.
5. Sutton, A.J., et al., *Meta-analysis of rare and adverse event data.* Expert Rev Pharmacoecon Outcomes Res, 2002. **2**(4): p. 367-79.

6. Katzan, I.L. and R.A.J.S.t.m. Rudick, *Time to integrate clinical and research informatics.* 2012. **4**(162): p. 162fs41-162fs41.
7. Overhage, J.M., et al., *Validation of a common data model for active safety surveillance research.* Journal of the American Medical Informatics Association, 2011. **19**(1): p. 54-60.
8. Hripcsak, G., et al., *Characterizing treatment pathways at scale using the OHDSI network.* Proceedings of the National Academy of Sciences, 2016. **113**(27): p. 7329-7336.
9. Boland, M.R., et al., *Birth month affects lifetime disease risk: a phenome-wide method.* Journal of the American Medical Informatics Association, 2015. **22**(5): p. 1042-1053.
10. Boland, M.R., et al., *Uncovering exposures responsible for birth season–disease effects: a global study.* Journal of the American Medical Informatics Association, 2017. **25**(3): p. 275-288.
11. *Large-scale adverse effects related to treatment evidence standardization (LAERTES): an open scalable system for linking pharmacovigilance evidence sources with clinical data.* Journal of biomedical semantics, 2017. **8**: p. 1-15.
12. Wu, Y., et al., *Grid Binary LOgistic REgression (GLORE): building shared models without sharing data.* J Am Med Inform Assoc, 2012. **19**(5): p. 758-64.
13. Weber, G.M., et al., *The Shared Health Research Information Network (SHRINE): a prototype federated query tool for clinical data repositories.* Journal of the American Medical Informatics Association, 2009. **16**(5): p. 624-630.
14. Boyle, D. and N. Rafael, *BioGrid Australia and GRHANITE™: privacy-protecting subject matching.* Studies in health technology and informatics, 2011. **168**: p. 24-34.
15. Ohno-Machado, L., et al., *pSCANNER: patient-centered Scalable National Network for Effectiveness Research.* Journal of the American Medical Informatics Association, 2014. **21**(4): p. 621-626.
16. Cox, D.R., *Regression models and life-tables*, in *Breakthroughs in statistics*. 1992, Springer. p. 527-541.
17. Lu, C.-L., et al., *WebDISCO: a web service for distributed cox model learning without patient-level data sharing.* 2015. **22**(6): p. 1212-1219.
18. Zhang, Y., M.J. Wainwright, and J.C. Duchi. *Communication-efficient algorithms for statistical optimization*. in *Advances in Neural Information Processing Systems*. 2012.
19. Battey, H., et al., *Distributed testing and estimation under sparse high dimensional models.* 2018. **46**(3): p. 1352-1382.
20. Jordan, M.I., J.D. Lee, and Y.J.J.o.t.A.S.A. Yang, *Communication-efficient distributed statistical inference.* 2018(just-accepted).
21. Wang, J., et al., *Efficient distributed learning with sparsity.* 2016.
22. Boland, M.R., F. Polubriaginof, and N.P. Tatonetti, *Development of A Machine Learning Algorithm to Classify Drugs Of Unknown Fetal Effect.* Scientific reports, 2017. **7**(1): p. 12839.
23. Forrest, C.B., et al., *PEDSnet: a national pediatric learning health system.* 2014. **21**(4): p. 602-606.
24. Boothby, L.A. and P.L.J.A.o.P. Doering, *FDA labeling system for drugs in pregnancy.* 2001. **35**(11): p. 1485-1489.
25. Nelen, W.L., et al., *Homocysteine and folate levels as risk factors for recurrent early pregnancy loss.* 2000. **95**(4): p. 519-524.

PVC Detection Using a Convolutional Autoencoder and Random Forest Classifier

Max Gordon† and Cranos Williams

Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, North Carolina 27607, USA

†*E-mail: mjgordo3@ncsu.edu*

www.ncsu.edu

The accurate detection of premature ventricular contractions (PVCs) in patients is an important task in cardiac care for some patients. In some cases, the usefulness to physicians in detecting PVCs stems from their long-term correlations with dangerous heart conditions. In other cases their potential as a precursor to serious cardiac events may make their detection a useful early warning mechanism. In many of these applications, the long-term nature of the monitoring required and the infrequency of PVCs make manual observation for PVCs impractical. Existing methods of automated PVC detection suffer from drawbacks such as the need to use difficult to extract morphological features, domain-specific features, or large numbers of estimated parameters. In particular, systems using large numbers of trained parameters have the potential to require large amounts of training data and computation and may have issues generalizing due to their potential to overfit. To address some of these drawbacks, we developed a novel PVC detection algorithm based around a convolutional autoencoder to address these weaknesses and validated our method using the MIT-BIH arrhythmia database.

Keywords: Electrocardiogram; Premature Ventricular Contraction (PVC) Detection; Autoencoder.

1. Introduction

Electrocardiograms (ECGs) are a useful and noninvasive diagnostic and monitoring tool in cardiac care.[1] One significant application of ECGs in cardiology is their use in the monitoring and treatment of arrhythmias. Premature Ventricular Contractions (PVCs) are a common arrhythmic beat type that occurs commonly in many patients, including individuals with good cardiac health.[2] However, when they occur in large numbers or with high frequency in patients with other risk factors, PVCs can be associated with serious cardiac problems and may precede heart attacks or sudden cardiac death in rare cases.[2] As a result, the automated detection of PVCs in ECG records would allow information about their long-term frequency to be tracked over time, providing a new means to track the trends in a patient's cardiac health as well as potentially providing an early warning of events requiring swift medical attention.

There are several main categories of approaches to feature extraction for the automated detection of PVCs: 1) morphological and timing features extracted from the ECG signal[3–5]

and 2) time-frequency features such as wavelet transforms of the ECG signal.[6,7] In addition to these two main approaches to PVC detection, there are methods utilizing other approaches to the connected problems of feature extraction and beat classification,[8] Markov models, independent component analysis,[9] and autoencoders.[10]

Geddes and Warner[3] used R-R interarrival time, QRS complex duration, and signal slope during several sections of the QRS complex as features in their detection system. They made classification decisions based on a manually constructed decision tree. This allowed for computationally simple evaluation of a QRS complex but sacrificed adaptibility and required heuristic tuning and domain specific knowledge of the PVC detection problem to adjust the classifier. Trahanias et al.[4] used a number of structural descriptors to create a syntactic description of the QRS complex. After this syntactic description was created, they used a normalized distance metric to form classes of QRS complexes, which were found to correspond to some clinically significant classes of heartbeats. However, this method did not lead to a direct and useful classification of the QRS complex. Zadeh et al.[5] used a total of 10 morphological features and 3 timing features extracted from the signal of a detected QRS complex. They compared several kinds of classifiers including MLP neural networks, RBF neural networks, probabilistic neural networks, and support vector machines. In addition to detecting PVCs, they used their classification system to identify non-PVC arrhythmias.

In all of these approaches, significant domain knowledge was used to determine feature sets and detection accuracy was dependent on the classification of different parts of the QRS complex for segmentation and measurement. It is desirable to avoid these issues by using a more general and robust method of feature extraction. Ham and Han[6] used two estimated linear prediction coefficients in combination with the mean squared value of the signal as features for classification. They used a fuzzy ARTMAP neural network to perform the classification. Lim[7] used a discrete wavelet transform with the Haar wavelet to generate a feature vector and used a fuzzy neural network for classification. While these approaches still require manual feature selection, the specific features extracted are less domain specific and do not require segmentation of the QRS complex to calculate.

One approach to avoid the challenges associated with engineering a problem-specific feature set is to use feature learning approaches such as independent component analysis or autoencoders to extract a feature set that is able to describe much of the information content of a signal in a low-dimensional latent space.[11] Yu and Chou[9] used independent component analysis to identify and extract a set of features, which were combined with QRS complex timing information to create the full feature set passed to their neural network classifier. Yang et al.[10] used a sparse autoencoder (SAE) to generate a feature vector for classification. This resulted in a large number of estimated network weights, which increased the computation and data required to train the network and increased the potential for overfitting.

The primary aims of this study are to develop a system for the detection of PVCs in ECG data that does not rely on manually selected features and has fewer parameters to be estimated than existing SAE methods. These improvements will reduce the possibility of overfitting and improve the generalization of the detection system. For this purpose, we used an autoencoder architecture based on convolutional layers to extract and select features for use in classifying

beats. Our architecture is differentiated from existing convolutional autoencoders (CAEs)[12] by its multi-stage encoding process, which allows it to encode information about the frequency content of a signal at different points in time.

2. Methods

2.1. *Data Set and Implementation*

We used ECG records from the MIT-BIH arrhythmia database annotated with beat locations and types.[13] This database consists of 48 30-minute 2 channel ECG records sampled at 360 Hz. Only channel 1 of the ECG was used for PVC detection because in the MIT-BIH arrhythmia database this signal is a modified limb lead II, which has clearer signals for non-ectopic beats than the modified lead V1 available on channel 2. As much of the information content of a QRS complex is centered on the R peak, the ECG signals obtained from the database were segmented based on the annotated R peaks, with 89 samples before and 160 samples after each annotated R peak extracted for feature calculation. In application outside the MIT-BIH database, this means we assume the QRS complexes are reliably detected before being passed to our detection system. We then removed the mean from each segmented QRS complex to reduce the impact of baseline drift, variations in instrumentation, and differences across patients. The PVC detection system was implemented in Python using the Keras,[14] TensorFlow,[15] and scikit-learn[16] libraries.

2.2. *Proposed PVC Detection Method*

A convolutional autoencoder (CAE)[12] was used to extract and select features for classification automatically and in an unsupervised manner from ECG data annotated with beat locations. This reduced the need for domain-specific knowledge as compared to manual feature selection. Compared to a SAE, a CAE reduces the number of weights that need to be trained, increases the robustness of the features extracted when the window alignment of the beats being processed is variable, and takes advantage of the structure of the ECG signal in its architecture. We used a Random Forest Classifier to perform the final PVC detection due to its resistance to overfitting and its performance with the indistinct groupings of PVC and non-PVC beats. Our system architectures for training the CAE and Random Forest Classifier are shown in Figure 1, while our classification architecture is shown in Figure 2. Examples of normal beats and PVCs are given in Figure 3

2.2.1. *Feature Extraction*

An autoencoder is a neural network that encodes its input to a latent space representation attempts to decode this representation to recover the inputs.[17] In a CAE, the layers responsible for encoding and decoding the latent space are convolutional, using shared weights to kernels to extract features from their input. After the network has been trained, the encoding layers alone can be used to reduce the dimensionality of the input data for further processing.

In the proposed PVC detection method, two convolutional layers with linear activations were used to encode the input to the CAE. The first of these layers generated n kernels of

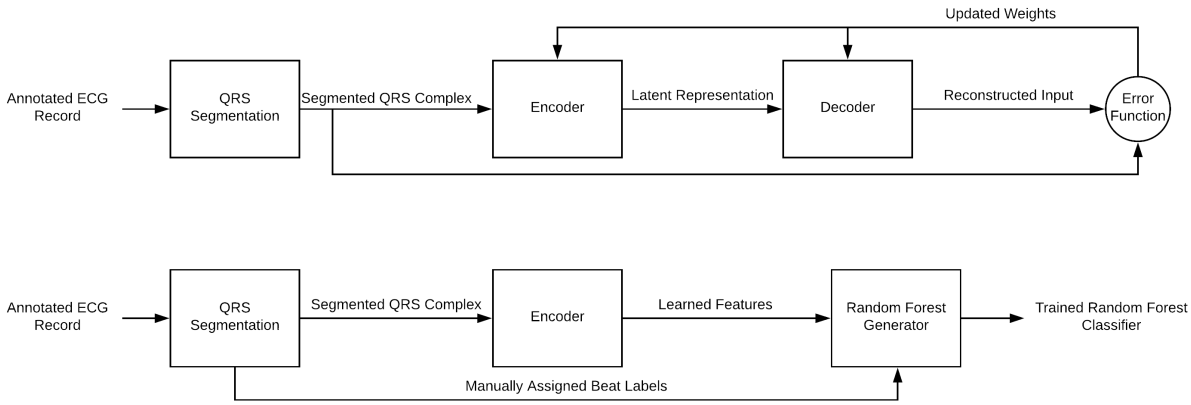


Fig. 1. CAE and Random Forest Training Architecture

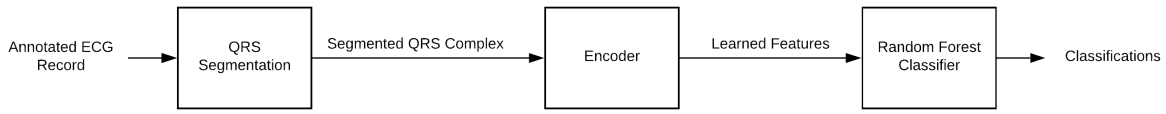


Fig. 2. Classification Architecture

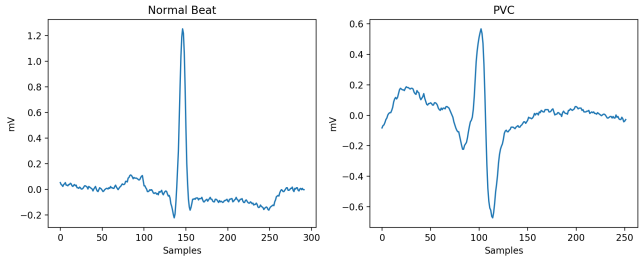


Fig. 3. Comparison of Normal Beat and PVC

order m to extract different features from the input. A stride length of k was used in this layer to downsample the input, reducing its dimensionality. The second convolutional layer generated a single kernel to compute a linear combination of the outputs of the previous layers kernels at each point. This second layer serves as a feature selection stage. As a result, each feature in the latent space representation of the input corresponds to a combination of all features extracted in the first layer from a continuous subset of the input. This provides information on the frequency components of the ECG signal most important for creating an accurate reconstruction of the original signal as well as some degree of temporal localization within the signal. This allows the encoded representation to contain distinct information about various stages in the progression of the QRS complex without the need to explicitly define and detect these stage, simplifying the PVC detection process in comparison to methods using morphological features of the QRS complex.

We used transposed convolutional layers to decode the latent space representation generated by the encoder. These layers have the same connectivity and dimensionality as the encoding layers but are reversed. This results in an output matching the dimensionality of the input to the CAE and allows us to train the network to replicate its inputs. In operation, only the encoding side of the network was used to generate the features used in classification. The resulting network architecture is shown in Figure 4.

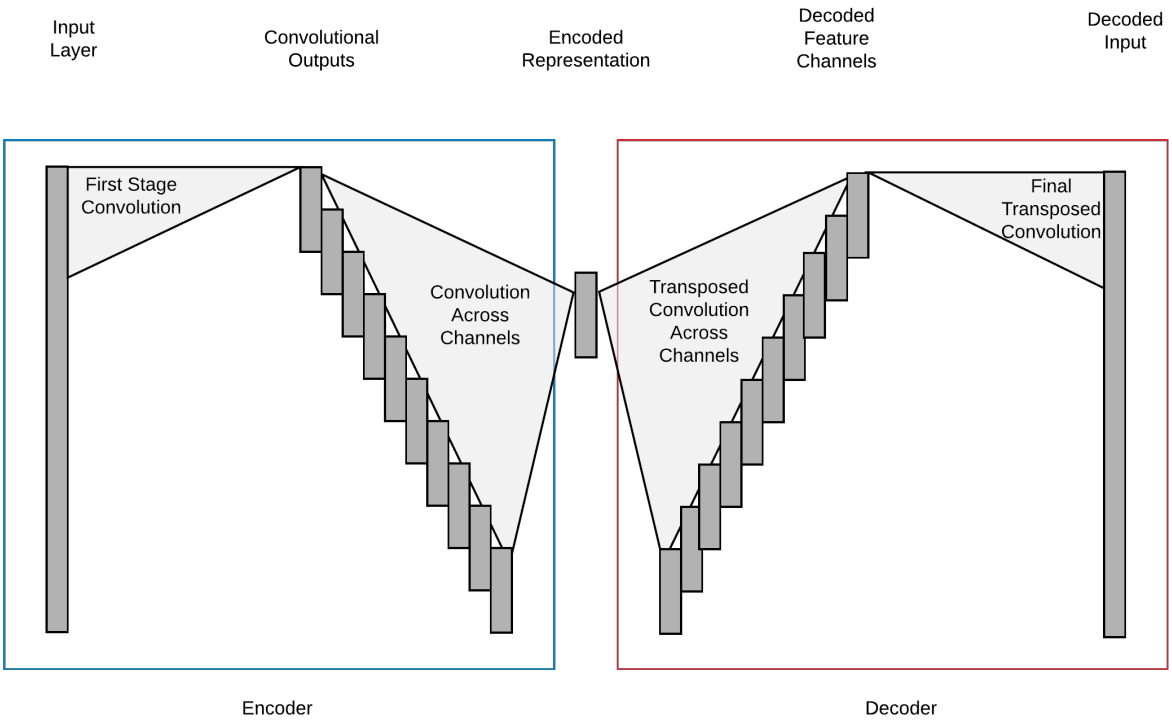


Fig. 4. Convolutional Autoencoder Architecture

For this application, the length of signal extracted around each beat even was 250 samples, with 89 samples before the annotation and 160 samples after the annotation. These values were selected because they were found to provide generally acceptable classification performance and allowed for a more direct comparison with the PVC detection system described by Yang et al.[10] An n value of 25 provided a sufficient number of base features for the following layer to perform feature selection on. An m value of 20 provided sufficiently complex filters to extract a wide range of characteristics from the signal. A k value of 10 allowed the final feature vector to be of dimension 25. This was found to provide sufficient segmentation of the input signal in time while also being of low enough dimensionality to allow for adequate classifier performance. The CAE was trained using an ADAM optimizer as described by Kingma and Ba[18] with a learning rate of 0.01 and a mean squared error loss function: $MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$, where Y is the input to the autoencoder and $\hat{Y}$ is the output of the autoencoder.

2.2.2. *Classification*

We used a Random Forest Classifier as described by Breiman.[19] The random forest used in this detection system consisted of 10 decision trees with Gini impurity as their splitting criterion. Gini impurity is the probability that a randomly selected element in a set would be mislabeled if labeled at random.[19] For J classes with probability of selection p, the Gini impurity of a set is given by $I_G = 1 - \sum_{i=1}^{J} p_i^2$. The features used to split each node of the tree were randomly determined. The classifier also used bagging to avoid overfitting, using a set of training examples of the same size as the full dataset sampled without replacement as the training dataset for each random tree. The Random Forest Classifier was chosen due to its low number of parameters, its resistance to overfitting, and its ability to handle fuzzy group boundaries in comparison to support vector methods, neural networks, and other common classifiers.

3. Results

We evaluated our method with 3 tests. First, we tested its performance when provided with ample training data including samples from each record. Next, we added a randomized error to the R peak location used in segmentation to simulate inaccurate QRS detection. Finally, we provided our system with training data that included no beats from the records used for testing to evaluate its ability to generalize to new patients. Each of these tests was also performed using a SAE to provide context to the performance of the CAE. In addition to the testing we performed, we examined the number of estimated weights and the number of training epochs necessary for convergence in both the CAE and SAE architectures.

3.1. *Full Database Evaluation*

We evaluated the classification system using the MIT-BIH arrhythmia database. Half of the beats from each record were selected as training data and the remainder were used as testing data. This resulted in a training set consisting of 54,695 beats with 3,495 PVCs and a testing set consisting of 54,675 beats with 3,633 PVCs. The results of this testing are shown in Table 8 with information for each record. A SAE similar to one described by Yang et al.[10] was constructed, with the sparsity imposed by L1 regularization instead of the Kullback-Leibler divergence derived regularization described, to compare the feature extraction provided by the CAE to that provided by an existing alternative architecture. A comparison of the performance of these two architectures is provided in table Table 1 and Table 2. This evaluation demonstrates that the CAE provides similar performance to the SAE when ample training data is available, with a difference in overall accuracy of 0.2%. However, the PVC sensitivity of the CAE is 4.88% higher than that of the SAE, meaning that fewer PVCs are missed by the CAE. This is desirable given the relative rarity of PVCs, although the importance of sensitivity and specificity will need to be evaluated for individual applications.

3.2. *Timing Disturbance Evaluation*

As QRS detection is necessary to the identification and segmentation of potential PVCs for processing by a PVC detection system, this property makes resistance to small shifts in the

Table 1. MIT-BIH Full Database Comparative Evaluation

Architecture	Correct	PVC Sensitivity	PVC Specificity
CAE	98.43	85.64	98.90
SAE	98.23	80.76	99.07

Table 2. Full Database CAE and SAE Confusion Matrices

	CAE		SAE	
	True Normal	True PVC	True Normal	True PVC
Detected Normal	50483	299	50565	489
Detected PVC	559	3334	477	3144

precise placement of the annotation within the beat desirable. We evaluated this robustness by applying a random shift of up to 36 samples to each beat, corresponding to a detection error of up to 100 milliseconds. The results of this testing on the CAE are shown in Table 9 with information for each record, while a comparison of the performance of the CAE and SAE architectures under these conditions is presented in Table 3 and Table 4. This shows that the CAE suffers a 0.83% reduction in PVC sensitivity as a result of this shifting, while the SAE suffers a 4.26% reduction in PVC sensitivity. This results in a total sensitivity improvement for the CAE of 8.43% relative to the SAE under these conditions.

Table 3. MIT-BIH Full Database Disturbed

Architecture	Correct	PVC Sensitivity	PVC Specificity
CAE	97.60	84.93	98.42
SAE	97.17	76.50	97.66

Table 4. Full Database CAE and SAE Disturbed Confusion Matrices

	CAE		SAE	
	True Normal	True PVC	True Normal	True PVC
Detected Normal	50542	810	50708	1217
Detected PVC	501	2823	335	2416

3.3. *Cross-Patient Training Evaluation*

In an applied setting, it may not always be practical to obtain annotated training data from a patient to train any monitoring system. As a result, system performance when trained only using data obtained from other individuals is potentially important to the practical utility of any PVC detection method. We evaluated this performance metric by training both PVC

detection systems using all beats in two ECG records and testing on all beats in four ECG records. All such combinations of records 116, 208, 210, 221, 228, and 233 in the MIT-BIH database were used to evaluate model generalization. We chose this subset of the MIT-BIH database because testing all combinations of records in the entire dataset is impractical and because it was selected as representative of the database by Ham and Han.[6] The averages of these results are given in Table 5, while Table 6 provides confusion matrices of the aggregated results. These show that the CAE provides 1.01% higher overall accuracy and 4.71% higher PVC sensitivity than the SAE. This meets our expectation that a reduced number of trained weights in the autoencoder would improve performance with reduced amounts of training data as well as improve the ability of the detection system to generalize to new data.

Table 5. MIT-BIH Restricted Training Cross-Validation

Architecture	Correct	PVC Sensitivity	PVC Specificity
CAE	87.80	86.56	88.09
SAE	86.79	81.85	87.91

Table 6. Cross-Validation CAE and SAE Confusion Matrices

	CAE		SAE	
	True Normal	True PVC	True Normal	True PVC
Detected Normal	111721	3874	111499	5234
Detected PVC	15109	24956	15331	23596

3.4. *Estimated Parameters and Convergence*

Our convolutional autoencoder architecture used 83.43% fewer network weights due to the weight sharing inherent in convolutional networks. For the 54695 example training set used in 3.1 and 3.2, this resulted in a decrease in the number of training epochs necessary for convergence from 5 to 1.

Table 7. Network Weights

Architecture	Estimated Weights
CAE	1702
SAE	10270

Table 8. MIT-BIH Full Database CAE Performance

Record	Beats	Normal	PVC	Correct	Sensitivity	Specificity
100	1135	1134	1	100.000	100.000	100.000
101	931	931	0	100.000	—	100.000
102	1092	1090	2	100.000	100.000	100.000
103	1040	1040	0	99.904	—	99.904
104	1113	1112	1	100.000	100.000	100.000
105	1285	1272	13	95.642	46.154	96.148
106	1012	673	339	96.542	89.676	100.000
107	1067	1020	47	99.438	87.234	100.000
108	880	873	7	99.432	28.571	100.000
109	1264	1242	22	81.487	77.273	81.562
111	1061	1061	0	99.811	—	99.811
112	1268	1268	0	100.000	—	100.000
113	896	896	0	100.000	—	100.000
114	938	936	2	100.000	100.000	100.000
115	975	975	0	100.000	—	100.000
116	1204	1158	46	99.917	97.826	100.000
117	766	766	0	100.000	—	100.000
118	1138	1130	8	99.385	25.000	99.912
119	992	747	245	100.000	100.000	100.000
121	930	929	1	99.785	0.000	99.892
122	1236	1236	0	100.000	—	100.000
123	758	756	2	100.000	100.000	100.000
124	808	789	19	98.886	52.632	100.000
200	1299	817	482	97.614	93.568	100.000
201	980	860	120	99.388	95.833	99.884
202	1066	1064	2	99.812	50.000	99.906
203	1489	1283	206	97.851	90.777	98.987
205	1326	1280	46	99.623	89.130	100.000
207	929	925	4	87.836	100.000	87.784
208	1476	1024	452	97.900	98.894	97.461
209	1501	1501	0	100.000	—	100.000
210	1323	1212	111	97.279	68.468	99.917
212	1372	1372	0	100.000	—	100.000
213	1624	1517	107	98.153	93.458	98.484
214	1129	1006	123	97.874	81.301	99.901
215	1680	1598	82	98.452	68.293	100.000
217	1103	1037	66	99.547	95.455	99.807
219	1076	1044	32	99.257	75.000	100.000
220	1022	1022	0	100.000	—	100.000
221	1212	1051	161	99.917	99.379	100.000
222	1240	1240	0	100.000	—	100.000
223	1301	985	316	96.772	88.291	99.492
228	1025	877	148	98.829	91.892	100.000
230	1127	1126	1	99.379	100.000	99.378
231	784	784	0	100.000	—	100.000
232	889	889	0	100.000	—	100.000
233	1538	1122	416	98.635	96.154	99.554
234	1375	1372	3	99.709	0.000	99.927
Total	54675	51042	3633	98.548	91.412	99.056

Table 9. MIT-BIH Full Database CAE Disturbed Performance

Record	Beats	Normal	PVC	Correct	Sensitivity	Specificity
100	1135	1134	1	99.912	0.000	100.000
101	931	931	0	100.000	—	100.000
102	1092	1090	2	99.817	50.000	99.908
103	1040	1040	0	100.000	—	100.000
104	1113	1112	1	100.000	100.000	100.000
105	1285	1272	13	93.541	23.077	94.261
106	1012	673	339	89.526	68.732	100.000
107	1067	1020	47	99.157	80.851	100.000
108	880	873	7	99.091	28.571	99.656
109	1264	1242	22	78.006	45.455	78.583
111	1061	1061	0	100.000	—	100.000
112	1268	1268	0	100.000	—	100.000
113	896	896	0	100.000	—	100.000
114	938	936	2	100.000	100.000	100.000
115	975	975	0	100.000	—	100.000
116	1204	1158	46	99.917	97.826	100.000
117	766	766	0	100.000	—	100.000
118	1138	1130	8	99.297	25.000	99.823
119	992	747	245	99.899	99.592	100.000
121	930	929	1	99.785	0.000	99.892
122	1236	1236	0	100.000	—	100.000
123	758	756	2	100.000	100.000	100.000
124	808	789	19	98.020	15.789	100.000
200	1299	817	482	95.766	88.589	100.000
201	980	860	120	97.449	79.167	100.000
202	1066	1064	2	99.906	50.000	100.000
203	1489	1283	206	95.433	76.214	98.519
205	1327	1281	46	99.171	76.087	100.000
207	929	925	4	95.048	100.000	95.027
208	1476	1024	452	96.206	95.354	96.582
209	1501	1501	0	100.000	—	100.000
210	1323	1212	111	93.878	28.829	99.835
212	1372	1372	0	99.927	—	99.927
213	1624	1517	107	97.845	85.047	98.748
214	1129	1006	123	93.711	47.154	99.404
215	1680	1598	82	96.845	35.366	100.000
217	1103	1037	66	98.368	84.848	99.229
219	1076	1044	32	98.792	84.375	99.234
220	1022	1022	0	100.000	—	100.000
221	1212	1051	161	99.752	98.137	100.000
222	1240	1240	0	99.919	—	99.919
223	1301	985	316	87.855	50.949	99.695
228	1025	877	148	97.268	81.081	100.000
230	1127	1126	1	99.734	100.000	99.734
231	784	784	0	100.000	—	100.000
232	889	889	0	100.000	—	100.000
233	1538	1122	416	95.904	85.817	99.643
234	1375	1372	3	99.782	33.333	99.927
Total	54676	51043	3633	97.608	77.814	99.017

4. Discussion

We developed a system for the detection of PVCs in ECG data annotated with beat locations using a CAE. This provided comparable performance to a SAE architecture for the task with reduced training time due to its reduced number of parameters. The CAE provided improvements in the resilience of the PVC detection system to beat detection timing variance and improved detection performance when trained using ECG records from different patients.

Some limitations of this approach to PVC detection include the computational complexity of representation learning methods as compared to manual feature engineering and the lack of direct and unambiguous physical or medical significance for the features extracted by the system. There is also no guarantee that homologous features will be generated by training on different ECG data, which precludes the possibility of retraining the convolutional autoencoder without also retraining the final classifier.

The relatively low number of parameters in our model make it well suited to implementation on the limited hardware available in an applied setting while not relying on potentially unreliable QRS segmentation or features that are difficult to measure or compute in real time. In addition to its advantage in computational expense, the improvement provided by our autoencoder architecture in cross-patient generalization is of significant importance in the application of a PVC detection system to real patients, where it may be impractical or impossible to obtain a sufficient amount of expert-annotated training data.

Based on the performance of this system, we envision the extension of our CAE architecture to facilitate the detection of other arrhythmias in ECG data. Another potential avenue for future work with this autoencoder architecture is to take advantage of its small number of trained parameters to allow the model to be retrained on the spot based on a subset of available annotated ECG records most similar to a sample of the ECG data from the current patient.

References

1. H. J. L. Marriott, G. S. Wagner and D. G. Strauss, *Marriott's Practical electrocardiography* (Wolters Kluwer Health / Lippincott Williams & Wilkins, Philadelphia, 2014).
2. C. L. Stanfield and W. J. Germann, *Principles of human physiology.* (Pearson Benjamin Cummings, San Francisco; London, 2008).
3. J. S. Geddes and H. R. Warner, *Computers and Biomedical Research* **4**, 493 (1971).
4. P. Trahanias, E. Skordalakis and G. Papaconstantinou, *Pattern recognition letters* **9**, 13 (1989).
5. A. E. Zadeh, A. Khazaee and V. Ranaee, *Computer Methods and Programs in Biomedicine* **99**, 179 (August 2010).
6. F. M. Ham and S. Han, *IEEE Transactions on Biomedical Engineering* **43**, 425 (1996).
7. J. Lim, *IEEE Transactions on Neural Networks* **20**, 522 (March 2009).
8. W. Gersch, P. Lilly and E. Dong Jr, *Computers and Biomedical Research* **8**, 370 (1975).
9. S. Yu and K. Chou, *Expert Systems with Applications* **34**, 2841 (May 2008).
10. J. Yang, Y. Bai, G. Li, M. Liu and X. Liu, *Bio-Medical Materials and Engineering* **26**, S1549 (August 2015).
11. Y. Bengio, A. Courville and P. Vincent, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**, 1798 (August 2013).
12. X. Guo, X. Liu, E. Zhu and J. Yin, Deep clustering with convolutional autoencodersOctober 2017.

13. G. B. Moody and R. G. Mark, *IEEE engineering in medicine and biology magazine: the quarterly magazine of the Engineering in Medicine & Biology Society* **20**, 45 (June 2001).
14. F. Chollet *et al.*, Keras `https://keras.io`, (2015).
15. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu and X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems (2015), Software available from tensorflow.org.
16. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, *Journal of Machine Learning Research* **12**, 2825 (2011).
17. C.-Y. Liou, W.-C. Cheng, J.-W. Liou and D.-R. Liou, *Neurocomputing* **139**, 84 (September 2014).
18. D. P. Kingma and J. Ba, Adam: A Method for Stochastic Optimization, in *International Conference on Learning Representations*, December 2014.
19. L. Breiman, *Machine Learning* **45**, 5 (October 2001).

Removing Confounding Factors Associated Weights in Deep Neural Networks Improves the Prediction Accuracy for Healthcare Applications

Haohan Wang[1], Zhenglin Wu[2], Eric P. Xing[1,3]
[1]*School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA*
[2]*School of Information Sciences, University of Illinois Urbana-Champaign Champaign, IL, USA*
[3]*Petuum Inc. Pittsburgh, PA, USA*
E-mail: haohanw@cs.cmu.edu

The proliferation of healthcare data has brought the opportunities of applying data-driven approaches, such as machine learning methods, to assist diagnosis. Recently, many deep learning methods have been shown with impressive successes in predicting disease status with raw input data. However, the "black-box" nature of deep learning and the high-reliability requirement of biomedical applications have created new challenges regarding the existence of confounding factors. In this paper, with a brief argument that inappropriate handling of confounding factors will lead to models' sub-optimal performance in real-world applications, we present an efficient method that can remove the influences of confounding factors such as age or gender to improve the across-cohort prediction accuracy of neural networks. One distinct advantage of our method is that it only requires minimal changes of the baseline model's architecture so that it can be plugged into most of the existing neural networks. We conduct experiments across CT-scan, MRA, and EEG brain wave with convolutional neural networks and LSTM to verify the efficiency of our method.

Keywords: neural networks, healthcare, confounding factor correction

1. Introduction

The increasing amount of data has led healthcare to a new era where the diagnosis can be made directly from raw data such as CT-scan or MRI with data-driven approaches. Machine learning methods, especially deep learning methods, have achieved significant successes in biomedical and healthcare applications, such as classifying lung nodule,[1] breast lesions,[2] or brain lesions[3] from CT-scans, segmentation of brain regions with MRI,[4,5] or emotion classification with EEG data.[6,7]

However, different from how deep learning has revolutionized many other applications, the "black-box" nature of deep learning and the high-reliability requirement of healthcare industry have created new challenges.[8] One of these challenges is about removing the false signals extracted by deep learning methods due to the existence of confounding factors. Acknowledging the recognition mistakes made by neural networks[9–11] and empirical evidence that deep neural networks can learn signals from confounding factors,[12] it is likely that a well-trained deep learning model will exhibit limited predictive performance on external data sets despite its high predictive power on lab collected data sets. The hazard of inappropriate control of

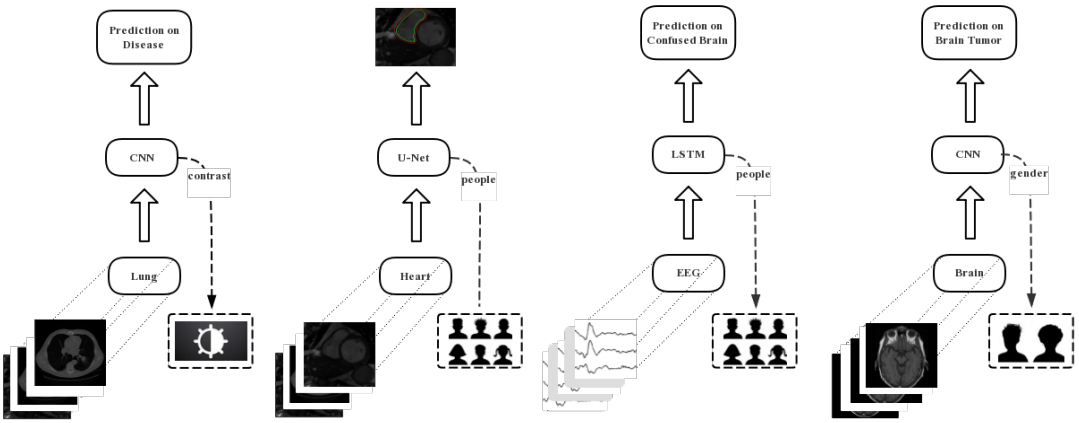


Fig. 1: An illustration of the empirical contribution of this paper. From left to right, 1) lung adenocarcinoma prediction from CT-scan with CNN, where contrast material is the confounding factor, 2) heart right ventricle segmentation from CT-scan with U-net, where subject identification is the confounding factor, 3) students' confusion status prediction from EEG signals with Bidirectional LSTM, where the students' demographic information is the confounding factor, 4) brain tumor prediction from CT-scan/MRA with CNN, where gender associated information is the confounding factor.

confounding factors in healthcare-related science has been discussed extensively,[13–15] but these discussions are mainly in the scope of causal analyses or association studies.

In addition to a very recent result showing that confounding factors can adversely affect the predictive performance of neural network models,[16] we offer a straightforward example as another motivation: a neural network predictive model for Hodgkin lymphoma diagnosis is trained on a data set collected from young volunteers with high predictive performance, but when the model is applied to the entire society, it may report more false positives than expected. One of the reason could be that the gender ratio reverses toward adolescence in Hodgkin lymphoma,[17] and a model trained over data collected from young volunteers is very likely to learn a different gender bias than what is expected in a data collected different age groups. In fact, even if the gender ratio does not change along the aging process, it is still inappropriate for a model to predict based on features related to gender because these features are not directly associated with disease status. As another example, skin cancer[18] and colorectal cancer[19] are also observed with gender bias, and it is already observed that there is a higher false negative rate in colorectal cancer diagnosis for women[19] with traditional methods. Confounding factors do not just exist in the forms of gender. Also, it is observed that other factors, such as age,[20] or demographic information,[21] will affect the model's performance if not handled appropriately. Considering that the generalization theory of neural networks is still an open research topic and people are unsure of how neural networks predict, it is particularly important to design methods to handle the influence of these confounding factors explicitly.

In this paper, inspired by previous de-confounding techniques applied to deep learning models,[12] we propose a Confounder Filtering (CF) method. A distinct advantage of our method

is that CF directly builds upon the original confounded neural network with a minimal change that replaces the original top layer with a layer that predicts the confounding factors. Further, we apply our methods to a broad spectrum of related tasks, such as:

- improved lung adenocarcinoma prediction with convolutional neural networks (CNN) by removing contrast material as confounding factors.
- improved heart right ventricle segmentation with U-net by removing subject identifications as confounding factors.
- improved students' confusion status prediction with Bidirectional LSTM by removing students' demographic information as confounding factors.
- improved brain tumor prediction with CNN by removing gender associated information as confounding factors.

We have observed consistent improvements in predictive performance by removing the confounding factors. These four empirical contributions have been conveniently summarized in Figure 1, which illustrates the experiments we perform in this paper, including the predictive task, the model we use, the data, and the confounding factors.

The remainder of this paper is organized as follows. In Section 2, we first briefly discuss the related work of this paper, mainly in the methodological perspective. In Section 3, we formally introduce our method, namely Confounder Filtering. Then in Section 4, we apply our method to a wide spectrum of experiments to show the effectiveness of our method and report relevant analysis. Finally, we conclude this paper with discussion of limitations and future directions in Section 5.

2. Related Work

The recent boom of deep learning techniques has allowed a large number of neural network methods developed for healthcare applications rapidly. Readers can refer to comprehensive reviews on how the deep learning can be applied to healthcare and biomedical areas.[8,22–24] In this section, we will mainly discuss the related work of our paper in the methodological perspective.

To the best of our knowledge, there are not many deep learning works that control the effects of confounding factors explicitly. Wang *et al* presented a two-phase algorithm named Select-Additive Learning.[12] In the first phase, the model uses information of confounding factors to select which components of the representation learned by neural networks are associated with confounding factors, and then in the addition phase, the algorithm forces the neural networks to discard these components by adding noises. Zhong *et al* also discussed how confounding factors affect the predictive performance of neural networks. They presented an augment training framework that requires little additional computational costs.[25] The idea is to add another neural classifier that predicts confounding factors while predicting original labels, and gradient descent optimizes both of these classifiers. The general additional structure is very similar to the Confounding Filtering method that we are going to present, but our method trains the network in differently so that we can differentiate the weights associated with confounding factors and filter them out explicitly.

In a broader view, correcting confounding factors is related to reducing the representations learned by neural networks through some components of the raw data that are not related to the predictive task. In this perspective, there is a significant amount of neural network methods that can be considered as related work, covering the fields such as domain adaptation,[26] transfer learning,[27,28] and domain generalization.[29] Readers can refer to the survey papers cited and the references therein if interested. Within the scope of this paper, we do not discuss with these methods for two reasons: 1) these methods are not designed for correcting confounding factors explicitly, therefore they may or may not be applicable in this specific situation, 2) even if our CF method behave similar to, or slightly shy of the performance of these methods, there is still a distinct advantage: CF is simple enough to be plugged into any neural networks with almost no changes of the architecture.

3. Confounder Filtering (CF) Method

In this section, we will formally introduce the Confounder Filtering (CF) method. CF method's goal is to reduce the effects of confounders, therefore improves the generalizability of deep neural networks. We first offer an intuitive overview of the main idea of CF, then we formalize our method, which is followed by a discussion of the availability of the implementation.

3.1. *Overview*

CF method is aimed to remove the effects of confounding factors by removing the weights that are associated with them. Therefore, the core step is to identify such weights. We first train a model, namely G, conventionally for the predictive task. Then we replace the top model layer with another classifier that predicts the labels of confounding factors, and we continue to train the model. During this training phase, we keep track of the updates of weights. Finally, we filter out all the weights that are frequently updated during this training phase out of G by replacing these weights with zeros, leading to a new confounder-free model. This process is illustrated in Fig. 2.

3.2. *Method*

We continue to formalize our method. For the convenience of discussion, we split a deep neural network architecture into two components: representation learner component and classification component, denoted by $g(\cdot;\theta)$ and $f(\cdot;\phi)$ respectively, where θ and ϕ stand for the corresponding parameters. Therefore, the complete neural network classifier is denoted as $f(g(\cdot;\theta);\phi)$. Given data $< y, X >$, the classical training process of the neural network is achieved via solving the following equation:

$$\hat{\theta}, \hat{\phi} = \underset{\theta,\phi}{\operatorname{argmin}}\ c(y, f(g(X;\theta);\phi)) \tag{1}$$

where $c(\cdot,\cdot)$ stands for the cost function, with famous examples such as mean-squared-error loss or cross-entropy loss.

Ideally, to effectively remove the effects of confounding factors, a method needs the labels of the confounding factors. In other words, we need data in the form of $< X, y, s >$, where s

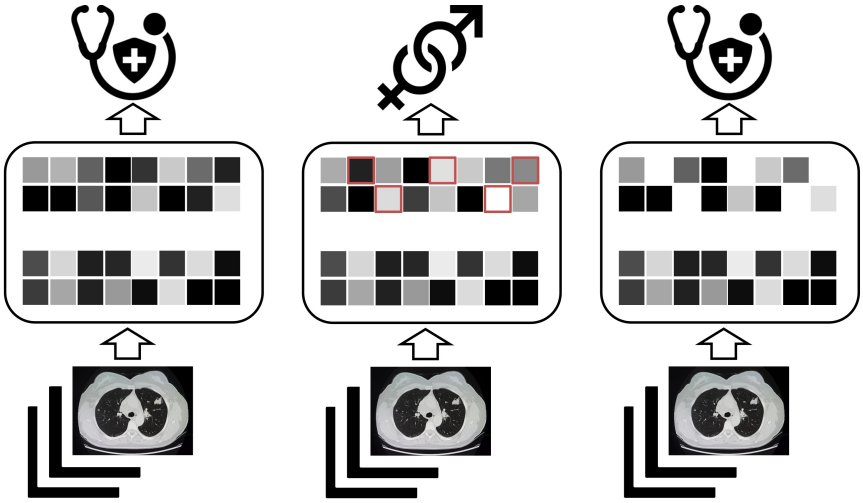


Fig. 2: This figure shows the overview of the CF method. From left to right: 1) Train the neural network conventionally. 2) Train the neural network to predict confounding factors (*e.g.* gender information) and inspect the changes of weights each iteration to locate the ones with largest changes. 3) Remove the located weights, then the model is ready for confounder-free prediction.

stands for the label of the confounding factors (*e.g.* age, gender, physical factors of medical devices *etc.*). This is also required by similar previous work.[12,25] However, our method does not require full correspondence between X, y, and s. For example, later in our experiment, we will show that with two independently collected data sets $< X_1, y_1 >$ and $< X_2, s_2 >$(*i.e.* we only have correspondence between X_1 and y_1, and between X_2 and s_2, but not between y_1 and s_2), we are able to correct the confounding factors between X_1 and y_1 with help of X_2 and s_2. For simplicity, we still present our method with $< X, y, s >$.

After we train the neural network following the conventional manner as showed in Equation 1 with $< X, y >$ and get $\hat{\theta}$ and $\hat{\phi}$, we continue to identify the weights associating with confounding factors through tuning the classification component via $< X, s >$. Formally, we solve the following problem:

$$\tilde{\phi} = \underset{\phi}{\operatorname{argmin}} \; c(s, f(g(X; \hat{\theta}); \phi))$$

During the optimization, our method inspects how the gradient of the cost function with respect to $< X, s >$ updates the previous trained weights (*i.e.* $\hat{\phi}$) with $< X, y >$. For the i^{th} value of ϕ (denoted as ϕ_i), we calculate the frequency of updating it during the entire training process (denoted as π_i). Formally, we have:

$$\pi_i = \frac{1}{n} \sum_{t=1}^{n} |\Delta \phi_{i,t}|$$

where n is the number of total steps, t stands for the index of step.

Further, we construct a masking matrix/tensor M of the same shape as ϕ, and M_i is constructed according to π_i. For example, common choices could be either through a Bernoulli

sampling

$$M_i = \text{Ber}(\pi_i)$$

or a straightforward thresholding procedure:

$$M_i = \begin{cases} 0, & \pi_i > \tau \\ 1, & \text{otherwise} \end{cases}$$

In the following experiment, we choose to use the thresholding procedure with τ, whose value lies between top 20% and top 25% of π_i's values.

Finally, we have $\hat{\phi}' = \hat{\phi} \otimes M$, where $\otimes$ stands for element-wise product, and the final trained neural network after confounding factor associated weights filtered out is as following:

$$f(g(X; \hat{\theta}); \hat{\phi}')$$

which is ready for confounder-free prediction.

3.3. *Availability*

The implementation of our method in TensorFlow is available online[a] with a simple example that trains a CNN for Cifar10 dataset, onto which we add some image patterns as confounding factors. Users can follow the online instruction to apply CF to their own customized neural networks.

4. Experiments

In this section, we will verify the performance of our CF method on four different tasks by adding CF towards the current baseline models. For each task, we will first introduce the data set, and then introduce the methods we compare and the results. After discussions of these four tasks, we will introduce some analyses of the model behaviors to further validate the performance of our method.

4.1. *lung adenocarcinoma prediction*

4.1.1. *Data*

We construct a data set to test the model performance in classifying adenocarcinomas and healthy lungs from CT-scans. Our experimental data set is a composition of three data sets:

- **Data Set 1:** The CT-images from healthy people are collected from ELCAP Public Lung Image Database[b]. The CT scans have obtained in a single breath hold with a 1.25 mm slice thickness that consists of 1310 DICOM images from 25 persons.
- **Data Set 2:** The CT-scans of diseased lungs are collected from 69 different patients by Grove *et al.*[30] These scans are diagnostic contrast-enhanced CT scans, being done at diagnosis and prior to surgery and slice thickness at variable from 3 to 6 mm.

[a]https://github.com/HaohanWang/CF
[b]http://www.via.cornell.edu/lungdb.html

- **Data Set 3:** Since these two data sets are collected differently, and one of them is a collection of contrast-enhanced CT scans. The contrast material will likely serve as the confounding factor in prediction. To correct the confounding factor. We noticed a processed version[c] of **Data Set 2**, which consists of explicit labels of contrast information. The data set contains 475 series from 69 different patients selected 50% with contrast and 50% without contrast.

Therefore, we use the 1290 healthy images from 20 persons in **Data Set 1** and 1214 diseased lung images from 61 patients in **Data Set 2** as the training set, and the rest from these two data sets as the testing set. We use the images from **Data Set 3** with corresponding contrast labels to correct confounding factors.

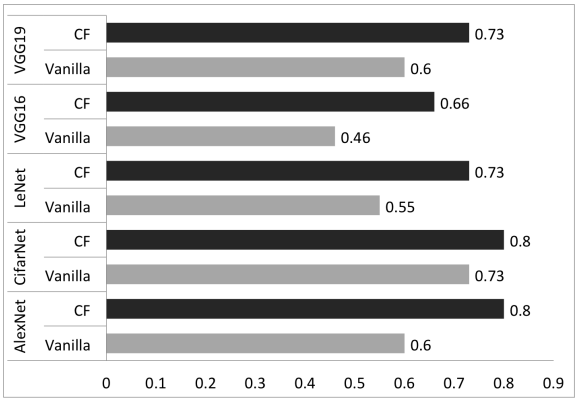


Fig. 3: Prediction accuracy of CNN in comparison with CF-CNN

4.1.2. *Results*

We experiment with the most popular architectures of CNNs, including AlexNet,[31] CifarNet,[32] LeNet,[33] VGG16,[34] and VGG19.[34] We first sufficiently train these baseline models with appropriate learning rate until the training accuracy converges, and then use our CF method to correct the confounding factors. We test the prediction accuracy of both vanilla CNNs and CF-improved CNNs. Fig. 3 shows the results. We can see that CF can consistently improve the predictive results over a variety of different CNNs.

4.2. *Segmentation on right ventricle(RV) of Heart*

4.2.1. *Data*

The data set[35] contains 243 physician-segmented CT images (216×256 pixels) from 16 patients. Data augmentation techniques, such as random rotations, translations, zooms, shears and elastic deformations (locally stretch and compress the image), are used to increase the number of samples. More information regarding the data set, including how the training/testing data sets are split, can be found online[d].

4.2.2. *Results*

The main baseline in this experiment is U-net, which is a convolutional network architecture for fast and precise segmentation of images. Previous experiments show that U-net can behave well

[c]https://www.kaggle.com/kmader/siim-medical-images/home
[d]https://blog.insightdatascience.com/heart-disease-diagnosis-with-deep-learning-c2d92c27e730

even with a small dataset.[36] We first test U-net following previous setting[35] and interestingly, we achieve a higher accuracy that what was reported. Vanilla U-net achieves an accuracy of 0.9477. Then, we use CF method to remove the subject identities as confounding factors and improve the accuracy from 0.9477 to **0.9565**.

4.3. *Students' confusion status prediction*

4.3.1. *Data*

The data set[37] contains EEG brainwave data from 10 college students while they watch MOOC video clips[e]. The EEG data is collected rom MindSet equipment wore by college students when watch ten video clips, five out of which are confusing ones. The students' identities are considered as confounding factors in this experiment.

Following previous work,[38] we normalize the training data in a feature-wise fashion (*i.e.*, each feature representation is normalized to have a mean of 0 and standard deviation of 1 across each batch of samples). The batch size is set to 20.

Table 1: Comparison with average accuracy for 5-fold cross validation[38]

Methods	Accuracy(%)
SVM	67.2
KNN	51.9
CNN	64.0
DBN	52.7
RNN-LSTM	69.0
BiLSTM	73.3
CF-BiLSTM	**75.0**

4.3.2. *Results*

We use the state-of-the-art method applied to this data set,[38] namely a Bidirectional LSTM, as the baseline method to compare with. The model is configured as following: the LSTM layer has 50 units, with *tanh* as activation function. The output is connected to a fully connected layer with a sigmoid activation. We compare five-fold-cross-validated results from CF-improved Bidirectional LSTM with results reported previously.[38] The results are shown in Table 1. As we can see, CF method helps improve the predictive performance once plugged in.

4.4. *Brain tumor prediction*

4.4.1. *Data*

We construct another data set for the last experiment of this paper. We test our method in predicting brain tumors with MRA scans of healthy brain[f] and CT-scans with tumor brain.[39] The healthy data set consists of images of the brain from 100 healthy subjects, in which 20 patients were scanned per decade and each group are equally divided by sex. The tumor data set is collected with 120 patients. The gender information is regarded as confounding factors in this experiment.

[e]https://www.kaggle.com/wanghaohan/confused-eeg/home
[f]http://insight-journal.org/midas/community/view/21

4.4.2. *Results*

Similar to the lung adenocarcinoma prediction experiment, we compare with the set of popular CNNs. The results are shown in Fig. 4. As we can see that, CF helps improve the prediction performance in most cases, except that in the VGG19 cases, when the model's performance deteriorates after CF is plugged in.

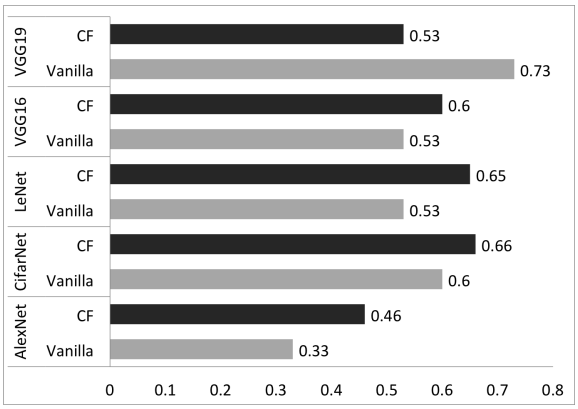


Fig. 4: Prediction accuracy of CNN in comparison with CF-CNN

4.5. *Analyses of the method behaviors*

To further understand the process of CF in identifying the weights that are associated with the confounding factors. We inspect how the weights are updated during the training process and visualize which part of the input data is related to confounding factors.

Fig. 5(a) visualizes the weights during each epoch. The figure splits into two panels, and the left panel is for lung adenocarcinoma prediction experiment, and the right panel is for brain tumor prediction experiment. The figure only shows eight weights of the top layer (in a 4×2 rectangle), and visualizes how the weights in the layer change as the training epoch increases. This figure visualizes 96 epochs for lung adenocarcinoma prediction and brain tumor prediction each. The blue dots visualize the weights when the model is trained during the first phase, and the green dots visualize the weights when the model is trained in the second phase for prediction confounding factors. The darker each dot is, the more frequent it gets updated in that epoch. As we can see, for the same 4×2 layer, the frequencies of the weights get updated are different between the training during the first phase and training during the second phase. This differences of updating frequencies verify the primary assumption of our method, that the weights associated with the task and the weights associated with the confounding factors are different. Therefore, we can remove the effects of confounding factors by removing the weights associated with them.

Further, we try to investigate which parts of the input data are corresponding to the confounding factors. With the help of Deep Feature Selection[40] method, we select the pixels of the image that are associated with the confounding factors. Fig 5 visualizes these pixels with yellow dots. From left to right, these four images are examples for healthy lung, diseased lung, healthy brain, tumorous brain respectively. Interestingly, we do not see clear patterns on the images that are related to the confounding factors. This observation further verify the importance of our CF method because these results indicate that it is barely possible to first exclude the information from raw images by conventional methods since these yellow dots do not form into any clear pattern.

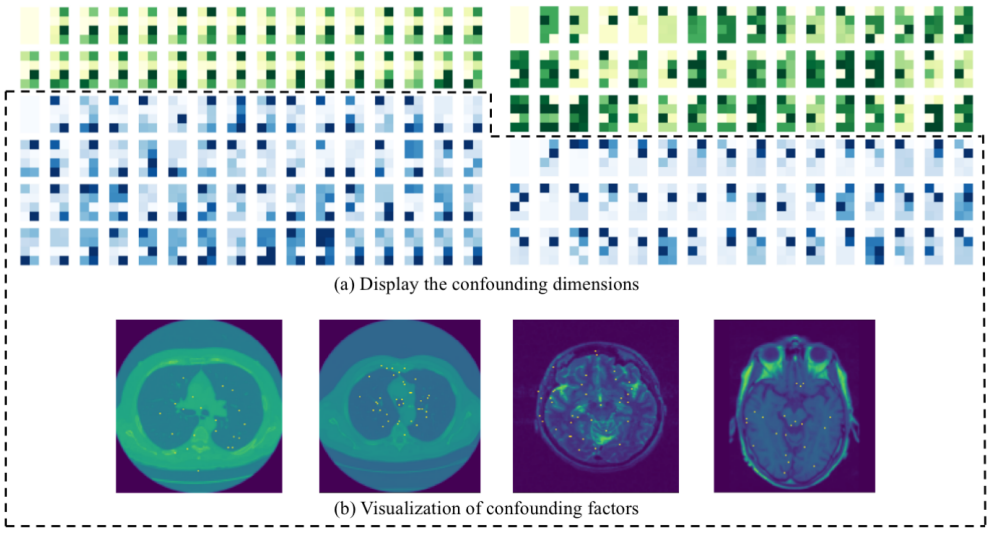


Fig. 5: (a) Display of trained weights and (b) the visualization of confounding factors.

5. Conclusion

In this paper, we proposed a straightforward method, named Confounder Filtering, which aims to reduce the effects of confounders and improve the generalizability of deep neural networks, to achieve a confounding-factor-free predictive model for healthcare applications. One distinct advantage of our method is that we only require minimal changes to the existing network model to adopt our method. There are still limitations of our method: despite our method only requires a minimal changes of the network architecture, it needs a repeated training process (the second phase training with confounding factors). Another limitation is that our method still requires the switching of the top classification layer from a label predictor to a confounder predictor, which may lose the one-to-one correspondence of weights at the top layer. In the future, in the methodological perspective, we look forward to further improving the training process of our method. On the practical side, as we have released our code, we hope to help the community to increase the performance of other predictive models for healthcare application by removing the confounding factors.

6. Acknowledgement

The authors would like to thank Mingze Cao and Yin Chen for discussions and creation of Fig 1 and Fig 5. This work is funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. This work is also supported by the National Institutes of Health grants R01-GM093156 and P30-DA035778. The MR brain images from healthy volunteers used in this paper were collected and made available by the CASILab at The University of North Carolina at Chapel Hill and were distributed by the MIDAS Data Server at Kitware, Inc

References

1. K.-L. Hua, C.-H. Hsu, S. C. Hidayati, W.-H. Cheng and Y.-J. Chen, Computer-aided classification of lung nodules on computed tomography images via deep learning technique, *OncoTargets and therapy* **8** (2015).
2. J.-Z. Cheng, D. Ni, Y.-H. Chou, J. Qin, C.-M. Tiu, Y.-C. Chang, C.-S. Huang, D. Shen and C.-M. Chen, Computer-aided diagnosis with deep learning architecture: applications to breast lesions in us images and pulmonary nodules in ct scans, *Scientific reports* **6**, p. 24454 (2016).
3. X. W. Gao, R. Hui and Z. Tian, Classification of ct brain images based on deep learning networks, *Computer methods and programs in biomedicine* **138**, 49 (2017).
4. A. Işın, C. Direkoğlu and M. Şah, Review of mri-based brain tumor image segmentation using deep learning methods, *Procedia Computer Science* **102**, 317 (2016).
5. F. Milletari, S.-A. Ahmadi, C. Kroll, A. Plate, V. Rozanski, J. Maiostre, J. Levin, O. Dietrich, B. Ertl-Wagner, K. Bötzel *et al.*, Hough-cnn: deep learning for segmentation of deep brain regions in mri and ultrasound, *Computer Vision and Image Understanding* **164**, 92 (2017).
6. S. Jirayucharoensak, S. Pan-Ngum and P. Israsena, Eeg-based emotion recognition using deep learning network with principal component based covariate shift adaptation, *The Scientific World Journal* **2014** (2014).
7. W.-L. Zheng, J.-Y. Zhu, Y. Peng and B.-L. Lu, Eeg-based emotion classification using deep belief networks, in *Multimedia and Expo (ICME), 2014 IEEE International Conference on*, 2014.
8. R. Miotto, F. Wang, S. Wang, X. Jiang and J. T. Dudley, Deep learning for healthcare: review, opportunities and challenges, *Briefings in bioinformatics* (2017).
9. C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow and R. Fergus, Intriguing properties of neural networks, *arXiv preprint arXiv:1312.6199* (2013).
10. A. Nguyen, J. Yosinski and J. Clune, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.
11. H. Wang, B. Raj and E. P. Xing, On the origin of deep learning, *arXiv preprint arXiv:1702.07800* (2017).
12. H. Wang, A. Meghawat, L. P. Morency and E. P. Xing, Select-additive learning: Improving generalization in multimodal sentiment analysis, in *IEEE International Conference on Multimedia and Expo*, 2017.
13. M. A. Brookhart, T. Stürmer, R. J. Glynn, J. Rassen and S. Schneeweiss, Confounding control in healthcare database research: challenges and potential approaches, *Medical care* **48**, p. S114 (2010).
14. A. C. Skelly, J. R. Dettori and E. D. Brodt, Assessing bias: the importance of considering confounding, *Evidence-based spine-care journal* **3**, 9 (2012).
15. M. Nørgaard, V. Ehrenstein and J. P. Vandenbroucke, Confounding in observational studies based on large health care databases: problems and potential solutions–a primer for the clinician, *Clinical epidemiology* **9**, p. 185 (2017).
16. J. R. Zech, M. A. Badgeley, M. Liu, A. B. Costa, J. J. Titano and E. K. Oermann, Confounding variables can degrade generalization performance of radiological deep learning models, *arXiv preprint arXiv:1807.00431* (2018).
17. M. T. Dorak and E. Karpuzoglu, Gender differences in cancer susceptibility: an inadequately addressed issue, *Frontiers in genetics* **3**, p. 268 (2012).
18. D. N. Syed and H. Mukhtar, Gender bias in skin cancer: role of catalase revealed, *Journal of Investigative Dermatology* **132**, 512 (2012).
19. S.-E. Kim, H. Y. Paik, H. Yoon, J. E. Lee, N. Kim and M.-K. Sung, Sex-and gender-specific disparities in colorectal cancer risk, *World journal of gastroenterology: WJG* **21**, p. 5167 (2015).

20. R. Guerreiro and J. Bras, The age factor in alzheimers disease, *Genome medicine* **7**, p. 106 (2015).
21. C. Fincher, J. E. Williams, V. MacLean, J. J. Allison, C. I. Kiefe and J. Canto, Racial disparities in coronary heart disease: a sociological view of the medical literature on physician bias., *Ethnicity & disease* **14**, 360 (2004).
22. C. Angermueller, T. Pärnamaa, L. Parts and O. Stegle, Deep learning for computational biology, *Molecular systems biology* **12**, p. 878 (2016).
23. D. Ravı, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo and G.-Z. Yang, Deep learning for health informatics, *IEEE journal of biomedical and health informatics* **21**, 4 (2017).
24. T. Yue and H. Wang, Deep learning for genomics: A concise overview, *arXiv preprint arXiv:1802.00810* (2018).
25. Y. Zhong and G. Ettinger, Enlightening deep neural networks with knowledge of confounding factors, in *Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on*, 2017.
26. M. Wang and W. Deng, Deep visual domain adaptation: A survey, *Neurocomputing* (2018).
27. K. Weiss, T. M. Khoshgoftaar and D. Wang, A survey of transfer learning, *Journal of Big Data* **3**, p. 9 (2016).
28. S. Moon, S. Kim and H. Wang, Multimodal transfer deep learning with applications in audio-visual recognition, *arXiv preprint arXiv:1412.3121* (2014).
29. K. Muandet, D. Balduzzi and B. Schölkopf, Domain generalization via invariant feature representation, in *International Conference on Machine Learning*, 2013.
30. O. Grove, A. E. Berglund, M. B. Schabath, H. J. Aerts, A. Dekker, H. Wang, E. R. Velazquez, P. Lambin, Y. Gu, Y. Balagurunathan *et al.*, Quantitative computed tomographic descriptors associate tumor shape complexity and intratumor heterogeneity with prognosis in lung adenocarcinoma, *PloS one* **10**, p. e0118261 (2015).
31. A. Krizhevsky, I. Sutskever and G. E. Hinton, Imagenet classification with deep convolutional neural networks, in *International Conference on Neural Information Processing Systems*, 2012.
32. J. Hosang, M. Omran, R. Benenson and B. Schiele, Taking a deeper look at pedestrians, in *Computer Vision and Pattern Recognition*, 2015.
33. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, Imagenet large scale visual recognition challenge, *International Journal of Computer Vision* **115**, 211 (2015).
34. K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, *Computer Science* (2014).
35. C.-H. Yee, Heart disease diagnosis with deep learning: State-of-the-art results with 60x fewer parameters `https://blog.insightdatascience.com/heart-disease-diagnosis-with-deep-learning-c2d92c27e730`.
36. O. Ronneberger, P. Fischer and T. Brox, U-net: Convolutional networks for biomedical image segmentation, in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.
37. H. Wang, Y. Li, X. Hu, Y. Yang, Z. Meng and K.-m. Chang, Using eeg to improve massive open online courses feedback interaction., in *AIED Workshops*, 2013.
38. Z. Ni, A. C. Yuksel, X. Ni, M. I. Mandel and L. Xie, Confused or not confused?: Disentangling brain activity from eeg data using bidirectional lstm recurrent neural networks, in *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics*, ACM-BCB '17 (ACM, New York, NY, USA, 2017).
39. L. Scarpace *et al.*, Data from rembrandt. the cancer imaging archive (2015).
40. Y. Li, C.-Y. Chen and W. W. Wasserman, Deep feature selection: Theory and application to identify enhancers and promoters., in *RECOMB*, 2015.

DeepDom: Predicting protein domain boundary from sequence alone using stacked bidirectional LSTM

Yuexu Jiang, Duolin Wang, Dong Xu
Department of Electrical Engineering and Computer Science, Bond Life Sciences Center, University of Missouri, Columbia, Missouri 65211, USA
Email: xudong@missouri.edu

Protein domain boundary prediction is usually an early step to understand protein function and structure. Most of the current computational domain boundary prediction methods suffer from low accuracy and limitation in handling multi-domain types, or even cannot be applied on certain targets such as proteins with discontinuous domain. We developed an *ab-initio* protein domain predictor using a stacked bidirectional LSTM model in deep learning. Our model is trained by a large amount of protein sequences without using feature engineering such as sequence profiles. Hence, the predictions using our method is much faster than others, and the trained model can be applied to any type of target proteins without constraint. We evaluated DeepDom by a 10-fold cross validation and also by applying it on targets in different categories from CASP 8 and CASP 9. The comparison with other methods has shown that DeepDom outperforms most of the current *ab-initio* methods and even achieves better results than the top-level template-based method in certain cases. The code of DeepDom and the test data we used in CASP 8, 9 can be accessed through GitHub at https://github.com/yuexujiang/DeepDom.

Keywords: protein domain; domain boundary prediction; deep learning; LSTM.

1. Introduction

Protein domains are conserved parts on protein sequences and structures that can evolve, function, and exist independently of the rest of the protein chain. While some proteins have only one domain, many proteins contain more than one domain. Molecular evolution uses domains as building blocks and these may be recombined in different arrangements to create proteins with different functions[1]. Thus, accurate identification of protein domains is crucial to understanding protein function and evolutionary mechanisms. Currently, the most reliable characterization of protein domain is through experimental methods. However, due to the large amount of data being generated by high-throughput technologies nowadays, it is impossible to manually identify domains for these proteins, not to mention that the experimental methods are time consuming and costly. Thus, computational domain prediction methods are in highly demand.

A variety of computational methods for protein domain prediction have been developed, and they can be roughly categorized as either template-based methods or *ab-initio* methods. The principle of most template-based methods is to find homologous sequences that have known domain information by sequence alignments and then map the domain information from these sequences to the query protein sequence. The methods belonging to this category are Pfam[2], CHOP[3], FIEFDOM[4], and ThreaDom[5]. A variation of template-based methods is to use 3D structural models to assist protein domain prediction, e.g. SnapDRAGON[6] and RosettaDom[7]. These methods first construct a tertiary structure model of the target using structural templates.

Domains are then assigned by domain parser tools from the constructed 3D model. The template-based methods can have a high prediction accuracy when close templates are found; however, their prediction performance may drop dramatically if there is no highly similar sequence in domain databases.

Ab-initio methods are more widely used than template-based methods, since these template-free methods can be applied to any protein. They are mainly statistical and machine learning algorithms that train models using the known protein domain boundary information stored in databases such as CATH[8] and SCOP[9]. Some of the representative methods in this category are PPRODO[10], DOMPro[11], PRODOM[12], DomCut[13], ADDA[14], DomNet[15], DROP[16], DOBO[17], and EVEREST[18]. Compared with the template-based approaches, the prediction accuracy of the *ab-initio* methods is low. This is mainly because these methods suffer from the weak domain boundary information in sequence, even after a deliberate but tedious process of feature extraction.

Deep learning is currently the most attractive area in machine learning. Among the various architectures of deep learning, Long Short Term Memory (LSTM)[19] has been successfully applied to problems such as speech recognition, language modeling, translation, image captioning[20-22]. Essential to these successes is its chain-like structure that can capture the sequential information, and its repeating module designed to avoid the vanishing gradient problem that the original Recurrent Neural Network (RNN) suffers[23]. Here, we consider protein sequences as strings of information just like language. Thus, in this paper we propose a new *ab-initio* protein domain boundary prediction method using LSTM. We assume that the signal pattern from a domain boundary region is different from the signals generated from other regions. So, we made each LSTM layer in our deep learning architecture bidirectional to capture the sequential information not just from the N-terminal side of the domain boundary region but also from the C-terminal side. Then we stack multiple such layers together to fit a high-order non-linear function in order to predict the complex domain boundary signal pattern. Instead of paying much effort in feature engineering on a small dataset, which is what traditional machine learning methods do, we train our LSTM model on a big dataset to learn data representations automatically. To the best of our knowledge, this is the first deep learning method applied on the protein domain boundary prediction problem.

2. METHODS

2.1 Data Set Preparation

We collected 456,128 proteins with domain boundary annotations in the CATH database (version 4.2). All the sequences of corresponding proteins were downloaded from the Uniprot database[24]. Then we used CD-HIT[25] to cluster similar proteins into clusters that meet our pre-defined similarity threshold (40%). The representative sequence in each cluster was extracted to form a non-redundant dataset in which every pair of proteins has sequence identity less than 40%[26]. This threshold instead of a lower number makes sure enough data were remained for deep learning. We further excluded proteins with sequence length less than 40 residues, since it needs at least 40 residues for a domain boundary signal to be significant according to Ref. [17]. The final dataset

contains 57,887 proteins. We used 10-fold cross validation to evaluate our model. In each fold, 90% proteins were used to train a model, the remaining 10% proteins were used for testing.

2.2 Input Encoding

Before using our data to train the model, we need to understand the distribution of the data. Figure 1 shows some statistics of our data, which let us believe that encoding the entire sequence for each protein was probably not a good idea. The first reason is that it introduces bias. When there is only one domain on a protein, the boundaries of the only domain are always near the protein's two termini. As shown in Figure 1(A), proteins with one domain represent the majority of the data, and this would make our model over-memorize this pattern and favor the prediction as one domain, which results in poor performance for multi-domain cases. The second reason is as illustrated in Figure 1(B), that proteins with different number of domains have different length distributions. When encoding the entire protein sequence using a dynamic length, we cannot train the model in batch, which is much faster to handle big data set. So, we decided to use a sliding window strategy independent of the protein length to encode an input sequence into equal-length fragments. And we use symbol "-" for padding when the last fragment is shorter than window size. After experiments, we determined the best combination of window size and stride is 200 residues and 80 residues.

Next, we need to encode each residue in every fragment. According to the work of Venkatarajan and Braun[27], a comprehensive list of 237 physical-chemical properties for each amino acid was compiled from the public databases. Their study showed that the number of properties could be reduced while retaining approximately the same distribution of amino acids in the feature space. Particularly, the correlation coefficient between the original and regenerated distances is more than 99% when using the first five eigenvectors. Thus, we used five numerical descriptors to represent each amino acid for computational efficiency while maintaining almost all the information at the same time. We also added the sixth encoding dimension as the padding indicator. For all the 20 types of amino acids, their sixth code is zero. The symbol "-", as the sixth

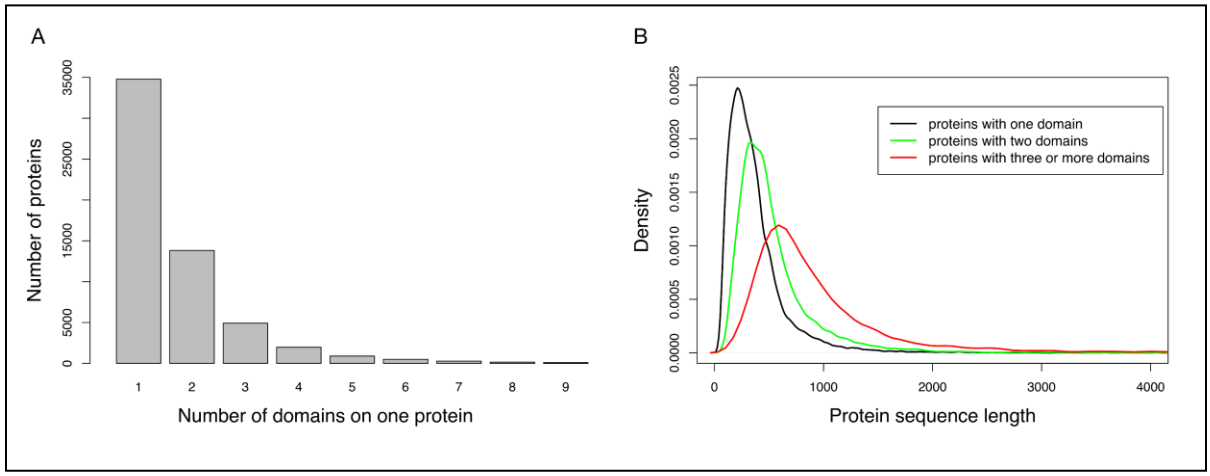


Figure 1. (A) The distribution of proteins with different numbers of domains. (B) The distribution of protein sequence lengths in different categories.

code with value 1, indicates a padding residue, and its first five codes are all zeros. Thus, for each input fragment, its coding dimension is 200 by 6.

For model training, we also need to encode the label for each residue. We derive the protein domain boundary annotation from the CATH database, and follow the convention that considers a residue as positive if it is within ± 20 residues of the true boundary. Thus, the coding dimension for output labels is 200 by 3. The three values represent the probability of a residue being a positive (within the true boundary), negative (outside the true boundary), and padding residue, respectively.

2.3 Model Architecture

Our deep learning architecture is shown in Figure 2. The bidirectional design in each middle layer captures the information from residues before and after a protein domain boundary. We stacked four such layers to capture the high order non-linear features that can detect complex boundary patterns or weak signals. Each neuron in the hidden layers is an LSTM unit.

The key to LSTM is the cell state C that runs through the entire chain. An LSTM unit has the ability to remove or add information to the cell state by a regulation structure called gate. Firstly, an LSTM unit uses its "forget gate" to decide what information to discard from the cell state. It takes the output h_{t-1} from the previous unit and the current input x_t as the input of a sigmoid function to produce a number between 0 and 1 for each number in the cell state. A 1 means completely keeping the value while a 0 means completely removing it. The formulas for the forget gate is shown as Eq. (1).

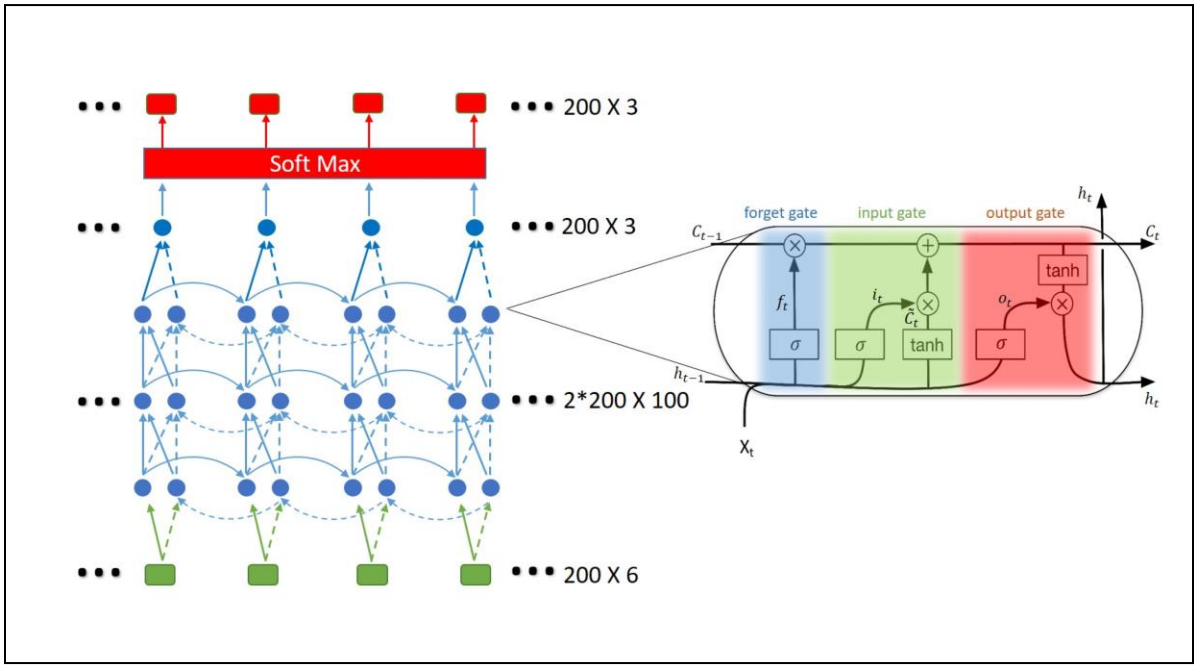


Figure 2. The stacked bidirectional LSTM model. Green boxes represents the input layer. Red boxes represents the output layer. Each box represents a residue. Blue dots form the bi-directional hidden layers. Signals from left to right are represented by solid arcs, while dashed arcs represent signals from the reverse direction. Each dot represents an LSTM unit. A magnified LSTM unit is shown. Its different gates are highlighted with different colors. At the end of the model, a Softmax layer is added to scale the output value with a sum of 1 so that they can be interpreted as probabilities.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

where W_f and b_f are the weight matrix and bias for the forget gate layer. Next, a tanh layer creates a new candidate input vector. It will be performed a pointwise product with a sigmoid layer called the "input gate" to decide which values to add to the cell state. The formula for candidate input creation and the input gate are shown as Eq. (2) and Eq. (3), respectively.

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

where W_C and W_i are weight matrix for the tanh layer and the input gate layer, respectively. b_C and b_i are bias for the tanh layer and the input gate layer, respectively. Then the LSTM unit can update the old cell state C_{t-1} into the new cell state C_t by Eq. (4).

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

Finally, the cell state goes through a tanh layer to scale the values between -1 and 1. The scaled cell state will be filtered by a sigmoid layer called "output gate" to decide which values to output. The formulas for output gate definition and the current output are shown as Eq. (5) and Eq. (6), respectively.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

The ability of avoiding vanishing gradient is mainly owing to the design of forget gate in LSTM. Thus, if a protein domain boundary prediction depends on some signals from remote residues, our model can be trained to set those forget gates' values as 1 on informative positions and let the far, weak but informative signal propagate far without significant loss.

2.4 Evaluation criteria

We used prediction precision, recall and Matthew's correlation coefficient (MCC) to evaluate our method and compare with others'. The definitions of precision, recall, MCC are listed in Eq. (7), Eq. (8) and Eq. (9), respectively:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (7)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (8)$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(FP+TN)(TN+FN)}} \quad (9)$$

where TP, FP, TN, FN are true positive, false positive, true negative and false negative prediction, respectively. When a residue has a predicted probability of being within a domain boundary region higher than a cutoff, we checked its surrounding ± 20 residues to see if there is a recorded domain boundary in the CATH database for the protein. If yes, then we have a true positive, otherwise it is a false positive. On the contrary, when there is a residue our model predicted it being outside of domain boundary regions, we checked its surrounding ± 20 residues to see if there is a recorded

Table 1. Prediction performance in different experiment designs

Window size	80			100			200		
Stride	20	40	80	20	40	80	20	40	80
Experiment ID	1	2	3	4	5	6	7	8	9
Precision_d1	0.572	0.625	**0.626**	0.609	0.622	0.588	0.465	0.547	0.618
Recall_d1	0.493	0.498	0.447	0.486	0.513	0.529	**0.602**	0.582	0.584
MCC_d1	0.442	0.478	0.450	0.462	0.485	0.472	0.415	0.471	**0.520**
Precision_d2	0.608	**0.655**	0.650	0.652	0.653	0.623	0.496	0.576	0.654
Recall_d2	0.361	0.338	0.291	0.346	0.366	0.365	**0.473**	0.443	0.426
MCC_d2	0.361	0.374	0.341	0.377	0.391	0.372	0.341	0.386	**0.426**
Precision_d3+	0.639	0.670	0.661	0.675	0.668	0.629	0.543	0.598	**0.669**
Recall_d3+	0.357	0.297	0.245	0.315	0.330	0.310	**0.453**	0.418	0.381
MCC_d3+	0.360	0.340	0.301	0.354	0.360	0.326	0.343	0.367	**0.391**
Precision_ALL	0.601	**0.644**	0.641	0.637	0.643	0.607	0.496	0.570	0.641
Recall_ALL	0.409	0.382	0.332	0.386	0.407	0.406	**0.513**	0.486	0.468
MCC_ALL	0.392	0.402	0.369	0.401	0.416	0.394	0.370	0.412	**0.450**

domain boundary in the CATH database for the protein. If yes, then we have a false negative; otherwise it is a true negative.

3. RESULTS AND DISCUSSION

3.1 Parameter configuration experiments on test data

We have done a series of experiments with different window sizes and stride values to determine the best combination of these two parameters. The prediction performance of each experiment design is listed in Table 1. And we presented the results separately based on the number of domains that a protein has. Each value is the result after the 10-fold cross validation. Note that in

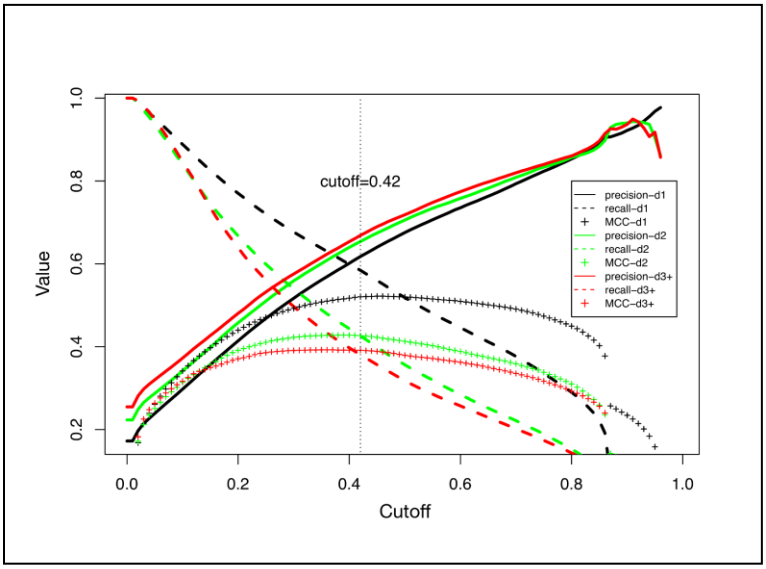


Figure 3. Illustration of the prediction precision, recall and MCC as a function of the decision threshold when the window size=200 and stride=80. The results are based on a 10-fold cross validation.

Experiment 3, we considered the situation that there is no overlap between windows. Under each experiment design (one column) in Table 1, we only presented the result that had the highest MCC-ALL at a certain threshold. We also conducted experiments using sliding window of 300 residues. However, the improvement for MCC-ALL is not significant (around 0.01) compared with cases when window size is 200 residues. So, we believe 200 is enough. As shown in Table 1, the highest MCC-ALL, also the overall best prediction performance is achieved when the sliding window size equals to 200 residues and the stride value equals to 80 residues. Figure 3 illustrates a plot of the precision, recall and MCC as functions of the decision threshold when using the optimum window size and stride value. The threshold at which the highest MCC-ALL reached is 0.42, and hence we used this value as the default threshold.

3.2 Comparison with Other Domain Boundary Predictors

To perform a fair comparison with other methods on a benchmark dataset, we tested our method on the proteins in the Critical Assessment of Techniques for Protein Structure Prediction (CASP). The definitions of domain boundaries on target proteins are provided by the CASP protein domain prediction contest sessions. Based on the categories those target proteins belong to, we conducted several experiments accordingly. In each experiment, the proteins that have a 40% or higher identity with any target protein were excluded from our training dataset.

3.2.1 Free modeling targets from CASP 9

Free modeling (FM) targets are proteins without any homologous templates. These targets are often regarded as “hard cases”, since their predictions usually had poor performance. We selected all the 22 FM targets in CASP 9 and applied different methods to predict their domain boundaries. By comparing the results in the two categories in Table 2, we found most template-based methods suffered a significant decrease in both precision and recall for FM targets. ThreaDom is currently the top 1 templated-based method using multiple threading alignments to extract protein domain boundary information. For FM targets, ThreaDom identifies multiple alignments or super-secondary structure segments from weakly homologous templates, then a domain conservation score profile extracts consensus information between the domain structure and alignment gaps. This way, ThreaDom maintained a good precision for FM targets. Our *ab-initio* method DeepDom achieved the overall best prediction results for FM targets, with the same precision as ThreaDom but higher recall. All the results by different methods are listed in Table 2, where some of them were generated from the tools provided and others were collected from Ref. [5] and Ref. [17], since they used the same data.

3.2.2 Multi-domain targets from CASP 9

We also selected all the 14 multi-domain targets from CASP 9 with the constraint that every domain on one protein must be continuous, since most other methods can only handle multi-domain targets of this kind. For this category, template-based methods generally have better results. ThreaDom achieved the overall best prediction performance. But DeepDom is still the best among ab-initio methods and also competitive with the template-based methods, as shown in Table 2.

Table 2. Comparison results from different methods on two category targets in CASP 9 contest

Category	Predictor	CASP9 protein boundary prediction	
		Precision	Recall
FM	DeepDom	**0.882**	0.468
	ThreaDom	**0.882**	0.455
	Pfam	0.323	**0.485**
	FIEFDom	0.231	0.182
	DomPro	0.500	0.182
	PPRODO	0.333	0.485
	DROP	0.429	0.182
Multi-Domain	DeepDom	0.689	0.441
	ThreaDom	**0.764**	0.534
	Pfam	0.500	**0.548**
	FIEFDom	0.340	0.233
	DomPro	0.500	0.140
	PPRODO	0.500	0.520
	DROP	0.679	0.260

3.2.3 Discontinuous domain target from CASP 8

Some protein domains consist of several separated segments. The prediction of such discontinuous domain is still an unsolved problem. Most mentioned methods above have been explicitly designed to handle domains without discontinuous segments, despite the fact that discontinuous domain is important in protein structural determination and function annotations.

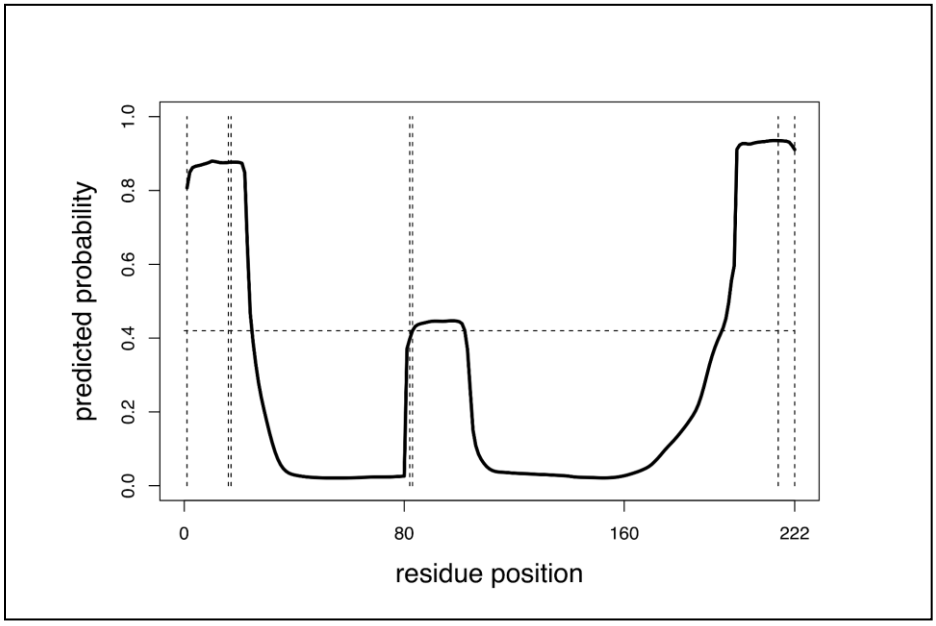


Figure 4. An illustration of discontinuous domain boundary prediction using target T0418 from CASP 8. The domain assignment is (1-16|83-216) (17-82), where the first domain has two segments. The defined domain boundaries are presented by vertical dash lines. The threshold of our model is 0.42.

To evaluate the ability of DeepDom in predicting discontinuous domain, we selected all the 18 targets that contain at least one discontinuous domain from CASP 8. The overall discontinuous domain boundary prediction precision is 81.2%, the recall is 34.8%, and with MCC of 0.38. However, currently we have not found a method to predict whether multiple segments belong to the same domain. Figure 4 gives an illustration of one discontinuous domain protein prediction.

4. CONCLUSION

In this paper, we designed a novel computational method called "DeepDom" for protein domain boundary prediction using deep learning. Our model does not need elaborated feature engineering. Instead, it extracts information from a large amount of raw sequence data. The comparison showed that DeepDom achieved better results than other *ab-initio* methods and is competitive with template-based methods. As an *ab-initio* method, DeepDom has the advantage to outperform the most successful template-based method when dealing with free modeling targets. Importantly, it can run much faster than other methods, all of which use sequence profiles that are time consuming to generate.

There is room for improvement of DeepDom. Ideally, a protein sequence should be encoded "globally", since breaking into fragments excludes the potential long distance dependency. By doing several experiments with varying window sizes and strides, an interesting discovery is that protein domain boundary prediction seems to depend on the signals from remote residues. However, this still requires further experiments to prove and develop a new method to use the information. The other limitation is that the prediction performance for template-available targets is lower than the best template-based method. We will develop a hybrid method that can take advantages of existing methods from both approaches (ab-initio and template-based). We also plan to make the hybrid method available as a web server. Most of the existing domain prediction web servers only allow users to submit one protein sequence a time. Since DeepDom avoids the time-consuming sequence profile generation process, the users can predict for a list of proteins in a short time.

5. ACKNOWLEDGEMENTS

The authors would like thank Dr. Jianlin Cheng and his student Jie Hou for their help to this work. This work was partially supported by National Institutes of Health grant R35-GM126985.

REFERENCES

1. Ponting CP, Russell RR, Annual review of biophysics and biomolecular structure. 31, 45-71 (2002).
2. Finn RD, Coggill P, Eberhardt RY, Eddy SR, Mistry J, Mitchell AL, Potter SC, Punta M, Qureshi M, Sangrador-Vegas A et al, Nucleic acids research. 44(D1), D279-285 (2016).
3. Liu J, Rost B, Proteins. 55(3), 678-688 (2004).
4. Bondugula R, Lee MS, Wallqvist A, Nucleic acids research. 37(2), 452-462 (2009).
5. Xue Z, Xu D, Wang Y, Zhang Y, Bioinformatics. 29(13), i247-256 (2013).
6. George RA, Heringa J, Journal of molecular biology. 316(3), 839-851 (2002).
7. Kim DE, Chivian D, Malmstrom L, Baker D, Proteins. 61 Suppl 7, 193-200 (2005).

8. Dawson NL, Lewis TE, Das S, Lees JG, Lee D, Ashford P, Orengo CA, Sillitoe I, Nucleic acids research. 45(D1), D289-D295 (2017).
9. Andreeva A, Howorth D, Chothia C, Kulesha E, Murzin AG, Nucleic acids research. 42(Database issue), D310-314 (2014).
10. Sim J, Kim SY, Lee J, Proteins. 59(3), 627-632 (2005).
11. Cheng J, Sweredoski MJ, Baldi P, Data Mining and Knowledge Discovery. 13(1), 1-10 (2006).
12. Servant F, Bru C, Carrere S, Courcelle E, Gouzy J, Peyruc D, Kahn D, Briefings in bioinformatics. 3(3), 246-251 (2002).
13. Suyama M, Ohara O, Bioinformatics. 19(5), 673-674 (2003).
14. Heger A, Wilton CA, Sivakumar A, Holm L, Nucleic acids research. 33(Database issue), D188-191 (2005).
15. Yoo PD, Sikder AR, Taheri J, Zhou BB, Zomaya AY, IEEE transactions on nanobioscience. 7(2), 172-181 (2008).
16. Ebina T, Toh H, Kuroda Y, Bioinformatics. 27(4), 487-494 (2011).
17. Eickholt J, Deng X, Cheng J, BMC bioinformatics. 12, 43 (2011).
18. Portugaly E, Harel A, Linial N, Linial M, BMC bioinformatics. 7, 277 (2006).
19. Hochreiter S, Schmidhuber J, Neural computation. 9(8), 1735-1780 (1997).
20. Graves A, Mohamed A-r, Hinton G, In: Acoustics, speech and signal processing (icassp), 2013 ieee international conference on: 2013. IEEE, 6645-6649 (Year).
21. Soutner D, Müller L, In: International Conference on Text, Speech and Dialogue: 2013. Springer, 105-112 (Year).
22. Vinyals O, Toshev A, Bengio S, Erhan D, In: Proceedings of the IEEE conference on computer vision and pattern recognition: 2015. 3156-3164 (Year).
23. Bengio Y, Simard P, Frasconi P, IEEE transactions on neural networks. 5(2), 157-166 (1994).
24. UniProt Consortium T, Nucleic acids research. 46(5), 2699 (2018).
25. Fu L, Niu B, Zhu Z, Wu S, Li W, Bioinformatics. 28(23), 3150-3152 (2012).
26. Wang D, Zeng S, Xu C, Qiu W, Liang Y, Joshi T, Xu D, Bioinformatics. 33(24), 3909-3916 (2017).
27. Venkatarajan MS, Braun W, Molecular modeling annual. 7(12), 445-453 (2001).

Res2s2aM: Deep residual network-based model for identifying functional noncoding SNPs in trait-associated regions

Zheng Liu[1,2], Yao Yao[1,2], Qi Wei[1,2], Benjamin Weeder[2], and Stephen A. Ramsey[1,2,†]

1. School of Electrical Engineering and Computer Science, Oregon State University
2. Department of Biomedical Sciences, Oregon State University
Corvallis, OR, 97330, USA
[†]*E-mail: stephen.ramsey@oregonstate.edu*

Noncoding single nucleotide polymorphisms (SNPs) and their target genes are important components of the heritability of diseases and other polygenic traits. Identifying these SNPs and target genes could potentially reveal new molecular mechanisms and advance precision medicine. For polygenic traits, genome-wide association studies (GWAS) are preferred tools for identifying trait-associated regions. However, identifying causal noncoding SNPs within such regions is a difficult problem in computational biology. The DNA sequence context of a noncoding SNP is well-established as an important source of information that is beneficial for discriminating functional from nonfunctional noncoding SNPs. We describe the use of a deep residual network (ResNet)-based model—entitled Res2s2aM—that fuses flanking DNA sequence information with additional SNP annotation information to discriminate functional from nonfunctional noncoding SNPs. On a ground-truth set of disease-associated SNPs compiled from the Genome-wide Repository of Associations between SNPs and Phenotypes (GRASP) database, Res2s2aM improves the prediction accuracy of functional SNPs significantly in comparison to models based only on sequence information as well as a leading tool for post-GWAS noncoding SNP prioritization (RegulomeDB).

Keywords: Deep Residual Network; Noncoding DNA; Sequence Analysis; GWAS.

1. Introduction

Prioritizing functional trait-associated noncoding SNPs in the human genome remains a critical and challenging problem. From thousands of genome-wide association studies, over 21,751 trait-associated SNPs have been reported.[1] However, noncoding SNPs can also have significant effects on trait variation including risks of certain diseases such as coronary artery disease or certain cancers.[2] Causal noncoding SNPs are thought affecting trait variation through gene regulatory mechanisms. Nevertheless, identifying such causal variants within trait-associated regions that have been implicated by GWAS is a difficult computational problem[3] because the noncoding DNA sequence and epigenomic determinants of regulatory sites are incompletely studied. While some genomic annotations are known to be informative for predicting whether or not a noncoding SNP is functional,[4] many sequence determinants of functional noncoding DNA are unknown and must be learned from training data. DNA sequence information up to a kilobase from a noncoding SNP can be informative as to whether or not that SNP is functional;[5] however, at that distance scale, the DNA sequence context of a SNP is

high-dimensional, posing significant challenges for traditional computational methods.

In recent years, significant advancements have been made in machine learning methods for handling high-dimensional datasets with complex interactions among features. Deep learning approaches are particularly powerful in this context because they enable the utilization of large-scale, high-dimensional, unstructured data as a substrate for predictive models. In machine-learning methods for image recognition, deep convolutional neural networks (CNNs) have emerged as a fundamental building block for deep learning approaches, due to the CNN's ability to learn composite data representations and the contours of objects from pixel-level data.[6] Recently, deep residual networks (ResNet)[7,8] have been proposed which have the advantage of smoothing the information propagation and more representing power with deeper network models. A key advantage of deep neural network models with differentiable activation functions is that the backpropagation algorithm for computing the loss function gradient can be used, which is compatible with computation on a graphical processing unit (GPU).

Deep learning methods have been used in computational biology in various contexts[9] including biomedical imaging, data-driven diagnostics, and pharmacogenomics. In the area of noncoding genome analysis, deep learning-based computational approaches have been used for both functional SNP prioritization and identification of regulatory sequence patterns, among which two approaches are notable: Basset[10] is a deep neural network model for predicting chromatin accessibility for cell-specific mutations using DNA sequences; and DeepSEA[5] is a convolutional neural network based framework trained on chromatin-profiling data that directly learns regulatory patterns *de novo* from SNP-flanking sequences. In the context of post-GWAS analysis to identify causal noncoding SNPs, the key computational problem relevant to this work can be defined as: given a DNA sequence acquired around a specific trait-associated noncoding SNP, and given a set of training (functional) SNPs, produce a score representing the confidence that the trait-associated SNP is functional.

In this work, we collated a set of training noncoding SNPs (divided into "functional" and "non-functional" classes) curated from GWAS studies, and obtained flanking genomic DNA sequences for the SNPs. We implemented 5 different neural network architectures for predicting the SNP class labels based on their flanking DNA sequences and (optionally) additional SNP annotation features from a database of noncoding SNP annotations (HaploReg): two CNN models based on DeepSEA,[5] a CNN model based on DeFine[11] (with two sets of optimization algorithms and loss functions), a new sequence-based deep residual network approach (which we call Res2s2a) that we propose, and a hybrid network (which we call Res2s2aM) fusing Res2s2a with HaploReg-derived SNP annotation features. We trained the neural network models using a stochastic gradient optimization method (Adam)[12] and evaluated their performance for discriminating functional from non-functional noncoding SNPs in hold-out examples. We found that the deep residual network models (Res2s2a and Res2s2aM) outperformed the CNN-based models, and that the hybrid model (Res2s2aM) outperformed the sequence-only model (Res2s2a). This work is the first application of deep residual networks for noncoding SNP prioritization of which we are aware, and it suggests that ResNet models can significantly advance the state-of-the-art for computational methods for post-GWAS SNP prioritization. All of the code for this work (including the new methods Res2s2a and Res2s2aM) is available

on the open-source software repository GitHub (https://github.com/zheng-liu/res2s2am).

2. Background theory

CNNs. In previous convolutional neural network based methods involving DNA sequences, the models take one-hot-encoded DNA sequence as input and predict class-specific scores as output. Through filtering kernels with variable weights, convolutional layers exploit spatial locality to develop discriminating signals at successively coarse-grained scales. The same filtering kernel (i.e., with identical weights) is applied at each neuron position in the layer. Pooling layers effect downsampling to reduce dimensionality issue and make abstracted representation binned in certain sections. Nonlinear activation layers (e.g., ReLU) aim to add nonlinearity in the model for larger and more flexible projecting space from sequences to labels. The convolutional layers are organized in a general form shown in Figure 1. By successive convolution

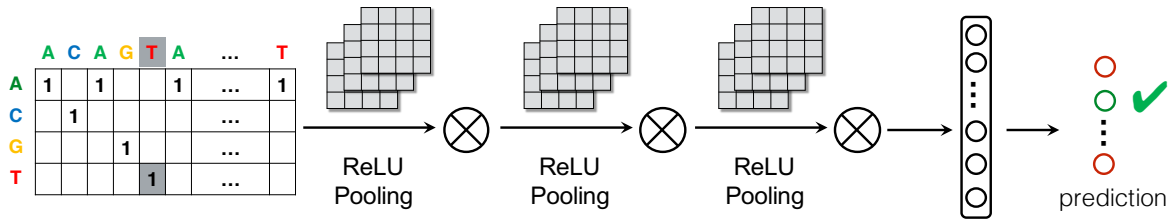


Fig. 1. General CNN models architecture. One-hot-encoded sequence data (left) shown as a $4 \times L$ matrix; ReLU denotes a network unit based on the rectifier function, $(f(x) = \max(0, x))$; the $\otimes$ symbol denotes convolution; the pooling layer selects the stronger signals from previous layer; the final rightmost arrow represents a prediction layer (e.g., softmax or logistic function).

operations, the network starts to learn the locality of data and produces advanced features in intermediate layer filters.[10] More layers bring larger parameter spaces and equivalently more representing power towards the input signals. Unexpectedly, as indicated by He et al.,[7] a degradation problem happens when deeper networks are built: the prediction performance becomes saturated with increasing number of of hidden layers.

Residual nets. Deep residual network (ResNet)[7,8] is an approach to address the saturating problem in the meanwhile tapping the potential of deeper nets. The ResNet approach is based on a feed-forward neural network with shortcut connections (based on the identity function $I(x) = x$) between non-adjacent layers. At the end of a module (made up of two or more layers), the mapped identical signal $I(x) = x$ is added into the output of stacked module layers. In the pipeline of ResNet, the model is established with multiple modules of hidden layers as shown in Figure 2.

Instead of fitting the original input signal x into each layer module, ResNet fits the residual signal $H(x) - x$ based on the assumption that the residual signal is more likely to overcome the local optimums in gradient-based optimization processes. In the training procedure, if the optimal fitting to $H(x)$ is the identity function $H(x) = x$, the stacked module layers are trying to fit an always-zero constant signal which is much easier than fitting an identity mapping using the nonlinear layers in the module. More importantly, as a common problem,

deeper nets tend to cause more vanishing gradient problem that small gradients multiplication following chain rule leads to loss of information at the end. ResNet with an identity function as a shortcut always possesses a 1.0 gradient component which largely stables the gradient calculation in backpropagation. Formally, the module output is defined in Equation (1):

$$\begin{aligned} H(x) &= f(x) + x \\ &= W_2 \times \mathrm{ReLU}(W_1 \cdot x + b_1) + b_2 + x, \end{aligned} \tag{1}$$

where W_1, W_2, b_1, and b_2 are coefficients.

ResNet mounts shortcuts of identity functions besides the stacked layer modules to make the weight matrix easier to fit the signal primarily when the intended signal is x itself. Even though adding extra coefficients to identity functions $I_i(x) = x$ as $I_i(x) = \lambda x$ seems to provide more flexibility to shortcuts, it is nontrivial to notice that those coefficients introduce more optimization difficulties.[8] Veit et al. explain the ResNet effectiveness in an ensemble view that ResNet is a collection of independent paths differing in length, and only short paths are trained.[13] Thus, compared to other CNN models, a ResNet architecture with identity skipping function is adapted to GWAS SNP prioritization problem in this work.

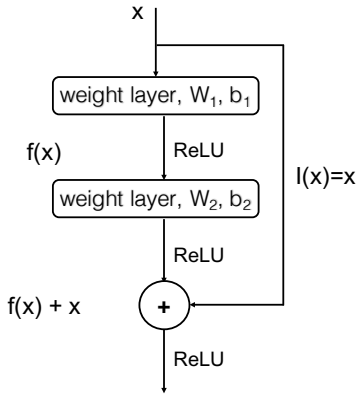


Fig. 2. A building block of Residual Network

3. Dataset for training and testing

To verify the model effectiveness, we assembled a dataset of trait-associated noncoding DNA sequences together with control cases (noncoding SNPs in the same genomic loci as positive SNPs but for which there is no trait association). In this section we describe the procedures used to build the dataset.

3.1. *Source databases*

In this work we used four source databases to obtain the information required to build a feature matrix on a set of example SNPs. From the GRASP database[14] we obtained a dataset of 2.48 M SNPs (identified by dbSNP RefSNP IDs or "rsIDs"[15]). GRASP was selected because it is comprised of significant SNPs from a large number (1390) of GWAS studies with diverse traits. We used the UCSC Genome Center knownGene database[16] for chromosomal coordinate information of SNPs in the GRCh37/hg19 genome assembly. We used the UCSC Genome Browser knownGene table of gene annotations to obtain chromosomal coordinates of genes, transcripts, and exons (in the same genome assembly). We used the web tool HaploReg[17] for mapping between GRASP SNPs and neighboring SNPs that are in linkage disequilibrium with the GRASP SNPs ("proxy SNPs") and for obtaining functional annotations for SNPs including consensus functional SNP scores that were assigned by the RegulomeDB project.[18] We used the UCSC Genome Browser to obtain flanking genomic sequence (1 kbp window size) for each SNP in our dataset.

3.2. *Dataset generation*

Positive dataset generation. We annotated each SNP based on its location relative to known gene annotations using all Ensembl transcripts,[19] assigning the SNP to an annotation category out of "pcexon (protein-coding exon)", "intron", "3′UTR", "5′UTR", "nonpcexon (non-protein-coding exon)", "intergenic". Following a specific strand direction, If a SNP overlapped a protein-coding exon in any transcript, it was annotated as coding. If a SNP was not marked as coding by the previous step but was found to overlap a UTR in any transcript, it was annotated with the corresponding UTR (3′ or 5′). If a SNP was not annotated as coding or UTR by the previous steps, but if that SNP was located in an intron for any transcript, it was annotated as intronic. If a SNP in a transcript did not overlap with any coding exon, it is assigned to "nonpcexon" category. Otherwise, the SNP was annotated as intergenic. Next, we filtered to obtain a positive-example set of SNPs following criteria: (1) SNPs residing in protein-coding exons were excluded. (2) Any SNP within 1 Mbp of a trait-associated ($P < 5 \times 10^{-8}$ in at least one record in GRASP) protein-coding SNP was excluded. (3) Remaining noncoding SNPs meeting the significance criteria ($P < 5 \times 10^{-8}$ in at least one GWAS) that had the lowest P value within 1 Mbp were retained as positive examples. (4) The rest noncoding SNPs with minimum P-value in the neighborhood of noncoding SNPs are specified as positive cases. This procedure yielded a set of 128,944 positive examples of noncoding SNPs.

Control case generation. Using HaploReg,[17] we obtained SNPs that are in linkage disequilibrium (within 250 kbp and with correlation coefficient $r^2 \geq 0.8$) with SNPs from the positive set. Each positive SNP was expanded to SNPs from four population groups ("AFR", "AMR", "ASN", "EUR") in the 1,000 Genome (1KG) Project[20] and then combined. In the set of resulting proxy SNPs, any SNPs that were listed in the GRASP database or protein-coding were excluded, resulting in a set of 1,412,452 noncoding control SNPs that were treated as negative examples. Additionally, we obtained annotation features about the SNP set using HaploReg, including allele frequencies, conservation scores et al. Table 1 details the biological features that we used in the Res2s2aM model. We obtained RegulomeDB scores from RegulomeDB webservice directly used as a categorical feature in the Res2s2aM model and also as a standalone predictor. We mapped the 15 RegulomeDB score categories ("1a", "1b", "1c", ... "5", "6", "7") to [1.0, 2.0, ..., 15.0] for this purpose, assigning the value 16.0 to missing RegulomeDB scores (note: a lower RegulomeDB score corresponds to greater evidence for a noncoding SNP to be functional[18]). This procedure yielded 1,541,396 SNPs in total with a class ratio of about 1:10.9 (positive SNPs : control SNPs).

SNP annotation feature evaluation. In order to quantify the discriminating power of individual SNP annotation features (from HaploReg) on our set of 1.5 million SNPs, we computed empirical log-likelihood ratios (positive:control) of each of the SNP annotation features (Fig. 3). This analysis showed that, consistent with the fact that it is comprised of multiple types of independent evidence for functional noncoding SNPs, RegulomeDB (Fig. 3e) is the strongest predictor among the SNP annotation features. Further, the analysis shows an strong association between the reference allele frequency and the likelihood ratio, in each of the 1KG population groups.

Table 1. The SNP annotation features used in the hybrid Res2s2aM model

feature name	*feature type*	*feature description*
AFR	continuous	RefAllele Freq in the African population (492 samples)
AMR	continuous	RefAllele Freq in the Ad Mixed American population (362 samples)
ASN	continuous	RefAllele Freq in the Asian population (572 samples)
EUR	continuous	RefAllele Freq in the European population group (758 samples)
reg_score_int	categorical	RegulomeDB score encoded from 1.0 to 16.0
GERP_cons	categorical	GERP phylogenetic sequence conservation score[21]
SiPhy_cons	categorical	SiPhy selective constraint score[22]

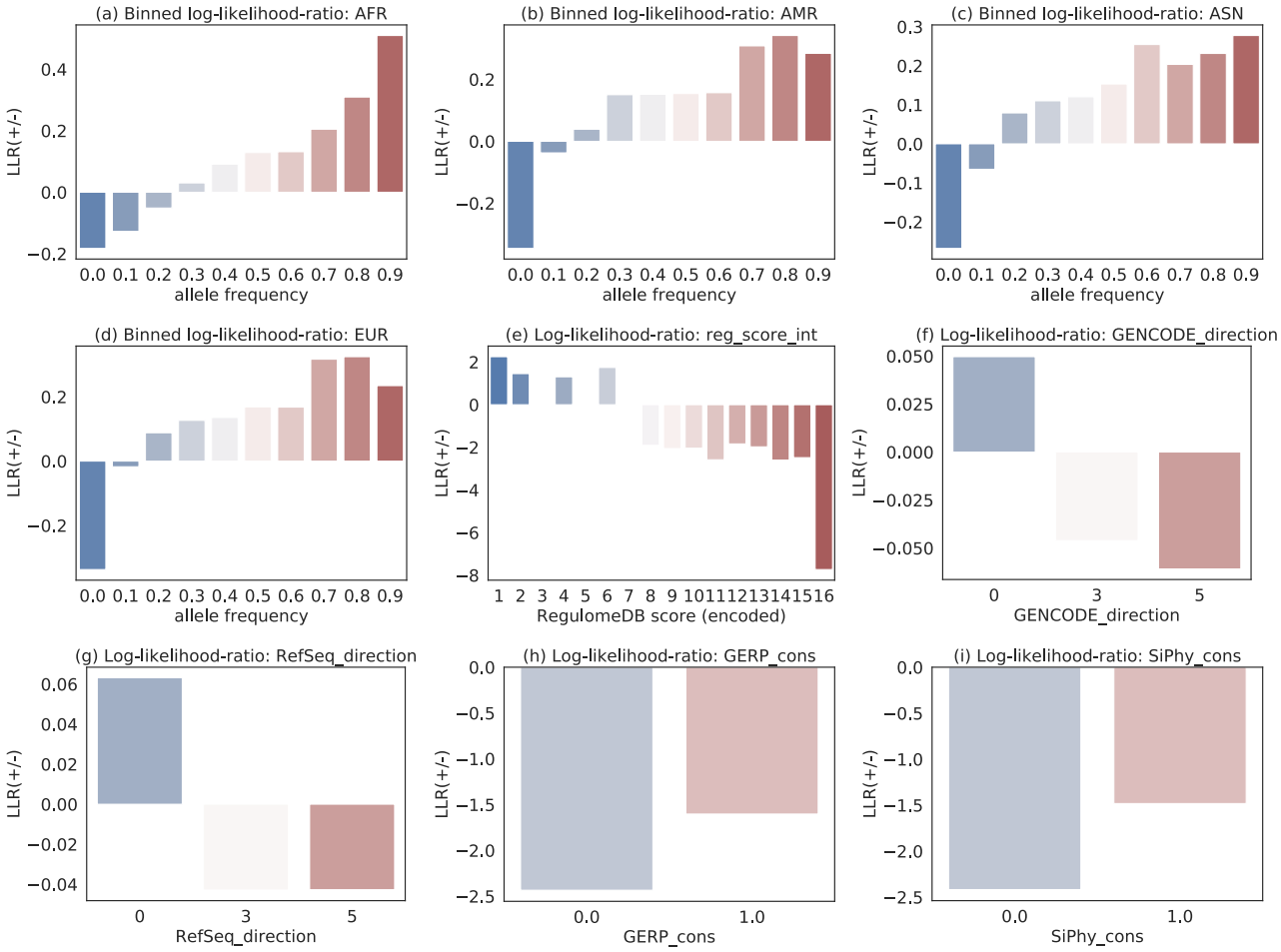


Fig. 3. Estimated log likelihood ratio (LLR) of features. "Direction" means the location of the SNP relative to the nearest gene (0 = within; 3 = downstream, 5 = upstream).

4. Methods

4.1. *ResNet architecture in our model*

Our model (Fig. 4) uses a 1 kbp sequence along each strand which is one-hot-encoded as a 4×1000 sparse matrix. The matrix is treated as a 4-channel input signal with each row as a

single channel input. After both encoded strands are input into the model, a convolution step based on 16 convolutional kernels (each of size 7×7) is performed on them with a stride of 2 bp. The output of the previous layer is batch normalized,[23] ReLU activated, and a max pooling layer is applied to reduce dimension. Next, 4 groups of residual blocks are built with various output channels, layers, and filter strides. Each residual block consists of 3 batch-normalized convolutional layers with ReLU activation and the residual skipping shortcut connections. An average pooling layer with kernel size to 4 bp is applied to the output of the residual block. The output of average pooling layers from both strands are expanded into 1-D vectors and combined into one single vector as the final output for both strands.

4.2. *Tandem inputs of forward- and reverse-strand sequences*

Genomic DNA is double-stranded, and thus, to make a consistent prediction with the same SNP sequences along both strand directions, we incorporate input DNA sequences along both "+" and "-" strands (the latter being reverse-complemented) into our CNN- and ResNet-based models. As it is demonstrated that reverse-complement parameter sharing contributes to deep learning in genomics,[24] the reverse-complement sequence segments are encoded in our model (along with the forward-strand sequence) as input signals. In the training process, each residual building block shares weights between both forward and reverse-complement sequences.

4.3. *Biallelic high-level network structure*

A key potential issue with using neural networks to score genomic sequence flanking a SNP is the need to account for the two alleles of the central SNP. Convolutional operations are the critical components in convolutional neural network based models including ResNet. Most existing models are trained merely on reference allele sequence flanking a specific variant position. In this paper, we aim at the contrast between the reference allele and the alternative allele and highlight the effect of the central SNPs. The architecture of the sequence learning module in the Res2s2aM model is illustrated in Figure 4.

4.4. *Incorporating HaploReg SNP annotation features*

In previous studies, SNP annotation features have proved essential for identifying functional noncoding SNPs.[25] We trained the Res2s2aM model to *learn* feature embeddings jointly with the encoded sequence. This method is inspired by natural language processing models where words are mapped to a fixed dimension of vectors. We used a fully connected layer of 100 nodes as the embedding layer to represent both continuous and categorical features (Fig. 4, dotted rectangle). The overall data fusion algorithm for Res2s2aM is defined in Algorithm 1.

4.5. *Training of models*

For parameter fitting in all models except "DeFine0," we used Adam,[12] a stochastic algorithm for parameter optimization, with cross-entropy as the loss function. [For the "DeFine0" model, following Wang et al.,[11] we used stochastic gradient descent as optimization algorithm

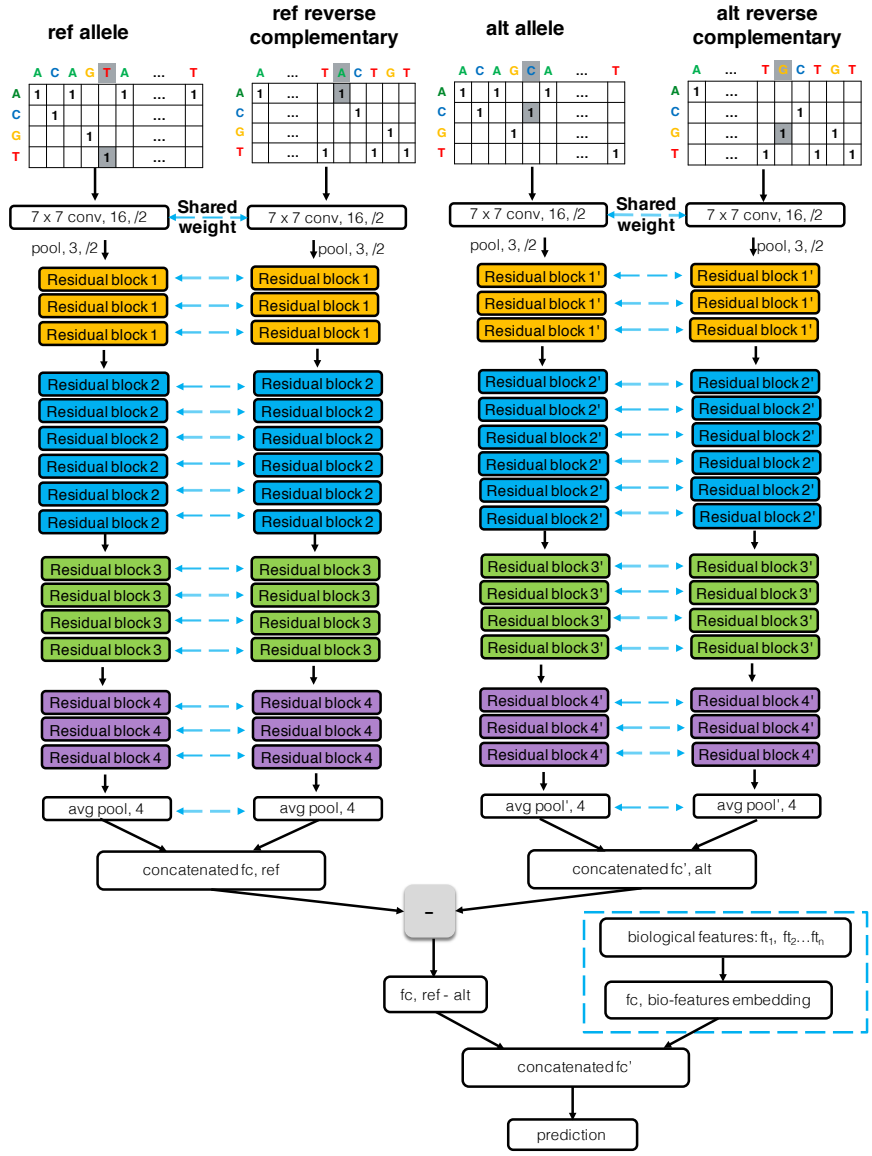


Fig. 4. Architecture of model Res2s2aM. In the Res2s2a model, the portion of the network shown in the dotted rectangle (which is based on SNP annotation data from HaploReg) is not included.

and mean squared error with L2 regularization as loss function.] Model parameters were initialized before training. All parameters in convolutional layers were initialized by sampling $\mathcal{N}(0, \sqrt{2.0/c})$, where c equals the total number of output dimensions [DeFine0 and DeFine initialized conv layers to $\mathcal{N}(0, 1)$]. All the batched norm layers were initialize their weights to 1.0 and biases to 0. We trained 40 epochs for each model and saved the model parameters at the epoch with lowest validation-set loss. Also, we used an early stop mechanism during training: training was terminated if the validation loss continuously increased for ten epochs. As seen in Figure 6, the training loss of ResNet-based models (on the validation set) reached a minimum in 10–15 epochs. Other models' architectures are shown in Figure 5.

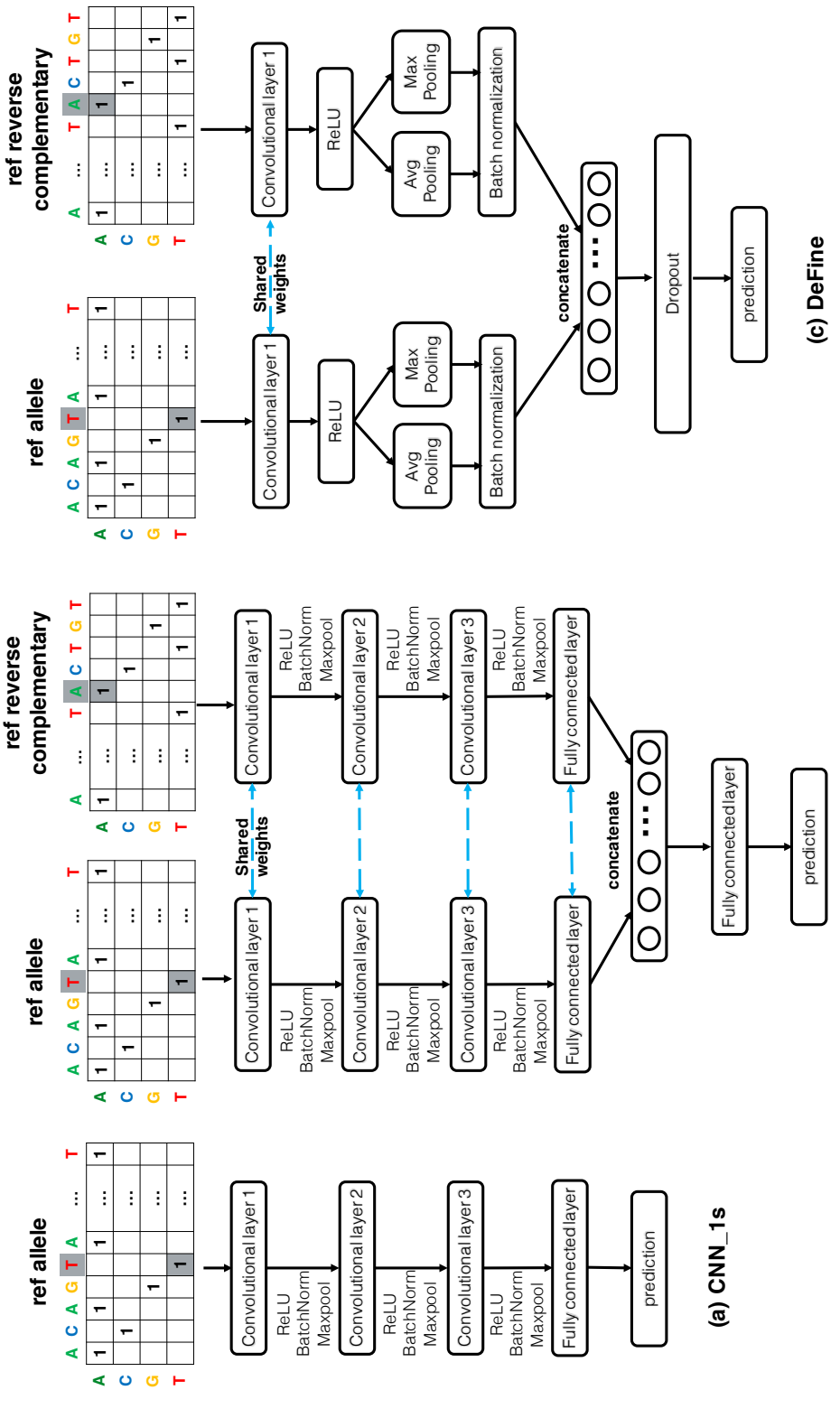


Fig. 5. Deep models to compare with: (a) CNN_1s is the CNN model with "+" strand sequence. (b) CNN_2s is the CNN model with "+" and "-" strands (c) DeFine[11]

Algorithm 1 Res2s2aM

procedure SEQAUGMENT(x) ▷ Expansion of ref seq
  $x_1 = x$ ▷ Ref seq: + strand
  $x_1^{'} = \bar{x_1}^{-1}$ ▷ Ref seq: - strand, reverse complement
  $x_2 = \text{alt}(x_1)$ ▷ Alt seq: + strand
  $x_2^{'} = \bar{x_2}^{-1}$ ▷ Alt seq: - strand, reverse complement
procedure SEQLEARN(x_1, $x_1^{'}$, x_2, $x_2^{'}$)
  Initialize Conv layers conv_i and BatchNorm layers bn_i, $i \in \{1, 2\}$
  $x_1, x_1^{'}, x_2, x_2^{'} = \mathbf{conv}_1(x_1), \mathbf{conv}_1(x_1^{'}), \mathbf{conv}_2(x_2), \mathbf{conv}_2(x_2^{'})$ ▷ Filters sharing
  $x_1, x_1^{'}, x_2, x_2^{'} = \mathbf{bn}_1(x_1), \mathbf{bn}_1(x_1^{'}), \mathbf{bn}_2(x_2), \mathbf{bn}_2(x_2^{'})$
  $x_1, x_1^{'}, x_2, x_2^{'} = \mathbf{maxpool}_1(x_1), \mathbf{maxpool}_1(x_1^{'}), \mathbf{maxpool}_2(x_2), \mathbf{maxpool}_2(x_2^{'})$
  $x_1, x_1^{'}, x_2, x_2^{'} = \mathbf{relu}_1(x_1), \mathbf{relu}_1(x_1^{'}), \mathbf{relu}_2(x_2), \mathbf{relu}_2(x_2^{'})$
  for i = $1 : n_r$ **do** ▷ Residual blocks
    $x_1, x_1^{'}, x_2, x_2^{'} = \mathbf{ResBlock}_1^i(x_1), \mathbf{ResBlock}_1^i(x_1^{'}), \mathbf{ResBlock}_2^i(x_2), \mathbf{ResBlock}_2^i(x_2^{'})$
  $x_1, x_1^{'}, x_2, x_2^{'} = \mathbf{avgpool}_1(x_1), \mathbf{avgpool}_1(x_1^{'}), \mathbf{avgpool}_2(x_2), \mathbf{avgpool}_2(x_2^{'})$
  $x_{ref}, x_{alt} = [x_1, x_1^{'}]_{1d}, [x_2, x_2^{'}]_{1d}$ ▷ Flatten and combine to 1-D vector
  $x_\Delta = x_{ref}$ - x_{alt} ▷ Train on difference of Ref and Alt seqs
procedure METAEMBED(x_{meta})
  $x_{meta} = \mathbf{fc}_{meta}(x_{meta})$ ▷ Metadata embedding
  $X = [x_\Delta, x_{meta}]_{1d}$
  $X = \mathbf{fc}(X)$
return X

5. Results

We trained and evaluated six models: Res2s2aM, Res2s2a, DeFine0 (the DeFine network model with the original optimization algorithm and objective function), DeFine (with Adam optimization and cross-entropy loss), CNN_1s, and CNN_2s on 5 random data spliting assignments. Additionally we compared the accuracy of the supervised models to an unsupervised approach in which SNPs were ranked by their scores from the RegulomeDB tool. We found that Res2s2aM significantly improves (Table. 2) over Res2s2a on testing-set area under the receiver operating characteristic (AUROC) curve (from 0.74 to 0.76). By area under the precision-versus-recall curve (AUPRC), Res2s2aM (0.21) also had higher performance than Res2s2a (0.18). In addition to having superior accuracy, Res2s2a and Res2s2aM trained significantly faster than the CNN-based models. Our model also has over 75% prediction accuracy to CVD, gastrointestinal and blood-related diseases. Validation-set losses during training Res2s2a and Res2s2aM terminate earlier than other models due to early stop mechanism (Fig. 6).

6. Conclusions and discussion

By introducing residual skipping connection and ResNet into functional noncoding SNP prioritization and multi-modal fusion of biological features with DNA sequence, Res2s2aM improves the performance of noncoding functional SNP prioritization. Res2s2aM makes full use of both

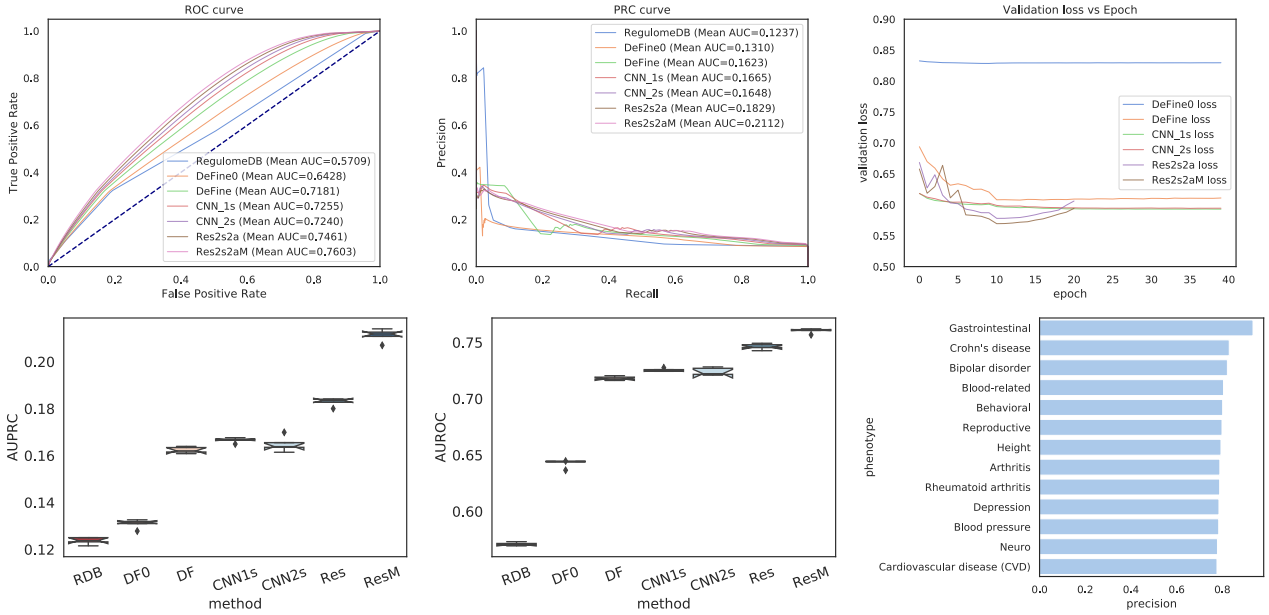


Fig. 6. Performance comparison of seven models: ResM = Res2s2aM , Res = Res2s2a, DF = DeFine, DF0 = DeFine0, CNN2s = CNN_2s, CNN1s = CNN_1s, and RDB = RegulomeDB. Lines, boxes, and marks denote median, interquartile range, and outliers, respectively.

unstructured sequence data and more biological features (continuous and categorical), leading to an end-to-end deep neural network architecture. The experimental performance suggests that (1) use of residual shortcut connections could potentially benefit the more general sequence based deep learning and (2) embedding biological features in an end-to-end fashion could be helpful for utilizing more information sources while training deep models. By improving prediction accuracy of the ground-truth SNPs using merely flanking sequences and accessible biological features, prediction scores can be obtained for SNPs in a loci, which prioritize functional noncoding SNPs following genotype-to-phenotype studies. However, from what we observed, the Res2s2aM model has some disadvantages including: high memory requirements, limitations in semi-supervised setting. We will adapt the ResNet-based model to semi-supervised setting in our future work.

Table 2. Validation-set performance (95% confidence interval and p-value vs. Res2s2aM)

method name	*AUROC (95% CI)*	*AUROC (p-value)*	*AUPRC (95% CI)*	*AUPRC (p-value)*
Res2s2aM	**(0.7579, 0.7627)**	-	**(0.2082, 0.2142)**	-
Res2s2a	(0.7432, 0.7491)	9.8×10^{-5}	(0.1809, 0.1848)	3.2×10^{-6}
cnn_2s	(0.7201, 0.7278)	9.2×10^{-6}	(0.1616, 0.1685)	4.3×10^{-6}
cnn_1s	(0.7240, 0.7269)	2.3×10^{-6}	(0.1654, 0.1677)	3.3×10^{-6}
DeFine	(0.7162, 0.7200)	1.1×10^{-6}	(0.1608, 0.1638)	8.0×10^{-7}
RegulomeDB	(0.5692, 0.5726)	6.7×10^{-10}	(0.1220, 0.1253)	1.1×10^{-8}

Acknowledgements

This work was supported by the Medical Research Foundation of Oregon (New Investigator Award to S.A.R.), Oregon State University (Health Sciences award to S.A.R.), the PhRMA Foundation (Research Starter Grant in Informatics to S.A.R.) and the National Science Foundation (awards 1557605-DMS and 1553728-DBI to S.A.R.).

References

1. J. MacArthur, E. Bowler, M. Cerezo, L. Gil, P. Hall, E. Hastings, H. Junkins, A. McMahon, A. Milano, J. Morales *et al.*, *Nucleic acids research* **45**, D896 (2016).
2. M. Nikpay, A. Goel, H.-H. Won, L. M. Hall, C. Willenborg, S. Kanoni, D. Saleheen, T. Kyriakou, C. P. Nelson, J. C. Hopewell *et al.*, *Nature genetics* **47**, p. 1121 (2015).
3. L. Gao, Y. Uzun, P. Gao, B. He, X. Ma, J. Wang, S. Han and K. Tan, *Nature Communications* **9**, p. 702 (February 2018).
4. G. R. S. Ritchie, I. Dunham, E. Zeggini and P. Flicek, *Nature Methods* **11**, 294 (March 2014).
5. J. Zhou and O. G. Troyanskaya, *Nature Methods* **12**, p. 931 (2015).
6. A. Krizhevsky, I. Sutskever and G. E. Hinton, 1097 (2012).
7. K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
8. K. He, X. Zhang, S. Ren and J. Sun, Identity mappings in deep residual networks, in *European Conference on Computer Vision*, 2016.
9. C. Angermueller, T. Pärnamaa, L. Parts and O. Stegle, *Mol Syst Biol* **12**, p. 878 (2016).
10. D. R. Kelley, J. Snoek and J. L. Rinn, *Genome Research* (2016).
11. M. Wang, C. Tai, W. E and L. Wei, *Nucleic Acids Research* **46**, e69 (2018).
12. D. P. Kingma and J. Ba, *arXiv preprint arXiv:1412.6980* (2014).
13. A. Veit, M. J. Wilber and S. Belongie, 550 (2016).
14. R. Leslie, C. J. O'donnell and A. D. Johnson, *Bioinformatics* **30**, i185 (2014).
15. S. T. Sherry, M.-H. Ward, M. Kholodov, J. Baker, L. Phan, E. M. Smigielski and K. Sirotkin, *Nucleic Acids Research* **29**, 308 (2001).
16. D. Karolchik, R. Baertsch, M. Diekhans, T. S. Furey, A. Hinrichs, Y. Lu, K. M. Roskin, M. Schwartz, C. W. Sugnet, D. J. Thomas *et al.*, *Nucleic acids research* **31**, 51 (2003).
17. L. D. Ward and M. Kellis, *Nucleic acids research* **40**, D930 (2011).
18. A. P. Boyle, E. L. Hong, M. Hariharan, Y. Cheng, M. A. Schaub, M. Kasowski, K. J. Karczewski, J. Park, B. C. Hitz, S. Weng *et al.*, *Genome Research* **22**, 1790 (2012).
19. T. Hubbard, D. Barker, E. Birney, G. Cameron, Y. Chen, L. Clark, T. Cox, J. Cuff, V. Curwen, T. Down *et al.*, *Nucleic acids research* **30**, 38 (2002).
20. 1000 Genomes Project Consortium, *Nature* **526**, p. 68 (2015).
21. G. M. Cooper, E. A. Stone, G. Asimenos, NISC Comparative Sequencing Program, E. D. Green, S. Batzoglou and A. Sidow, *Genome Research* **15**, 901 (July 2005).
22. M. Garber, M. Guttman, M. Clamp, M. C. Zody, N. Friedman and X. Xie, *Bioinformatics* **25**, 54 (May 2009).
23. S. Ioffe and C. Szegedy, *arXiv preprint arXiv:1502.03167* (2015).
24. A. Shrikumar, P. Greenside and A. Kundaje, *bioRxiv* , p. 103663 (2017).
25. Y. Yao, Z. Liu, S. Singh, Q. Wei and S. A. Ramsey, CERENKOV: Computational Elucidation of the Regulatory Noncoding Variome, in *ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics*, (ACM, New York, NY, USA, 2017).

DNA Steganalysis Using Deep Recurrent Neural Networks

Ho Bae [1], Byunghan Lee [2, 3], Sunyoung Kwon [2, 4] and Sungroh Yoon [1, 2, 5,*]

[1] *Interdisciplinary Program in Bioinformatics, Seoul National University, Seoul 08826, Korea*
[2] *Electrical and Computer Engineering, Seoul National University, Seoul 08826, Korea*
[3] *Electronic and IT Media Engineering, Seoul National University of Science and Technology, Seoul 01811, Korea*
[4] *Clova AI Research, NAVER Corp., Seongnam 13561, Korea*
[5] *ASRI and INMC, Seoul National University, Seoul 08826, Korea*
E-mail: sryoon@snu.ac.kr

Recent advances in next-generation sequencing technologies have facilitated the use of deoxyribonucleic acid (DNA) as a novel covert channels in steganography. There are various methods that exist in other domains to detect hidden messages in conventional covert channels. However, they have not been applied to DNA steganography. The current most common detection approaches, namely frequency analysis-based methods, often overlook important signals when directly applied to DNA steganography because those methods depend on the distribution of the number of sequence characters. To address this limitation, we propose a general sequence learning-based DNA steganalysis framework. The proposed approach learns the intrinsic distribution of coding and non-coding sequences and detects hidden messages by exploiting distribution variations after hiding these messages. Using deep recurrent neural networks (RNNs), our framework identifies the distribution variations by using the classification score to predict whether a sequence is to be a coding or non-coding sequence. We compare our proposed method to various existing methods and biological sequence analysis methods implemented on top of our framework. According to our experimental results, our approach delivers a robust detection performance compared to other tools.

Keywords: Deep recurrent neural network, DNA steganography, DNA steganalysis, DNA watermarking

1. Introduction

Steganography serves to conceal the existence and content of messages in media using various techniques, including changing the pixels in an image[1]. Generally, steganography is used to achieve two main goals. On the one hand, it is used as digital watermarking to protect intellectual property. On the other hand, it is used as a covert approach to communicating without the possibility of detection by unintended observers. In contrast, steganalysis is the study of detecting hidden messages. Steganalysis also has two main goals, which are detection and decryption of hidden messages[1,2].

Among the various media employed to hide information, deoxyribonucleic acid (DNA) is appealing owing to its chemical stability and, thus is a suitable candidates as a carrier of

concealed information. As a storage medium, DNA has the capacity to store large amounts of data that exceed the capacity of current storage media[3]. For instance, a gram of DNA contains approximately 10^{21} DNA bases (108 terabytes), which indicates that only a few grams of DNA can store all information available[4]. In addition, with the advent of next-generation sequencing, individual genotyping has become affordable[5], and DNA in turn has become an appealing covert channels.

To hide information in a DNA sequence, steganography methods require that a reference target sequence and a message to be hidden[6]. A naïve example of a substitution-based method for watermarking that exploits the preservation of amino acids is shown in Fig. 1 (see the caption for details). The hiding space of this method is restricted to exon regions using a complementary pair that does not interfere with protein translation. However, most DNA steganography methods are designed without considering the hiding spaces, and they change a sequence into a binary format utilizing well-known encryption techniques.

In this regard, Clelland et al.[7], first proposed DNA steganography that utilized the microdot technique. Yachie et al.[8], demonstrated that living organisms can be used as data storage media by inserting artificial DNA into artificial genomes and using a substitution cipher coding scheme. This technique is reproducible and successfully inserts four watermarks into the cell of a living organism[9]. Several other encoding schemes have been proposed [10,11]. The DNA-Crypt coding scheme [12] translates a message into 5-bit sequences, and the ASCII coding scheme[13] translates words into their ASCII representation, converts them from decimals to binary, and then replaces 00 with adenine (`A`), 01 with cytosine (`C`), 10 with guanine (`G`), and 11 with thymine (`T`).

With the recent advancements with respect to steganography methods, various steganalysis studies have been conducted using traditional storage media. Detection techniques that are based on statistical analysis, neural networks, and genetic algorithms[14] have been developed for common covert objects such as digital images, video, and audio. For example, Bennett[1] exploits letter frequency, word frequency, grammar style, semantic continuity, and logical methodologies. However, these conventional steganalysis methods have not been applied to DNA steganography.

In this paper, we show that conventional steganalysis methods are not directly applicable to DNA steganography. Currently, the most commonly employed detection schemes, i.e., a statistical hypothesis testing methods, are limited with respect to the number of input queries in order to estimate distribution to perform statistical test[15]. To overcome the limitations of these existing methods, we propose a DNA steganalysis method based on learning the internal structure of unmodified genome sequences (*i.e.*, intron and exon modeling[16,17]) using deep recurrent neural networks (RNNs). The RNN-based classifier is used to identify modified genome sequences. In addition, we enhance our proposed model using unsupervised pre-training of a sequence-to-sequence autoencoder in order to overcome the restriction of the robustness of detection performance. Finally, we compare our proposed method to various machine learning-based classifiers and biological sequence analysis methods that were implemented on top of our framework.

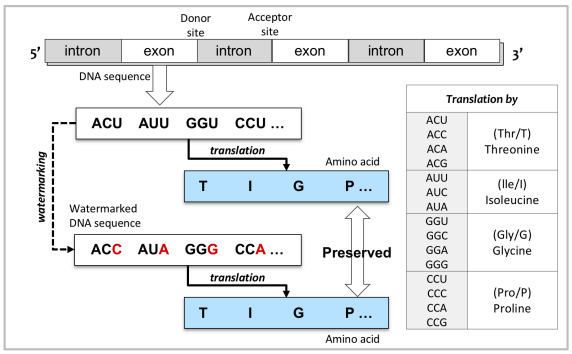


Fig. 1. DNA hiding scheme using synonymous codons. A watermark is a scheme used to deter unauthorized dissemination by marking hidden symbols or texts. For the conservation of amino acids, DNA watermarking can be changed to one of the synonymous codons.

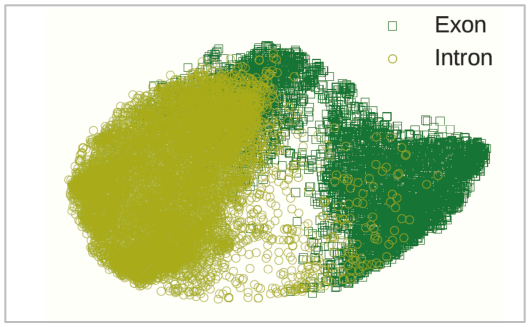


Fig. 2. Learned representation of DNA sequences. The learned representations for each coding and non-coding region projected into a two-dimensional (2-D) space using t-SNE.[18] The representation is based on sequence-to-sequence learning using an autoencoder and stacked RNNs.

2. Background

We use the standard terminology of information hiding[19] to provide a brief explanation of the related background. For example, two hypothetical parties, (i.e., a sender and a receiver) wish to exchange genetically modified organisms (GMOs) protected by patents. A third party detects watermark sequence from the GMOs for unauthorized use. Both the sender and receiver use the random oracle[20] model, which posits existing steganography schemes, to embed their watermark message, and the third party uses our proposed model to detect the watermarked signal. A random oracle model posits the randomly chosen function H, which can be evaluated only by querying the oracle that returns $H(m)$ given input m.

2.1. *Notations*

The notations used in this paper are as follows: $\mathbf{D} = \{D_1, \cdots, D_n\}$ is a set of DNA sequences of n species; $\hat{\mathbf{D}} = \{\hat{D}_1, \cdots, \hat{D}_n\}$ is a set of DNA sequences of n species and the hidden messages are embedded for some species $\hat{D}_i$; $m \in \{\texttt{A,C,G,T}\}^\ell$ is the input sequence where ℓ is the length of the input sequence; $\hat{m} \in \{\texttt{A,C,G,T}\}^\ell$ is the encrypted value of m where ℓ is the length of the encrypted sequence; E is an encryption function, which takes input m and returns the encrypted sequence $E(m) \rightarrow \hat{m}$; $\mathbf{M}_{D_i}$ is a trained model that takes target species D_i as training input; $\overline{y}$ is an averaged output score y; $\hat{y}$ is a probability output given by the trained model $\mathbf{M}_{D_i}(\hat{m}) \rightarrow \hat{y}$ given input $\hat{m}$, where $\hat{m} \in \hat{D}_i$; $\mathcal{A}$ is a probabilistic polynomial-time adversary. The adversary[21] is an attacker that queries messages to the oracle model; ϵ is the standard deviation value of score y.

2.2. *Hiding Messages*

The hiding positions of a DNA sequence segment are limited compared to those of the covert channel because the sequences are carried over after the translation and transcription processes in the exon region. For example, assume that `ACGGTTCCAATGC` is a reference sequence, and

01001100 is the message to be hidden. The reference sequence is then translated according to any coding schemes. In this example, we apply the DNA-crypt coding scheme[12], which converts the DNA sequence to binary replacing `A` with 00, `C` with 01, `G` with 10, and `T` with 11. The reference sequence is then translated to 000110101111010100000111001 and divided into key bits that are defined by the sender and receiver. Assume that the length of the key is 3, the reference sequence can be expressed as 000, 110, 101, 111, 010, 100, 001, 110, 01, and the message is concealed at the first position. The DNA sequence with the concealed messages are then represented as 0000, 1110, 0101, 0111, 1010, 1100, 0001, 0110, 01. Finally, the sender transmits the transformed DNA sequence of `AATGCCCTGGTAACCG`. The recipient can extract the hidden message using the pre-defined key.

2.3. *Determination of Message-Hiding Regions*

Genomic sequence regions (i.e., exons and introns) are utilized depending on whether the task is data storage or transport. Intron regions are suitable for transportation since they are not transcribed and are removed by splicing[22,23] during transcription. This property of introns provides large sequence space for concealing data, creating potential covert channels. In contrast, data storage (watermarking) requires data to be resistant to degradation or truncation. Exons are a suitable candidate for storage because underlying DNA sequence is conserved after the translation and transcription processes[24]. These two components of internal structure components in eukaryote genes are involved in DNA steganography as the payload (watermarking) or carrier (covert channels). Fig. 2 shows the learned representations of introns and exons which are calculated by softmax function. The softmax function reduces the outputs of intron and exons to range between 0 and 1. The 2D projection position of introns and exons will change if hidden messages are embedded without considering shared patterns between the genetic components (e.g., complementary pair rules). Thus, the construction of a classification model to enable a clear separation axis of these shared patterns is an important factor in the detection of hidden messages.

3. Methods

Our proposed method uses RNNs[25] to detect hidden messages in DNA. Fig. 3 shows our proposed steganalysis pipeline. The pipeline comprises of training and detection phases. In the model training phase, the model learns the distribution of unmodified genome sequences that distinguishes between introns and exons (see Section 3.2 for the model architecture). In the detection phase, we obtain a prediction score exhibiting the distribution of introns and exons. By exploiting the obtained prediction score, we formulate a detection principle. The details of the detection principle are described in Section 3.1.

3.1. *Proposed DNA Steganalysis Principle*

The security of the random oracle is based on an *experiment* E involving an adversary $\mathcal{A}$, as well as $\mathcal{A}$'s indistinguishability of the encryption. Assume that we have the random oracle that acts like a current steganography scheme S with only a negligible success probability.

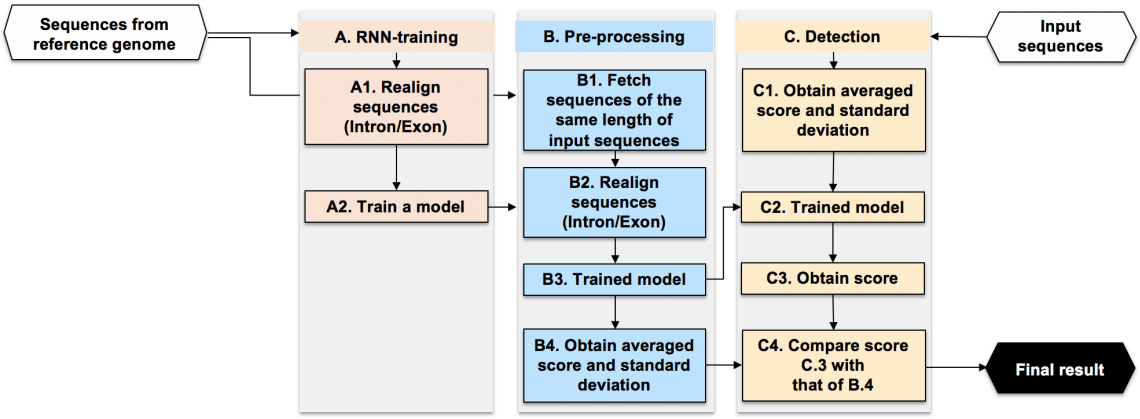


Fig. 3. Flowchart of proposed DNA steganalysis pipeline.

The experiment E can be defined for any encryption scheme S over message space $\mathbf{D}$ and for adversary $\mathcal{A}$. We describe the proposed method to detect hidden messages using the random oracle. For the E, the random oracle chooses a random steganography scheme S. Scheme S modifies or extends the process of mapping a sequence with length n input to a sequence with length ℓ with a random sequence as the output. The process of mapping sequences can be considered as a table that indicates for each possible input m the corresponding output value $\hat{m}$. With chosen scheme S, $\mathcal{A}$ chooses a pair of sequences $m_0, m_1 \in D_i$. The random oracle which posits the scheme S selects a bit $b \in \{0, 1\}$ and sends encrypted message $S(m_b) \rightarrow \hat{m}$ to the adversary. The adversary outputs a bit b'. Finally, the output of the E is defined as 1 if $b' = b$, and 0 otherwise. $\mathcal{A}$ succeeds in the E in the case of distinguishing m_b. Our methodology using E is described as follows:

(i) We construct M_{D_i} (Fig. 3-A) that runs on a random oracle where selected species $D_i \in \mathbf{D}$. Note that a model M can be based on any classification model, but the key to select a model is to reduce the standard deviation. Our proposed model M is described in Section 3.2.
(ii) $\mathcal{A}$ computes y (Fig. 3-B4) using $M_{D_i}(m)$ given $m \in D_i$.
(iii) $\mathcal{A}$ computes the standard deviation ϵ of y (Fig. 3-B).
(iv) $\mathcal{A}$ computes $\hat{y}$ (Fig. 3-C3) using $M_{D_i}(\hat{m})$ given $\hat{m} \in \hat{D}_i$.
(v) $\hat{m}$ is successfully detected (Fig. 3-C4) if

$$|\overline{y} - \hat{y}| > \epsilon. \tag{1}$$

This gives two independent scores y and $\hat{y}$ from M_{D_i}. The score y will have the same range of the unmodified genome sequences whereas the score $\hat{y}$ will have a different range of modified genome sequences. If the score difference between y and $\hat{y}$ is larger than the standard deviation of the unmodified genome sequence distribution, it may be that the sequence has been forcibly changed. Fig. 4 shows the histogram of the final score of y and $\hat{y}$ returned from softmax of the neural network. If the message is hidden, we can see that the final score from softmax of the neural network differs over the range $\overline{y} \pm \epsilon$. From Eq. (1) below, we show that detection is possible using information theoretical proof based on entropy H (Ref.[26]).

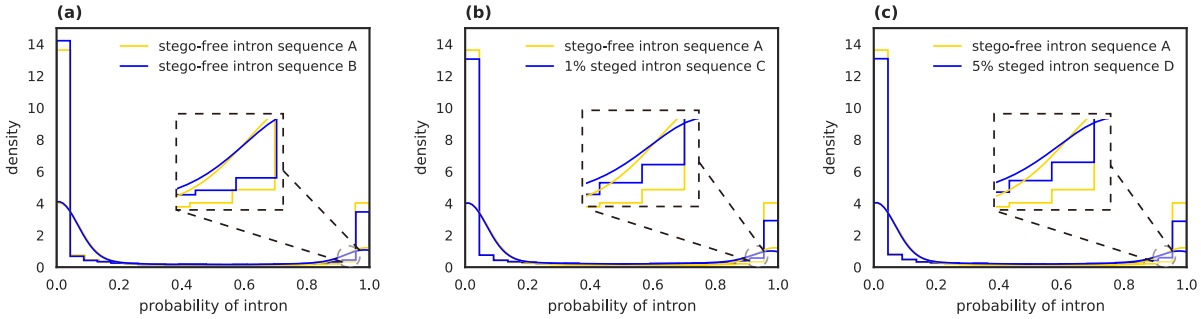


Fig. 4. Final score of intron/exon sequence obtained from the softmax of the neural network (best viewed in color). (a) kernel density differences between two stego-free intron sequences (b) kernel density differences between stego-free and 1% perturbed steged intron sequences. (c) kernel density differences between stego-free and 5% perturbed steged intron sequences.

Lemma 1. *A DNA steganography scheme is not secure if* $H(\mathbf{D}) \neq H(\hat{\mathbf{D}}|\mathbf{D})$.

Proof. The mutual joint entropy $H(\mathbf{D}, \hat{\mathbf{D}}) = H(\mathbf{D}) + H(\hat{\mathbf{D}}|\mathbf{D})$ is the union of both entropies for distribution $\mathbf{D}$ and $\hat{\mathbf{D}}$. According to Gallager at el[27], the mutual information of $I(\mathbf{D}; \hat{\mathbf{D}})$ is given as $I(\mathbf{D}; \hat{\mathbf{D}}) = H(\mathbf{D}) - H(\mathbf{D}|\hat{\mathbf{D}})$. It is symmetric in $\mathbf{D}$ and $\hat{\mathbf{D}}$ such that $I(\mathbf{D}; \hat{\mathbf{D}}) = I(\hat{\mathbf{D}}; \mathbf{D})$, and always non-negative. The conditional entropy between two distribution is 0 if and only if the distributions are equal. Thus, the mutual information must be zero to define secure DNA steganography schemes:

$$I(\mathbf{C}; (\mathbf{D}, \hat{\mathbf{D}})) = H(\mathbf{C}) - H(\mathbf{C}|(\mathbf{D}, \hat{\mathbf{D}})) = 0. \tag{2}$$

where $\mathbf{C}$ is message hiding space and it follows that:

$$H(\mathbf{C}) = H(\mathbf{C}|(\mathbf{D}, \hat{\mathbf{D}})). \tag{3}$$

Eq. (2) indicates that the amount of entropy $H(\mathbf{C})$ must not be decreased based on the knowledge of $\mathbf{D}$ and $\hat{\mathbf{D}}$. It follows that the secure steganography scheme is obtained if and only if:

$$\forall_i \in \mathbb{N}, m_i \in \mathbf{D}, \hat{m}_i \in \hat{\mathbf{D}} : m_i = \hat{m}_i. \tag{4}$$

Note that for $m_i = \hat{m}_i$ it is not possible to distinguish between the original sequence and the stego sequence. Considering that the representations of $\hat{m}$ are limited to {`A,C,G,T`}, it is nearly impossible to satisfy the condition because current steganography schemes are all based on the assumption of addition or substitution. Because $\mathbf{C}$ is independent of $\mathbf{D}$, the amount of information will increase over distribution $\mathbf{D}$ if hidden messages are inserted over distribution $\hat{\mathbf{D}}$. We can conclude that the schemes are not secure under condition $H(\mathbf{C}) > H(\mathbf{C}|(\mathbf{D}, \hat{\mathbf{D}}))$. □

3.2. *Proposed Steganalysis RNN Model*

The proposed model is based on sequence-to-sequence learning using an autoencoder and stacked RNNs[28], where the model training consists of two main steps: 1) unsupervised pre-training of sequence-to-sequence autoencoder for modeling an overcomplete case, and 2) supervised fine-tuning of stacked RNNs for modeling patterns between canonical and non-canonical

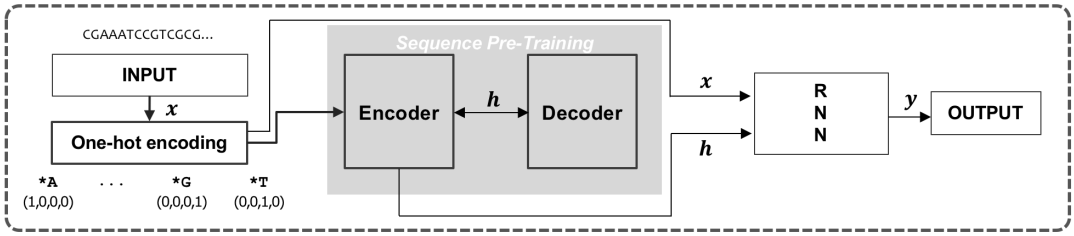


Fig. 5. Overview of proposed RNN methodology.

splice sites (see Fig. 5). In the proposed model, we use a set of DNA sequences labeled as introns and exons. These sequences are converted into a binary vector by orthogonal encoding[29]. It employs n_c-bit one-hot encoding. For $n_c = 4$, {`A,C,T,G`} is encoded by

$$\langle [1,0,0,0], [0,1,0,0], [0,0,1,0], [0,0,0,1] \rangle. \tag{5}$$

For example, the sequence `ATTT` is encoded into a 4×4 dimensional binary vector $\langle [1,0,0,0], [0,0,0,1], [0,0,0,1], [0,0,0,1] \rangle$. The encoded sequence is a tuple of a four-dimensional (4D) dense vector, and is connected to the first layer of an autoencoder, which is used for the unsupervised pre-training of sequence-to-sequence learning. An autoencoder is an artificial neural network (ANN) that is used to learn meaningful encoding for a set of data in a case involving unsupervised learning. An autoencoder consists of two components, namely an encoder and decoder.

The encoder RNN encodes $\mathbf{x}$ to the representation of sequence features $\mathbf{h}$, and the decoder RNN decodes $\mathbf{h}$ to the reconstructed $\hat{\mathbf{x}}$; thus minimizing the reconstruction errors of $\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2$, where $\mathbf{x}$ is one-hot encoded input. Through unsupervised learning of the encoder-decoder model[30], we obtain representations of inherent features $\mathbf{h}$, which are directly connected to the second activation layer. The second layer is RNNs layer used to construct the model. The model in turn is used to determine patterns between canonical and non-canonical splice signals. We then obtain the tuple of fine-tunned $\mathbf{h} =< \mathbf{h_1}, \cdots, \mathbf{h_d} >$, where $\mathbf{h}$ is the representation of sequence features learned by features, which is a representation of introns and exons in hidden layers, and $\mathbf{d}$ is the dimension of a vector.

The features $\mathbf{h}$ learned from the autoencoder are connected to the second stacked RNN layer, which consists of our proposed architecture for outputting a classification score for the given sequence $D_i \in \mathbf{D}$. For the fully connected output layer, we use the sigmoid function as the activation. The activation score is given by $\Pr(y = i|\mathbf{h}) = \frac{1/(1+\exp(-\mathbf{w}_i^T \mathbf{h}))}{\sum_{k=0}^{1} 1/(1+\exp(-\mathbf{w}_k^T \mathbf{h}))}$, where y is the label that indicates whether the given region contains introns ($y = 1$) or exons ($y = 0$). For our training model, we use a recently proposed optimizer of multi-class logarithmic loss function Adam[31]. The objective function $\mathcal{L}(\mathbf{w})$ that must be minimized is defined as follows:

$$\mathcal{L}(\mathbf{w}) = -\frac{1}{N} \sum_{n=1}^{N} (y_i \log(p_i) + (1 - y_i) \log(1 - p_i)) \tag{6}$$

where N is the mini-batch size. A model $\mathbf{M}_{D_i}$ has a possible score of p_i for one species, where p_i is the score of given non perturbed sequences.

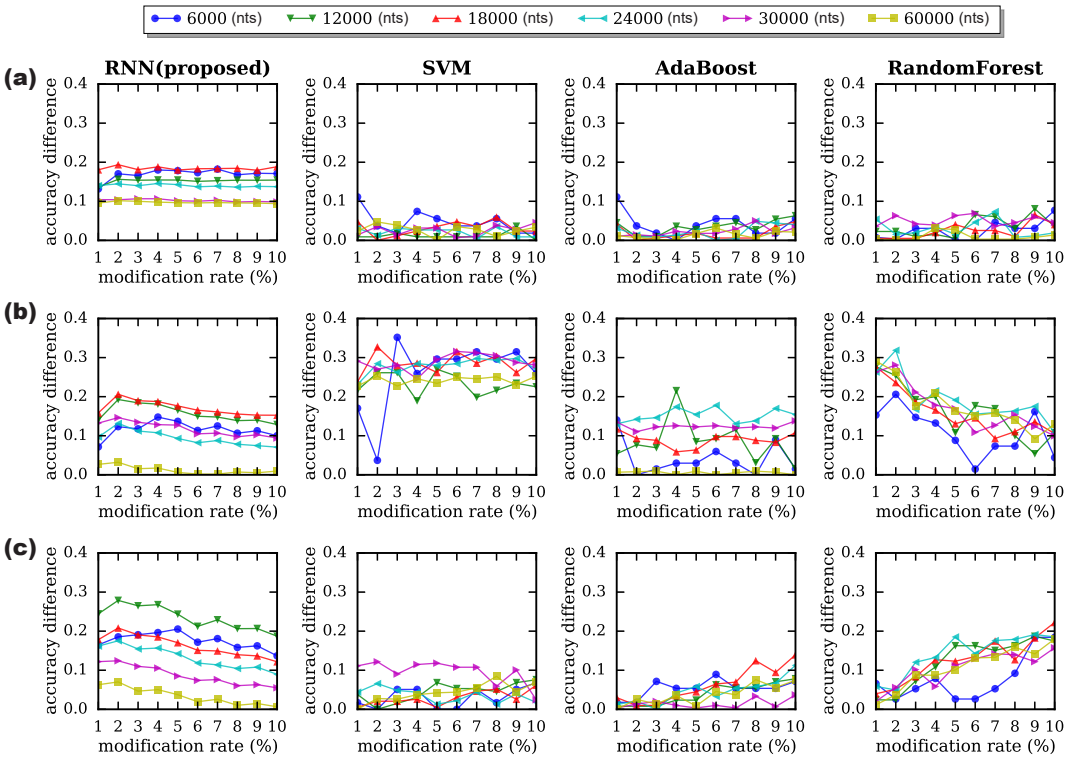


Fig. 6. Comparison of learning algorithms with random hiding algorithms (best viewed in color). (a) differences in accuracy for intron region (b) differences in accuracy for exon region (c) difference in accuracy for both region. [The performances of four supervised learning algorithms when detecting hidden messages are shown for six variable lengths of nucleotides (nts).]

4. Results

4.1. *Dataset*

We simulated our approach using the Ensembl human genome dataset and human UCSC-hg38 dataset[32], which include sequences from 24 human chromosomes (22 autosomes and 2 sex chromosomes). The Ensembl human genome dataset has a two-class classification (coding, and non-coding) and the UCSC-hg38 dataset has a three-class classification (donor, acceptor, and non-site).

4.2. *Input Representation*

The machine learning approach typically employs a numerical representation of the input for downstream processing. Orthogonal encoding, such as one-hot coding[29], is widely used to convert DNA sequences into a numerical format. It employs n_c-bit one-hot encoding. For $n_c = 4$, {`A,C,T,G`} is encoded as described in Eq. (5). According to Lee et al.[17], the vanilla one-hot encoding scheme tends to limit generalization because of the sparsity of its encoding (75% of the elements are zero). Thus, our approach encodes nucleotides into a 4D dense vector that follows the direct architecture of a normal neural network layer[33], which is trained by the gradient decent method.

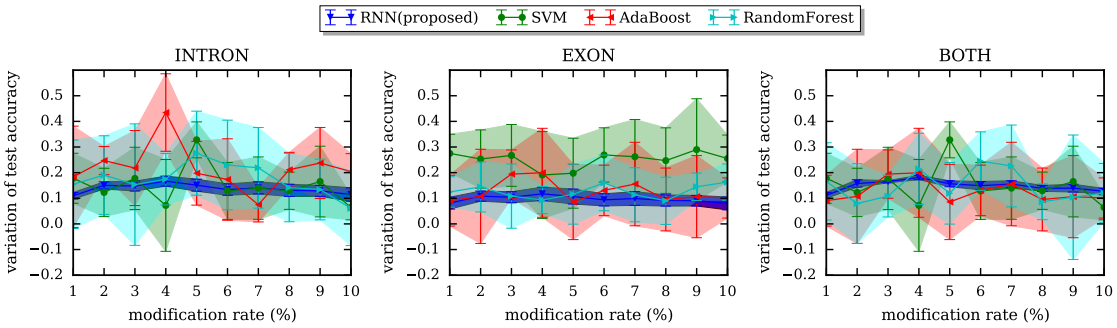


Fig. 7. Comparison of learning algorithms in terms of robustness (best viewed in color). Mean and variance of accuracy are measured for the fixed DNA sequence length of 6000 for 500 cases by changing one percent of the hidden message. The shaded line represents the standard deviation of the inference accuracy.

4.3. *Model Training*

The proposed RNN-based approach uses unsupervised training for the autoencoder and supervised training for the fine-tuning. The first layer of unsupervised training uses 4 input units, 60 hidden RNNs units with 50 epochs and 4 output units that are connected to the second layer. The second layer of supervised training uses 4 input units that are connected to stacked LSTM layers with full version including forget gates and peephole connections. The 4 input layers are used for 60 hidden units with 100 epochs, and the 4 output units are a fully connected output layer containing K units for K-class prediction.

In our experiment, we used $K = 2$ to classify sequences (coding or non-coding). For the fully connected output layer, we used the softmax function to classify sequences and the sigmoid function to classify sites for the activation. For our training model, we used a recently proposed optimizer of multi-class logarithmic loss function Adam[31]. The objective function $\mathcal{L}(\mathbf{w})$ that has to be minimized is as described in Eq (6). We used a batch size of 100 and followed the batch normalization[34]. We initialized weights according to a uniform distribution as directed by Glorot and Bengio[35]. The training time was approximately 46 hours and the running time was less than 1 second (Ubuntu 14.04 on 3.5GHz i7-5930K and 12GB Titan X).

4.4. *Evaluation Procedure*

For evaluation of performance, we used the score obtained from the softmax of the neural network. We exploited the state-of-the-art algorithm[2] to embed hidden messages for the message hiding. We randomly selected DNA sequences from the validation set using the Ensembl human genome dataset. We obtained the score of the stego-free sequence from the validation set. In the next step, we embedded hidden messages to a selected DNA sequence from the validation set, and we obtained the score. Using the score distribution of the stego-free and steged sequences, we evaluated the different scores for the range $\overline{y} \pm \epsilon$. The output from softmax of the neural network is expected to have a similar score distribution as the unmodified genome sequences. However, the score distribution changes if messages are embedded. As shown in Fig. 4(b) and Fig. 4(c), modified sequences are distinguishable using our RNNs model.

Table 1. Detection performance of sequence alignment and denoising tools.

	Both Region (%)	**Intron Region (%)**	**Exon Region (%)**
RNN (proposed)	**99.93**	**99.96**	**99.94**
BLAST[36]	84.00	85.00	85.00
Coral[37]	0.00	0.00	0.00
Lighter[38]	0.00	0.00	0.00

4.5. *Performance Comparison*

We evaluated the performance of our proposed method based on four supervised learning algorithms (RNNs, SVM, random forests, and adaptive boosting) to detect hidden messages. For the performance metric, we used the differences in accuracy.[a] Using the prediction performance data, we evaluated learning algorithms with respect to the following three regions; introns dedicated, exons dedicated, and both regions together.

For each algorithm, we generated simulated data for different lengths of DNA sequences (6000, 12000, 18000, 24000, 30000, and 60000) using the UCSC-hg38 dataset[32]. We also randomly selected 1000 cases for the fixed DNA sequence length for the modification rate 1 to 10%. Using selected DNA sequences, we obtained the average prediction accuracy of different numbers of samples against non-perturbed DNA sequences for 1000 randomly selected cases. In the next step, we obtain the prediction accuracy for the modified data generated according to the hiding algorithms. Using the averaged prediction accuracy for both the perturbed and non-perturbed cases, we evaluated the differences between the prediction accuracy rates for varying different numbers of samples. We carried out five-fold cross-validation to obtain the mean/variance of the differences in accuracy.

Fig. 6 shows an experiment for each algorithm using six variable DNA sequence lengths. Each algorithm was compared to three different regions based on the six variable DNA sequence lengths. The experiments were conducted by changing from one to then percent of the hidden message. SVM showed good detection performance in the exon region, but showed inferior performance in the intron as well as both regions category. In the case of adaptive boosting, the detection performance was similar in both regions and in intron only categorie, but performed poorly in exon regions. In the case of the random forest, the cases with the exon and both regions showed good performance except for some modification rates. In the intron regions, the detection performance was similar to that of other learning algorithms. Notably, our proposed methodology based on RNNs outperformed all of the existing hidden messages detection algorithms for all genomic regions evaluated.

In addition, we examined our proposed methodology based on denoising methods using Coral[37] and Lighter[38]. The UCSC-hg38 dataset was used to preserve local base structures and perturbed data samples were used as random noise. As shown in Table 1, the results showed that both Coral and Lighter missed detection for all modification rates in all regions. In addition, the sequence alignment method performed poorly. The results suggest that there is a 15 to 16% chance that hidden messages may not be detected in all three regions.

[a]Accuracy $= (TP + TN)/(TP + TN + FP + FN)$, where TP, FP, FN, and TN represent the numbers of true positives, false positives, false negatives, and true negatives, respectively.

To validate the learning algorithms with respect to robustness, we tested them with a fixed DNA sequence length of 6000 with 500 cases for each modification rate to measure the mean and variance of the test accuracy. Fig. 7 shows how the performance measures (mean and variance of accuracy differences) change for modification rates ranging from 1 to 10 in the intron, exon, and both regions categories. The plotted entries represents the the averaged mean over the 500 cases, and shade lines show the average of the variances over the 500 cases. The results indicate that hidden messages may not be detected if the prediction difference is less than the variance. The overall analysis with respect to the robustness showed that the learning algorithms of SVM, random forests and adaptive boosting performed poorly.

5. Discussion

The development of next-generation sequencing has reduced the price of personal genomics[39], and the discovery of the CRIPSPR-Cas9 gene has provided unprecedented control over genomes of many species[40]. While the technology is yet to be applied to simulations involving artificial DNA, human DNA sequences may become an area in which we can apply DNA watermarking. Our experiments using the real UCSC-hg38 human genome implicitly consider that unknown relevant sequences are also detectable because of the characteristics of similar patterns in non-canonical splice sites. The number of donors with `GT` pairs and acceptors with `AG` pairs were found to be 86.32% and 84.63%, respectively[16]. Existing steganography techniques modify several nucleotides. Considering few single nucleotide modifications, we can transform DNA steganography to the variant calling problem. In this regard, we believe that our methodology can be extended to the field of variant calling.

Although there are many advantages to using machine learning techniques to detect hidden messages[41–43], the following improvements are required: parameter tuning is dependent on the steganalyst, e.g., the training epochs, learning rate, and size of the training set; the failure to detect hidden messages cannot be corrected by the steganalyst. However, we expect that the future development of such techniques will resolve the limitations. According to Alvarez and Salzmann[44], the numbers of layers and neurons of deep networks can be determined using an additional class of methods, sparsity regularization, to the objective function. The sizes of vectors of grouped parameters of each neuron in each layer incur penalties if the loss converges. The affected neurons are removed if the neurons are assigned a value of zero.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT) [2014M3C9A3063541, 2018R1A2B3001628], and the Brain Korea 21 Plus Project in 2018.

References

1. K. Bennett (Citeseer, 2004).
2. B. A. Mitras and A. Abo, *International Journal of Information Technology and Business Management* **14**, 96 (2013).
3. M. B. Beck, E. C. Rouchka and R. V. Yampolskiy, 204 (2012).

4. A. Gehani, T. LaBean and J. Reif, 167 (2003).
5. H. J. Cordell and D. G. Clayton, *The Lancet* **366**, 1121 (2005).
6. S. Katzenbeisser and F. Petitcolas (Artech house, 2000).
7. C. T. Clelland, V. Risca and C. Bancroft, *Nature* **399**, 533 (1999).
8. N. Yachie, K. Sekiyama, J. Sugahara, Y. Ohashi and M. Tomita, *Biotechnology progress* **23**, 501 (2007).
9. D. G. Gibson, J. I. Glass, C. Lartigue, V. N. Noskov, R.-Y. Chuang, M. A. Algire, G. A. Benders, M. G. Montague, L. Ma, M. M. Moodie *et al.*, *science* **329**, 52 (2010).
10. S. Brenner, S. R. Williams, E. H. Vermaas, T. Storck, K. Moon, C. McCollum, J.-I. Mao, S. Luo, J. J. Kirchner, S. Eletr *et al.*, *Proceedings of the National Academy of Sciences* **97**, 1665 (2000).
11. K. Tanaka, A. Okamoto and I. Saito, *Biosystems* **81**, 25 (2005).
12. D. Heider and A. Barnekow, *BMC bioinformatics* **8**, p. 176 (2007).
13. S. Jiao and R. Goutte, (2008).
14. I. K. Maitra, *Journal of Global Research in Computer Science* **2** (2011).
15. K. Grosse, P. Manoharan, N. Papernot, M. Backes and P. McDaniel, *arXiv preprint arXiv:1702.06280* (2017).
16. T. Lee and S. Yoon *International Conference on Machine Learning* 2015.
17. B. Lee, T. Lee, B. Na and S. Yoon, *arXiv preprint arXiv:1512.05135* (2015).
18. L. v. d. Maaten and G. Hinton, *Journal of Machine Learning Research* **9**, 2579 (2008).
19. R. Anderson (Springer Science & Business Media, 1996).
20. R. Canetti, O. Goldreich and S. Halevi, *Journal of the ACM (JACM)* **51**, 557 (2004).
21. M. Bellare and P. Rogaway, 62 (1993).
22. H. Keren, G. Lev-Maor and G. Ast, *Nature Reviews Genetics* **11**, 345 (2010).
23. D. J. Lockhart and E. A. Winzeler, *Nature* **405**, 827 (2000).
24. B. Shimanovsky, J. Feng and M. Potkonjak, 373 (2002).
25. J. Schmidhuber, *Neural networks* **61**, 85 (2015).
26. R. E. Blahut (Addison-Wesley Longman Publishing Co., Inc., 1987).
27. R. G. Gallager, *Information theory and reliable communication* (Springer, 1968).
28. S. M. Peterson, J. A. Thompson, M. L. Ufkin, P. Sathyanarayana, L. Liaw and C. B. Congdon, *Frontiers in genetics* **5**, p. 23 (2014).
29. P. Baldi and S. Brunak, *Bioinformatics: the machine learning approach* (MIT press, 2001).
30. N. Srivastava, E. Mansimov and R. Salakhutdinov, 843 (2015).
31. D. Kingma and J. Ba, *arXiv preprint arXiv:1412.6980* (2014).
32. W. J. Kent, C. W. Sugnet, T. S. Furey, K. M. Roskin, T. H. Pringle, A. M. Zahler and D. Haussler, *Genome research* **12**, 996 (2002).
33. F. Chollet *et al.*, *URL: https://keras. io/k* **7** (2015).
34. S. Ioffe and C. Szegedy, *arXiv preprint arXiv:1502.03167* (2015).
35. X. Glorot and Y. Bengio, in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010.
36. S. F. Altschul, W. Gish, W. Miller, E. W. Myers and D. J. Lipman, *Journal of molecular biology* **215**, 403 (1990).
37. L. Salmela, *Bioinformatics* **26**, 1284 (2010).
38. L. Song, L. Florea and B. Langmead, *Genome biology* **15**, p. 509 (2014).
39. S. C. Schuster, *Nature methods* **5**, p. 16 (2008).
40. P. D. Hsu, E. S. Lander and F. Zhang, *Cell* **157**, 1262 (2014).
41. S. Lyu and H. Farid, **5306**, 35 (2004).
42. S. M. Erfani, S. Rajasegarar, S. Karunasekera and C. Leckie, *Pattern Recognition* **58**, 121 (2016).
43. S. Min, B. Lee and S. Yoon, *Briefings in bioinformatics* **18**, 851 (2017).
44. J. M. Alvarez and M. Salzmann, in *Advances in Neural Information Processing Systems*, 2016.

Bi-directional Recurrent Neural Network Models for Geographic Location Extraction in Biomedical Literature

Arjun Magge[1,2] and Davy Weissenbacher[3] and Abeed Sarker[3] and Matthew Scotch[1,2] † and Graciela Gonzalez-Hernandez[3]

[1] *College of Health Solutions,* [2] *Biodesign Center for Environmental Health Engineering, Arizona State University, Tempe, AZ 85281, USA*

[3] *Department of Biostatistics, Epidemiology and Informatics, The Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA 19104, USA.*

†*E-mail: Matthew.Scotch@asu.edu*

Phylogeography research involving virus spread and tree reconstruction relies on accurate geographic locations of infected hosts. Insufficient level of geographic information in nucleotide sequence repositories such as GenBank motivates the use of natural language processing methods for extracting geographic location names (toponyms) in the scientific article associated with the sequence, and disambiguating the locations to their co-ordinates. In this paper, we present an extensive study of multiple recurrent neural network architectures for the task of extracting geographic locations and their effective contribution to the disambiguation task using population heuristics. The methods presented in this paper achieve a strict detection F_1 score of 0.94, disambiguation accuracy of 91% and an overall resolution F_1 score of 0.88 that are significantly higher than previously developed methods, improving our capability to find the location of infected hosts and enrich metadata information.

Keywords: Named Entity Recognition; Toponym Detection; Toponym Disambiguation; Toponym Resolution; Natural Language Processing; Deep Learning;

1. Introduction

Nucleotide sequence repositories like GenBank contain millions of records from various organisms collected around the world that enables researchers to perform phylogenetic tree and spread reconstruction. However, a vast majority of the records (65-80%)[1,2] contain geographic information that is deemed to be at an insufficient level of granularity; information that is often present in the associated published article. This motivates the use of natural language processing (NLP) methods to find the geographic location (or toponym) of infected hosts in the full text. In NLP, this task of detecting toponyms from unstructured text, and then disambiguating the locations to their co-ordinates is formally known as toponym resolution.

Toponym resolution in scientific articles can be used to obtain precise geospatial metadata of infected hosts which is highly beneficial in building transmission models in phylogeography that could enable public health agencies to target high-risk areas. Improvement in geospatial metadata also enriches other scientific studies that utilize GenBank data, such as those in population genetics, environmental health, and epidemiology in general, as geographic location

is often used in addition to or as a proxy of other demographic data. Toponym Resolution is typically accomplished in two stages (1) toponym detection (geotagging), a named entity recognition (NER) task in NLP and (2) toponym disambiguation (geocoding).

For instance, given the sentence *"Our study mainly focused on pediatric cases with different outcomes from the most populated city in Argentina and one of the hospitals in Buenos Aires where patients are most often referred."*, the detection stage deals with extracting the locations *"Argentina"* and *"Buenos Aires"*.[3] The disambiguation stage deals with assigning the most likely, unique, identifiers from gazetteer resources like Geonames[a] to each location detected e.g. *"3865483:Argentina"* from 145 candidate entries containing the same name and *"3435910:Buenos Aires"* from 943 candidate entries with variations of the same name. Both tasks bring forth interesting NLP challenges with applications in a wide number of areas.

In this work, we present a system for toponym detection and disambiguation that improves substantially over previously published systems for this task, including our own.[4–6] Since detection is the first step in the process, its impact on the overall performance of the combined task is multiplied, as locations not detected can never be disambiguated. We use recurrent neural network (RNN) architectures that use word embeddings, character embeddings and case features as input for performing the detection task. In addition to these, we also experiment with the use of conditional random fields (CRF) on the output layer as they have known to improve performance. We perform ablation studies/leave-one-out analysis with repetitive runs with different seed values for drawing strong conclusions about the use of deep recurrent neural networks, their architectural variations and common features. We evaluate the impact of the results from the detection task on the upstream disambiguation task, performed using the commonly assumed *population heuristic*[7] whereby the location with the greatest population is chosen as the correct match.

The rest of the document is structured as follows. In Section 2, we summarize research efforts in the area of toponym detection and disambiguation and list the contributions of this paper in light of previous work. We distinguish the RNN architectures used for evaluation along with the population heuristic used for measurement in Section 3. Finally, we present and discuss the results of the toponym detection and disambiguation experiments in Sections 4 and discuss limitations and scope for improvements in Section 5.

2. Related Work

Toponym detection and toponym disambiguation have been widely researched by the NLP community, with numerous publications on both detection and disambiguation tasks.[8–10] Toponym detection is commonly tackled as a NER challenge where toponyms are recognized among other named entities like organization names and people's names. Previous studies[11] have identified the performance of the NER as an important source of errors in enhancing geospatial metadata in GenBank, motivating the development of tools for performing detection and resolution of named entities such as infected hosts and geographical locations.[12,13] The annotated dataset used in this work[4,11] includes both span and normalized Geonames ID

[a]http://www.geonames.org/ Accessed:Sept 30 2018

annotations. Since the performance of the overall resolution task is deeply influenced by the NER, some of the previous works using this dataset have looked specifically at improving the NER's performance. Our previous research on toponym detection have used rule-based methods,[4] traditional machine learning sequence taggers using conditional random fields (CRF)[5] and deep learning methods using feed forward neural networks.[6] NER performance since the introduction of the dataset has increased from an F1-score of 0.70 to 0.91 closing in on the human-level annotation agreement of 0.97. In the previous baseline for toponym resolution[4] a rule based extraction system was used to detect toponyms. In subsequent work, traditional machine learning algorithms such as conditional random fields (CRFs)[5] and feedforward neural nets[6] were introduced for improving the NER's performance. There exist some studies involving RNN experiments that explore the use of RNN architectures for sequence tagging tasks in the generic domain.[14,15] While these tasks measure the performance on specific tasks, the effect of optimal performances haven't been measured in upstream tasks.

On the other hand, toponym disambiguation has been commonly tackled as an information retrieval challenge by creating an inverted index of Geonames entries.[4,16] Given a toponym, candidate locations are first retrieved based on words used in the toponym and subsequently heuristics are used to pick the most appropriate location. Popular techniques use metrics such as entity co-occurrences, similarity measures, distance metrics, context features and topic modeling.[7,16–20] This approach is largely adopted due the large number of Geonames entries (about 12 million) to choose from. We also find that the most common baseline used for measuring the disambiguation performance is the population heuristic where the place with the most population is chosen as the correct match. Most research articles that focus specifically on the disambiguation problem use Stanford-NER or the Apache-NER tool[20–22] for detection which has been trained on datasets like CoNLL-2003, ACE-2005 and MUC. Some studies assume gold standard labels and proceed with the task of disambiguation which makes it difficult to assess the strength of the overall system. It is also important to note that a majority of efforts have been focused on texts from a general domain like Wikipedia or news articles.[20–22] Only a handful of publications deal with the problem in other domains like biomedical scientific articles[4,23] which contain a different and broader vocabulary. Similar to the previous disambiguation method developed for this dataset,[4] we build an inverted index using Geonames entries but use term expansion techniques to improve the performance and usability of the system in various contexts.

In light of previous work, the main contributions of this work can be summarized as follows:

(i) We perform a comprehensive and systematic evaluation of multiple RNN architectures from over 400 individual runs for the task of toponym detection in scientific articles and arrive at state-of-the-art results compared to previous methods.
(ii) We discuss the impact of significant performance improvement in toponym detection in the upstream task of toponym resolution.

3. Methods

Our approach for detection and disambiguation of geographic locations are tackled independently, as described in the following subsections. For the purposes of training and evaluation,

we use the publicly available human annotated corpus of 60 full-text PMC articles containing 1881 toponyms.[4] Of the 60, the standard test set for the corpus includes only 12 articles containing a total of 285 toponyms, a large majority of which are countries and major locations. The annotated dataset contains both span annotations and gazetteer ID annotations linking ISO-3166-1 codes for countries and GeonamesIDs for the remaining toponyms. For uniformity, we converted all ISO-3166-1 codes to equivalent GeonameIDs.

3.1. *Toponym Detection*

The task of toponym detection typically involves identifying the spans of the toponyms in an NER task where the sequence of actions is illustrated in Fig 1. As input features, we use publicly available pre-trained word embeddings that were trained on Wikipedia, PubMed abstracts and PubMed Central full text articles.[24] In addition to word embeddings, we experiment with orthogonal features such as (1) a case feature to explicitly distinguish all-uppercase, all-lowercase and camel-case words encoded as one-hot vectors that are appended to the word, and (2) fixed length character embeddings. Character embeddings have shown to improve the performances of deep neural networks and are employed in few different ways. One of the popular methods used involves the use of a CNN layer[25] or an LSTM layer[26] on vectors from a randomly initialized character embeddings that are fine tuned during training appended to the input word embedding layer. During initial experiments we found that implementation of this architecture added significantly to the training time and hence we employ the use of a simpler model where character embeddings are pre-trained using word2vec and appended directly to the input layer along with word embeddings and case features.

The proposed RNN units and their variations can be used on their own for NER purposes. However, bidirectional architectures are popularly employed for NER as they have the combined capability of processing input sentences in both directions and making tagging decisions collectively using an output layer as illustrated in figure 1. In this paper, we specifically look at bi-directional recurrent architectures. It is also common to observe the use of a CRF output layer on top of the output layer of bidirectional RNN architecture. CRF's are known to add consistency in making final tagging decisions using IOB or IOBES styled annotations. We experiment between combinations of the RNN variants along with the optional features in an ablation study to identify the impact of these additive layers on the NER's performance as well as its impact on the upstream resolution task.

3.1.1. *Recurrent Neural Networks*

RNN architectures have been widely used for auto-encoders and sequence labeling tasks such as part-of-speech tagging, NER, chunking among others.[27] RNNs are variants of feedforward neural networks that are equipped with recurrent units to carry signals from the previous output y^{t-1} for making decisions at time y^t as shown in equation 1.

$$y_t = \sigma\left(W \cdot x_t + U \cdot y_{t-1} + b\right) \tag{1}$$

Here, W and U are the weight matrices and b is the bias term that are randomly initialized and updated during training. σ represents the sigmoid activation function. In practice

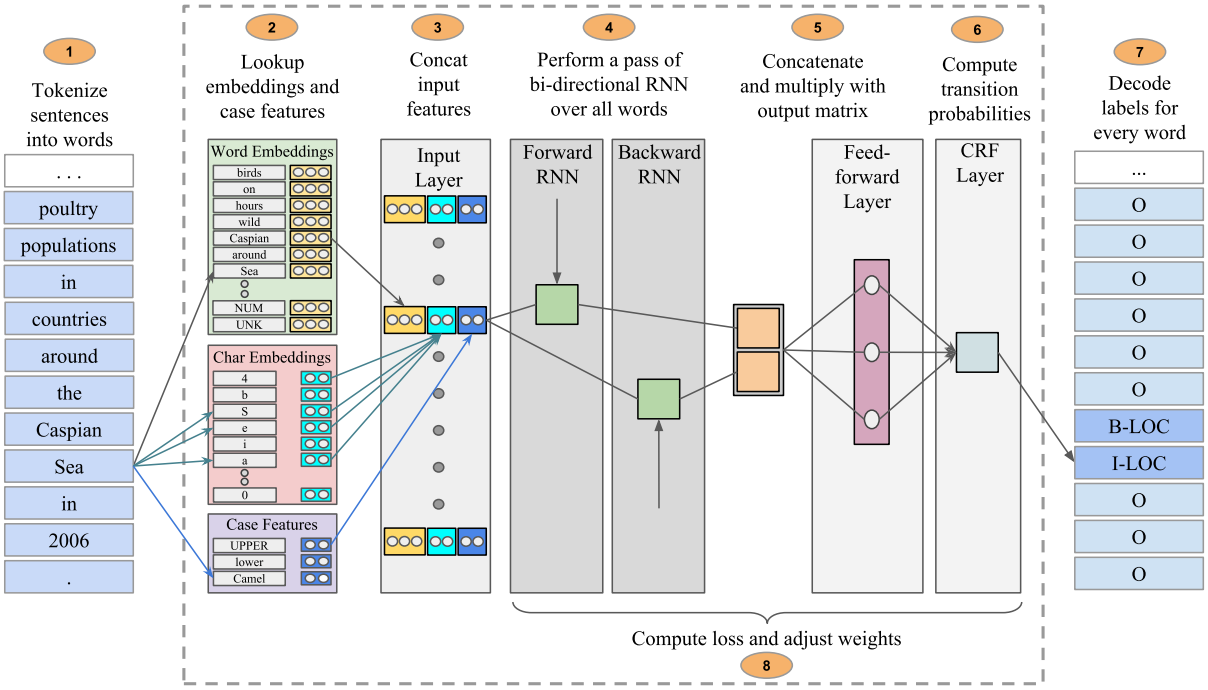


Fig. 1. A schematic representation of the sequence of actions performed in the NER equipped with bi-directional RNN layers and an output CRF layer. RNN variants discussed in this paper involve replacing RNN units with LSTM, LSTM-Peepholes, GRU and UG-RNN units.

other activation functions such as *tanh* and rectified linear units (*ReLU*) are also used. This characteristic recurrent feature simulates a memory function that makes it ideal for tasks involving sequential predictions dependent on previous decisions. However, learning long term dependencies that are necessary have been found to be difficult using RNN units alone.[28]

3.1.2. *LSTM*

LSTM networks[29] are variants of RNN that have proven to be fairly successful at learning long term dependencies. A candidate output g is calculated using an equation similar to equation 1 and further manipulated based on previous and current states of a cell that retains signals simulating long-term memory. The LSTM cell's state is controlled by *forget (f)*, *input (i)* and *output (o)* gates that control how much information flows from the input to the state and from state to the output. The gates themselves depend of current input and previous outputs.

$$g = tanh(W^g \cdot x_t + U^g \cdot y_{t-1} + b^g) \tag{2}$$

$$f = \sigma(W^f \cdot x_t + U^f \cdot y_{t-1} + b^f) \tag{3}$$

$$i = \sigma(W^i \cdot x_t + U^i \cdot y_{t-1} + b^i) \tag{4}$$

$$o = \sigma(W^o \cdot x_t + U^o \cdot y_{t-1} + b^o) \tag{5}$$

The future state of the cell c_t is calculated as a combination of (1) signals from forget gate g and the previous state of the cell c_{t-1} which determines the information to forget (or retain)

in the cell, and (2) signals from the *input* gate i and the candidate output g that determines the information from the input to be stored in the cell. Eventually the output y_t is calculated using signals from the output gate o and the current state of the cell c_t.

$$c_t = f \odot c_{t-1} + i \odot g \tag{6}$$

$$y_t = o \odot tanh(c_t) \tag{7}$$

In the above equations, $\odot$ indicates pointwise multiplication operation. While the above equations represent LSTM in its most basic form, many variations of the architecture have been introduced to simulate retention of long-term signals a few of which have been summarized in the following subsections and subsequently evaluated in the results section. For reasons of brevity, we do not include the formulas used for calculating the output y_t but they can be inferred from the works cited.

3.1.3. *Other Gated RNN Architectures*

We evaluate in our experiments one of the LSTM variations introduced for speech processing[30] that introduced the notion of peepholes (LSTM-Peep) where the idea is that state of the cell influences the *input*, *forget* and *output* gates. Here, signals for the *input* and *forget* gates i and f depend not only on the previous output y_{t-1} and current input x_t but also the previous state of the cell c_{t-1} and the *output* gate o depends on the current state of the cell c_t.

Gated Recurrent Unit (GRU)[31] also known as coupled input and forget gate LSTM (CIFG-LSTM)[15] is a simpler variation of LSTM with only two gates: update z and reset r. Their signals are determined based on the current input x and previous output y_{t-1} similar to the gates in LSTMs. The update gate z attempts to combine the functionality of input and forget gates of LSTMs i and f and eliminates the need for an output gate as well as an explicit cell state. A singular update gate signal z controls the information flow to the output value. Although it appears far more simple, GRU has gained a lot of popularity in the recent years in a variety of NLP tasks.[32,33]

Update gate RNN (UG-RNN)[34] is a much simpler variation of LSTM and GRU architectures containing only an update gate z is also included in our experiments. The importance of the update gate is often highlighted in RNN based architectures.[15] Hence, we include this model to perform a gate based ablation study to understand their contributions to the overall resolution task.

3.1.4. *Hyperparameter search and optimization*

The performance of deep neural networks relies greatly on optimization of its hyperparameters and the performance of the models have been found to be sensitive to changes in seed values used for initializing the weight matrices.[27] We first performed a grid search over the previously recommended optimal range of hyperparameter space for NER tasks[27] and to arrive at potential candidates of optimal configurations. We then performed up to 5 repetitions of experiments at the optimal setting for the model at different seed values to obtain the median performance scores. All models were developed using the TensorFlow framework and trained on NVIDIA Titan Xp GPUs equipped with an Intel Xeon CPU (E5-2687W v4).

3.2. *Toponym Disambiguation*

For toponym disambiguation, we use the Geonames gazetteer data to build an inverted index using Apache Lucene[b] and search for the toponym terms extracted in the toponym detection step in the index.

3.2.1. *Building Geonames Index*

Individual Geonames entries in the index are documents with common fields such as *GeonameID*, *LocationName*, *Latitude*, *Longitude*, *LocationClass*, *LocationCode*, *Population*, *Continent* and *AncestorNames*. Here, *LocationName* contains the common name of the place. For countries, we expand this field by using official names, ISO and ISO3 abbreviations (e.g. *United States of America*, *US* and *USA*, respectively, for *United States*). For ADM1 (Administrative Level 1) entries that have available abbreviations (e.g. *AZ* for *Arizona*, and *CA* for *California*), we add such alternate names to the *LocationName* field. In addition to the above fields we add the *County*, *State* and *Country* fields depending on the type of geoname entry. Fields such as *LocationName*, *County*, *State*, *Country* and *AncestorNames* are chosen to be reverse indexed such that partial matches of names offers the possibility of being matched with the right disambiguated toponym on a search.

3.2.2. *Searching Geonames Index*

Most cities and locations commonly have their parent locations listed as comma separated values (e.g. *Philadelphia, PA, USA*). In such cases, the index provides the capability to perform compound searches (e.g. *LocationName:"Philadelphia" AND AncestorNames:"PA, USA"*). We find that this method offers the best scalable framework for toponym disambiguation among approximately 12 million entries. Efficient search capabilities aside, the solution internally provides documents to be sorted by a particular field. In this case, we choose the *Population* field as the default sorting heuristic such that search results are sorted by highest population first. An additional motivation for the implementation of this solution is the flexibility of using external information to narrow down search results. For example, when Country information is available in the GenBank record, we can use queries like *LocationName:"Paris" AND Country:"France"* to narrow down the location of infected hosts.

4. Results and Discussion

For the NER task, we use the standard metric scores of precision, recall, and F_1-scores for toponym entities across two modes of evaluation:(1) *Strict* where the predicted spans of the toponym have to match exactly with the gold standard spans to be counted as a true positive and (2) *Overlapping* where predicted spans are true positives as long as one of its tokens overlap with gold standard annotations. For toponym disambiguation, we compare the predicted and gold standard GeonameIDs to measure precision, recall and f_1-scores as long as the spans overlap. We compare our scores with the previous systems that were trained and tested on the

[b]http://lucene.apache.org/ Accessed:Sept 30 2018

same dataset. To evaluate the performance of the overall resolution task, it is important to examine the performance of the individual systems to assess the cause of errors and identifying regions for improvement.

4.1. *Toponym Disambiguation*

Our toponym disambiguation system is unsupervised, giving us the capability to test its performance on the entire dataset assuming gold standard toponym terms to be available. Under this assumption, the accuracy of the disambiguation system was found to be 91.6% and 90.5% on training and test set respectively. Analyzing the errors, we found that comparing ids directly is a very strict mode of evaluation for the purposes of phylogeography as Geonames contains duplicate entries for many locations that belong to two or more classes of locations such as administrative division (ADM) and populated area or city (PPLA, PPLC) but refer to the same geographical location. For instance, when we look at the test set alone, which had 27 errors from a total of 285 locations, 19 appeared to be roughly the same location. These included locations like *Auckland, Lagos, St. Louis, Cleveland, Shantou, Nanchang, Shanghai, and Beijing* which were assigned the ID of the administrative unit by the system, while the annotated locations were assigned the ID of the populated area or city or *vice versa*. Given these reasons, we find that the performance of the resolution step exceeds the reported scores by 5% to arrive at an approximate accuracy of 95-96%. However, for the purposes of comparison with previous systems we report the overall resolution performance in Table 1 without making such approximations. We did however observe 8 errors where the system assigned GeonamesIDs were drastically different from their original locations due to the population heuristic. For example, a toponym of *Madison* was incorrectly assigned the ID of *Madison County, Alabama* which had a higher population than the gold standard annotation *Madison, Dane County, Wisconsin(WI)*.

4.2. *Toponym Resolution*

Analyzing the errors across the architectures, we find that 80-90% of the erroneous instances to be repeating across the RNN architectures making it challenging to use ensemble methods for reducing errors. These included false negative toponyms such as *Plateau, Borno, Ga, Gurjev, Sokoto* etc. which appear in tables and structured contexts making it difficult to recognize them. However, as discussed in our previous work,[6] we plan to handle table structures differently by employing alternative methods of conversions from pdf to text. Almost all false positives appeared to be geographic locations, however in the text they were found to be referring to other named entities like virus strains and isolates rather than toponyms.

We found that the LSTM-Peep based architecture appeared to have marginally better performance scores on the NER task and hence the overall resolution task. Feature ablation analysis shown in Figure 2 indicate that inclusion of the character embedding feature contributed to increase in the overall performance of RNN models. However, inclusion of case feature in combination with the character embeddings appeared to be redundant. Inclusion of the CRF output layer seemed to have a positive impact on most models while additive layers seemed to have more effect on GRU, LSTM and LSTM-Peep architectures.

Table 1. Median Precision(P), Recall(R) and F_1 scores for NER and Resolution. Bold-styled scores indicate highest performance. All recurrent neural network units were used in a bidirectional setup with inputs containing pre-trained word embeddings, character embeddings and case features, and an output layer with an additional CRF layer.

Method	NER-Strict			NER-Overlapping			Resolution		
	P	R	F_1	P	R	F_1	P	R	F_1
Rule-based[4]	0.58	0.876	0.698	0.599	0.904	0.72	0.547	**0.897**	0.697
CRF-All[5]	0.85	0.76	0.80	0.86	0.77	0.81	-	-	-
FFNN + DS[6]	0.90	0.93	0.91	-	-	-	-	-	-
RNN	0.910	0.891	0.901	0.931	0.912	0.922	0.896	0.817	0.855
UG-RNN	0.948	0.902	0.924	0.959	0.912	0.935	0.903	0.824	0.862
GRU	**0.952**	0.919	0.935	**0.967**	0.930	0.948	0.888	0.835	0.860
LSTM	0.932	0.926	0.929	0.954	0.947	0.950	0.892	0.842	0.866
LSTM-Peep	0.934	**0.944**	**0.939**	0.951	**0.961**	**0.956**	**0.907**	0.863	**0.884**

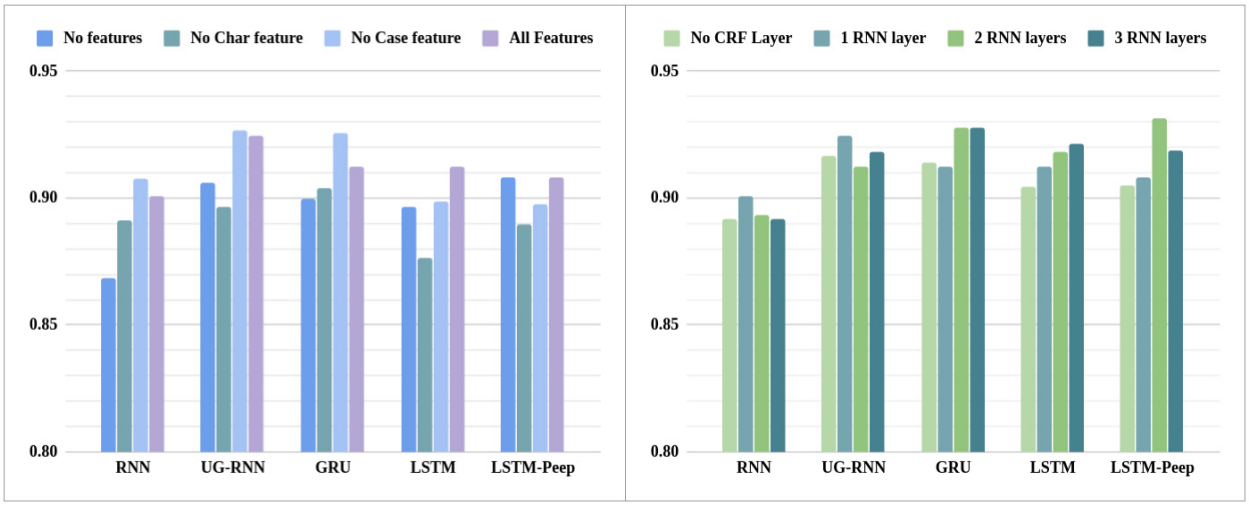


Fig. 2. *(Left)* Ablation/leave-one-out analysis showing the contribution of individual features to the NER performance across the RNN models. *(Right)* Impact of additive layers on the performance of the NER across the RNN models. Here, RNN layers refer to respective variants of RNN architectures. Y-axis shows strict F_1 scores.

5. Limitations and Future Work

In this work, we find that utilizing state-of-the-art NER architectures help us obtain performances that are inching close to human performance. However, we do find that the articles in the test set may perhaps be relatively easier than the average article for the detection task when we compare it to randomly selected validation/development set performances. As discussed in our previous work,[6] distance supervision datasets can contain toponym spans in close proximity to each other generating noisy training examples. This makes it challenging to

use distance supervision techniques to increase the size of training data for training sequence tagging models based on RNN architectures. Hence, to address this issue, we are in the process of expanding the annotation dataset from 60 articles to 150 articles for a more comprehensive training and evaluation of the system.

Irrespective of the ease of detection in the test set, there appear to be false negative toponyms (discussed in the previous section) that could possibly be the location of infected hosts(LOIH). While there are chances that toponyms that are LOIH appear repeatedly in the scientific article in varying contexts thus increasing the chances of them being detected, in our following work we wish to evaluate the impact of these false negatives on the overall task of identifying the LOIH. To reduce false positives where locations could infact refer to other named entities like virus strains and isolates than toponyms themselves, we intend to explore approaches from metonymy resolution[35] for filtering out such false positives.

6. Conclusion

Phylogeography research relies on accurate geographical metadata information from nucleotide repositories like GenBank. In records that contain insufficient metadata information, there is a motivation to extract the geographical location from the associated articles to determine the location of the infected hosts. In this work we present and evaluate methods built on recurrent neural networks that extract geographical locations from scientific articles with a substantial increase in performance from an F_1 score of 0.88 which improves significantly over the previous toponym resolution system F_1 of 0.69. Our implementations of the toponym detection and toponym disambiguation[c] systems along with the updated version of the annotations containing GeonameIDs[d] are available online.

Acknowledgments

AM designed and trained the neural networks, ran the experiments, performed the error analysis, and wrote most of the manuscript. DW and AS reviewed, restructured and contributed many sections and revisions of the manuscript. MS and GG provided overall guidance on the work and edited the final manuscript. The authors would also like to acknowledge Karen OConnor, Megan Rorison and Briana Trevino for their efforts in the annotation processes. The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. The authors are also grateful to ASU-BMI's computing resources used for conducting the experiments in the paper.

Funding

Research reported in this publication was supported by the National Institute of Allergy and Infectious Diseases (NIAID) of the National Institutes of Health (NIH) under grant number R01AI117011 to MS and GG. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

[c] `https://bitbucket.org/pennhlp/toponym-resolution-using-rnns Accessed:30 Sept 2018`
[d] `https://healthlanguageprocessing.org/software-and-downloads/ Accessed:30 Sept 2018`

References

1. M. Scotch, I. N. Sarkar, C. Mei, R. Leaman, K.-H. Cheung, P. Ortiz, A. Singraur and G. Gonzalez, Enhancing phylogeography by improving geographical information from genbank *Journal of biomedical informatics* **44** (Elsevier, 2011).
2. T. Tahsin, R. Beard, R. Rivera, R. Lauder, G. Wallstrom, M. Scotch and G. Gonzalez, Natural language processing methods for enhancing geographic metadata for phylogeography of zoonotic viruses *AMIA Summits on Translational Science Proceedings* **2014** (American Medical Informatics Association, 2014).
3. P. Barrero, M. Viegas, L. Valinotto and A. Mistchenko, Genetic and phylogenetic analyses of influenza a h1n1pdm virus in buenos aires, argentina *Journal of virology* **85** (Am Soc Microbiol, 2011).
4. D. Weissenbacher, T. Tahsin, R. Beard, M. Figaro, R. Rivera, M. Scotch and G. Gonzalez, Knowledge-driven geospatial location resolution for phylogeographic models of virus migration *Bioinformatics* **31** (Oxford University Press, 2015).
5. D. Weissenbacher, A. Sarker, T. Tahsin, M. Scotch and G. Gonzalez, Extracting geographic locations from the literature for virus phylogeography using supervised and distant supervision methods *AMIA Summits on Translational Science Proceedings* **2017** (American Medical Informatics Association, 2017).
6. A. Magge, D. Weissenbacher, A. Sarker, M. Scotch and G. Gonzalez-Hernandez, Deep neural networks and distant supervision for geographic location mention extraction *Bioinformatics* **34**2018.
7. J. L. Leidner, Toponym resolution in text: annotation, evaluation and applications of spatial grounding, in *ACM SIGIR Forum*, (2)2007.
8. M. Gritta, M. T. Pilehvar, N. Limsopatham and N. Collier, Whats missing in geographical parsing? *Language Resources and Evaluation* **52** (Springer, 2018).
9. J. L. Leidner and M. D. Lieberman, Detecting geographical references in the form of place names and associated spatial natural language *SIGSPATIAL Special* **3** (ACM, 2011).
10. R. Tobin, C. Grover, K. Byrne, J. Reid and J. Walsh, Evaluation of georeferencing, in *proceedings of the 6th workshop on geographic information retrieval*, 2010.
11. T. Tahsin, D. Weissenbacher, R. Rivera, R. Beard, M. Firago, G. Wallstrom, M. Scotch and G. Gonzalez, A high-precision rule-based extraction system for expanding geospatial metadata in genbank records *Journal of the American Medical Informatics Association* **23** (Oxford University Press, 2016).
12. T. Tahsin, D. Weissenbacher, D. Jones-Shargani, D. Magee, M. Vaiente, G. Gonzalez and M. Scotch, Named entity linking of geospatial and host metadata in genbank for advancing biomedical research *Database* **2017** (Oxford University Press, 2017).
13. T. Tahsin, D. Weissenbacher, K. Oconnor, A. Magge, M. Scotch and G. Gonzalez-Hernandez, Geoboost: accelerating research involving the geospatial metadata of virus genbank records *Bioinformatics* **34** (Oxford University Press, 2017).
14. R. Jozefowicz, W. Zaremba and I. Sutskever, An empirical exploration of recurrent network architectures, in *International Conference on Machine Learning*, 2015.
15. K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber, Lstm: A search space odyssey *IEEE transactions on neural networks and learning systems* **28** (IEEE, 2017).
16. S. Overell and S. Rüger, Using co-occurrence models for placename disambiguation *International Journal of Geographical Information Science* **22** (Taylor & Francis, 2008).
17. A. Spitz, J. Geiß and M. Gertz, So far away and yet so close: augmenting toponym disambiguation and similarity with text-based networks, in *Proceedings of the third international ACM SIGMOD workshop on managing and mining enriched geo-spatial data*, 2016.

18. Y. Ju, B. Adams, K. Janowicz, Y. Hu, B. Yan and G. McKenzie, Things and strings: improving place name disambiguation from short texts by combining entity co-occurrence with topic modeling, in *European Knowledge Acquisition Workshop*, 2016.
19. M. D. Lieberman and H. Samet, Adaptive context features for toponym resolution in streaming news, in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 2012.
20. E. Kamalloo and D. Rafiei, A coherent unsupervised model for toponym resolution, in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, 2018.
21. M. D. Lieberman and H. Samet, Multifaceted toponym recognition for streaming news, in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011.
22. J. Hoffart, Discovering and disambiguating named entities in text, in *Proceedings of the 2013 SIGMOD/PODS Ph. D. symposium*, 2013.
23. J. Tamames and V. de Lorenzo, Envmine: A text-mining system for the automatic extraction of contextual information *BMC bioinformatics* **11** (BioMed Central, 2010).
24. S. Pyysalo, F. Ginter, H. Moen, T. Salakoski and S. Ananiadou, Distributional semantics resources for biomedical text processing (2013).
25. X. Ma and E. Hovy, End-to-end sequence labeling via bi-directional lstm-cnns-crf, in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016.
26. G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami and C. Dyer, Neural architectures for named entity recognition, in *Proceedings of NAACL-HLT*, 2016.
27. N. Reimers and I. Gurevych, Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging, in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Copenhagen, Denmark, 2017).
28. Y. Bengio, P. Simard and P. Frasconi, Learning long-term dependencies with gradient descent is difficult *IEEE transactions on neural networks* **5**1994.
29. S. Hochreiter and J. Schmidhuber, Long short-term memory *Neural computation* **9** (MIT Press, 1997).
30. H. Sak, A. Senior and F. Beaufays, Long short-term memory recurrent neural network architectures for large scale acoustic modeling, in *Fifteenth annual conference of the international speech communication association*, 2014.
31. K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, Learning phrase representations using rnn encoder–decoder for statistical machine translation, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
32. Z. Che, S. Purushotham, K. Cho, D. Sontag and Y. Liu, Recurrent neural networks for multivariate time series with missing values *Scientific reports* **8** (Nature Publishing Group, 2018).
33. Y. Luo, Recurrent neural networks for classifying relations in clinical notes *Journal of biomedical informatics* **72** (Elsevier, 2017).
34. J. Collins, J. Sohl-Dickstein and D. Sussillo, Capacity and trainability in recurrent neural networks, in *Profeedings of the International Conference on Learning Representations (ICLR)*, 2017.
35. M. Gritta, M. T. Pilehvar, N. Limsopatham and N. Collier, Vancouver welcomes you! minimalist location metonymy resolution, in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017.

Automatic Human-like Mining and Constructing Reliable Genetic Association Database with Deep Reinforcement Learning

Haohan Wang[1,†], Xiang Liu[2,†], Yifeng Tao[3], Wenting Ye[1], Qiao Jin[4],
William W. Cohen[5,6,‡], Eric P. Xing[5,7]

[1]*Language Technologies Institute,*
[3]*Computational Biology Department,*
[5]*Machine Learning Department,*
Carnegie Mellon University
Pittsburgh, PA, USA
[2]*Chinese University of Hong Kong*
Shenzhen, China
[4]*Tsinghua University*
Beijing, China
[6]*Google AI*
Pittsburgh, PA, USA
[7]*Pettum Inc.*
Pittsburgh, PA, USA
E-mail: haohanw@cs.cmu.edu
[†]*Equal Contribution*
[‡]*The work is done while the author is at CMU.*

The increasing amount of scientific literature in biological and biomedical science research has created a challenge in continuous and reliable curation of the latest knowledge discovered, and automatic biomedical text-mining has been one of the answers to this challenge. In this paper, we aim to further improve the reliability of biomedical text-mining by training the system to directly simulate the human behaviors such as querying the PubMed, selecting articles from queried results, and reading selected articles for knowledge. We take advantage of the efficiency of biomedical text-mining, the flexibility of deep reinforcement learning, and the massive amount of knowledge collected in UMLS into an integrative artificial intelligent reader that can automatically identify the authentic articles and effectively acquire the knowledge conveyed in the articles. We construct a system, whose current primary task is to build the genetic association database between genes and complex traits of human. Our contributions in this paper are three-fold: 1) We propose to improve the reliability of text-mining by building a system that can directly simulate the behavior of a researcher, and we develop corresponding methods, such as Bi-directional LSTM for text mining and Deep Q-Network for organizing behaviors. 2) We demonstrate the effectiveness of our system with an example in constructing a genetic association database. 3) We release our implementation as a generic framework for researchers in the community to conveniently construct other databases.

Keywords: Biomedical text-mining, Deep Reinforcement Learning, Genetic Association

1. Introduction

Understanding the biological and biomedical science is one of the most fundamental goals of research and an essential step towards the realization of "precision medicine" in this era. Scientists all over the world are collaboratively contributing to this final goal, leading to an accompanying growth of the scientific literature. For example, PubMed[a] has seen exponential growth regarding the number of publications in recent years[1] and has collected over 27 million abstracts.[2] These massive amount of articles consequently bring in the challenge of integrating the information conveyed effectively and accurately.

Biomedical information extraction has been the answer to this challenge for a long time.[3,4] However, due to the demand of high reliability in biomedical research, following a typical general-purpose information extraction protocol and examining every article in the corpus nondiscriminatorily may lead to falsely constructed knowledge because of the non-negligible number of scientific literature with the issues of reproducibility.[5–7]

To fulfill the need of reliability in text mining and knowledge-base construction, instead of requiring the system to scan the entire corpus uniformly, we propose to train the system to directly simulate the behavior of a scientist with a sequence of actions including 1) querying the web, 2) evaluating the article, 3) studying the article for knowledge if necessary, 4) rejecting the knowledge if necessary, and 5) storing the knowledge. The 2^{nd} and 4^{th} steps play the essential roles in maintaining the reliability in constructed databases in our proposed system. Boosted by the power of deep reinforcement learning in organizing these actions, the ability of deep Bi-directional long short-term memory (LSTM) in text mining, and massive amount of knowledge encoded in Unified Medical Language System (UMLS),[8] we are able to present our human-like system that can imitate the behaviors of a real scientist and construct the database of reliable and cutting-edge biomedical publications efficiently and endlessly. Therefore, we name our system the Everlasting Iatric Reader (Eir)[b]. We further apply our system to construct a genetic association database, where we can verify the performance of Eir with a manually crafted database of 167k gene-trait associations from high quality articles.[9]

The contributions of this paper are three-fold:

- We propose to improve the reliability of text-mining by building a system that can directly simulate the behavior of a researcher, and we develop corresponding methods, such as Bi-directional LSTM for text mining and Deep Q-Network for organizing behaviors.
- We demonstrate the effectiveness of our system with an example in constructing a genetic association database.
- We release our implementation as a generic framework for researchers in the community to conveniently construct other databases.

[a]the database maintained by the National Center for Biotechnology Information (NCBI)

[b]We name our system Everlasting Iatric Reader because it can endlessly construct the knowledge in the medical area, where the high reliability is an issue, and also because the acronym (Eir) shares the name of the goddess of medical knowledge in Norse mythology, which is related to the final goal of this and following-up projects.

The remainder of this paper is organized as follows. In Section 2, we will introduce the related works in biomedical text mining. In Section 3, we will systematically introduce our system, mainly with deep reinforcement learning module that organizes the actions, text mining module that extracts the information, and implementation specifications. In Section 4, we will compare the performance to validate the strategy of Eir. Finally, in Section 5, we will draw conclusions and discuss about the future work.

2. Related Work

Text mining from biomedical literature has been studied extensively for a long time with a variety of different applications, such as patient analysis from electronic health records,[10–12] gene annotations from protein networks,[13] and drug repositioning from literature.[14] One can refer to comprehensive reviews[4,15,16] and the references therein for more detailed discussions.

The text mining usually leads to automatic construction of knowledge bases. In recent years, Mallory *et al.*[17] curated a database of gene-gene interactions. They applied the information extraction engine DeepDive[18] to around 100k full text PLOS articles for extracting direct and indirect gene-gene interactions. Poon *et al.*[19] introduced the Literome project, where they extracted directed genic interactions and genotype-phenotype associations from PubMed articles. Lossio-Ventura *et al.*[20] introduced a pipeline to build an obesity and cancer knowledge base. Very recently, Lossio-Ventura *et al.* also noticed the reliability issue of knowledge base, so they further proposed to incorporate cross-sourcing process to improve the reliability of the their previously developed knowledge base.[21]

On the other hand, the boom of deep learning techniques has allowed many more advanced methods developed for biomedical applications.[22–24] As a result, LSTM and its variants,[25,26] and word embedding techniques[27,28] have been studied extensively for a variety of applications.

In comparison, a difference between most of previous work and our work is that we aim to improve the reliability of the extracted knowledge by examining the source unstructured data (*i.e.* the PubMed literature in our case). To put in simpler words, while most previous work are extending human's intelligence of comprehending the articles, our system aims to extend human's intelligence of the entire research process that starts with querying the web and selecting the interesting article. To the best of our knowledge, this paper is the first one that simulates the entire research process in biomedical information extraction to improve the reliability of the constructed knowledge base. However, many similar concepts[29–31] have been proposed previously. Most relevantly, Kanani *et al*[32] utilized reinforcement learning to reduce computational bottlenecks, minimizing the number of queries, document downloads and extraction action, a similar strategy has been proposed independently for biomedical text mining with the concept "focused machine reading",[33] which is inspired by Narasimhan *et al*,[34] who built an information extraction system that can query the web for extra information with reinforcement learning.

3. Method

In this section, we officially introduce the our system. We will start with the main framework, and continue to introduce the deep reinforcement learning module that organize differ-

ent actions of the system, which is followed by the discussions of proprecessing module and biomedical text mining module. After a systematic introduction of the detailed algorithms, this section is concluded with implementation specifications.

3.1. *Model Framework*

Eir's research process is a markov decision process (MDP), where the model learns to query the search engine for scientific articles to read for the knowledge. We represent the MDP as a tuple $< S, A, T, R >$, where $S = s$ is the space of all possible states, $A = a$ is the set of all actions, $R(s, a)$ is the reward function, and $T(s'|s, a)$ is the transition function.

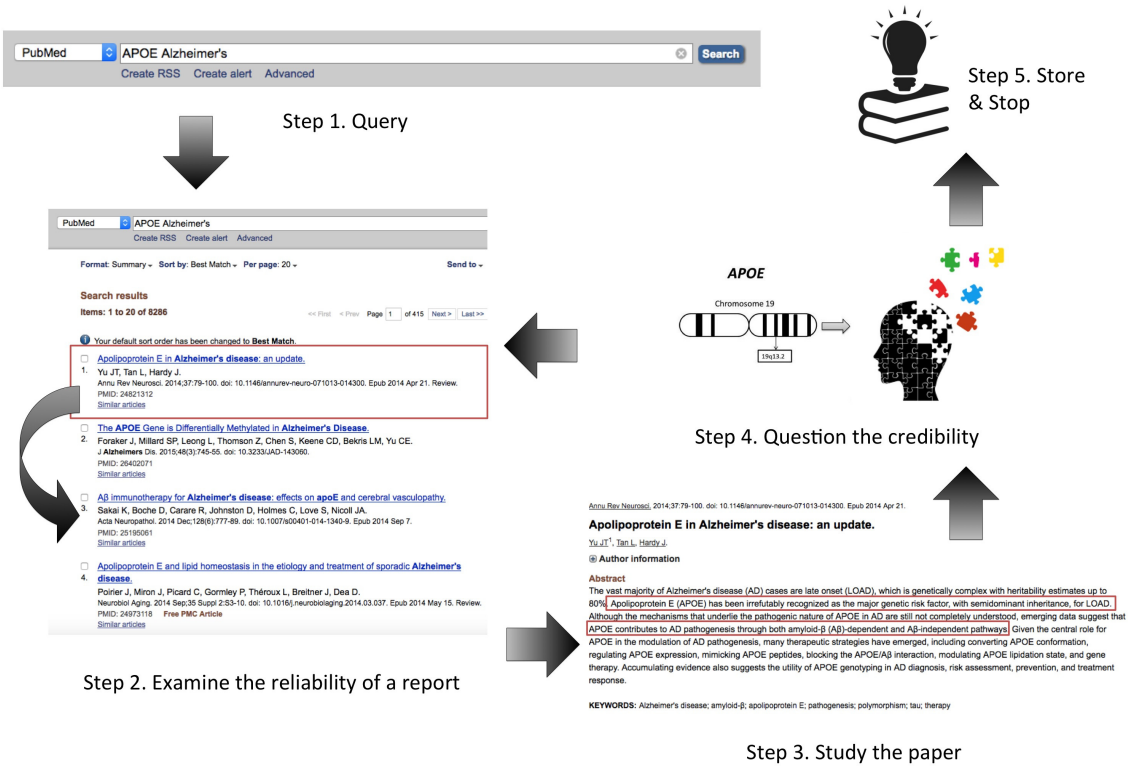


Fig. 1: Overview of Eir's possible behaviors

We present the details of these components as following:

- **Actions:** Action (we use a to denote action throughout this paper) is a set of Eir's behaviors to simulate a real researcher, including
  1. Query the search engine.
  2. Evaluate whether the article is reliable.
  3. Read the article for detailed information.
  4. Exam credibility of the information and querying again.
  5. Stop.

As shown in Figure 1, for every interesting query, Eir starts with the 1st action and then enters the loop from the 2nd action to the 4th action until Eir is satisfied with the finding of current research interest and ceases with the 5th action. Then Eir repeats the entire process with another query.

- **States:** The state s in the MDP describes the research status of Eir, possible candidate states include the ones that are precedent or after each aforementioned action. There are only a countable number of actions, but we use continuous real-valued vector to represent each state so that we could have a better modeling power to distinguish Eir's research status after each action. The state is constructed with a variety of information, including the embedding vector that the Bidirectional LSTM yields, the confidence of biomedical text mining module, the confidence of selecting an article to read, etc.
- **Rewards:** The reward function is chosen to maximize the intermediate paper selection accuracy and final extraction accuracy together while minimizing the number of queries. The accuracy component is calculated using the difference between the accuracy of the current and the previous set of entity values.
- **Transitions:** Transition $T(s'|s,a)$ is modeled as a function of how the next state s' is updated given the current state s and action a taken.

3.2. *Deep Reinforcement Learning for Organizing Actions*

As we have introduced previously, we utilized deep reinforcement learning to arrange the sequence of actions a to perform, given a state function denoted as $Q(s,a)$. To update $Q(s,a)$, we used the popular Q-learning,[35] which iteratively updates $Q(s,a)$ as following:

$$Q_{i+1}(s,a) = \mathbf{E}[r + \gamma \max_{a'} Q_i(s',a')|s,a]$$

where $r = R(s,a)$ is the reward and γ is a discounting factor.

Because of the continuous nature of our state space S, we use a deep Q-network (DQN)[36] as a function approximator $Q(s,a) = Q(s,a;\theta)$. The Q-function of DQN is approximated by a neural network, whose parameters (*i.e.* θ) are updated through stochastic gradient descent. We followed the detailed parameter learning strategies introduced previously.[34]

3.3. *Preprocessing and Name Entity Recognition with UMLS*

Before we feed in the texts into the text mining module, we notice that the literature is filled with alternative, idiosyncratic and arbitrary names and symbols. The text mining module will only exhibit its full power when the texts are processed into a uniform representation. Therefore, we utilize the rich information collected by the unified medical language system (UMLS).[8] UMLS defines a unique concept for all the terms that are interchangeable. For example, "Alzheimer's disease", "Alzheimer's", and "alzheimer" will be mapped into the same concept. UMLS contains over one million biomedical concepts that are split into 133 broad categories (such as "Organisms", "Anatomical structures", "Biologic function"). With the help of MetaMap,[37] we are able to translate the unstructured texts into a sequence of concepts, together with the category information, an associated confidence score, and two binary values to indicate whether the concept is in gene ontology, and in disease ontology respectively.

3.4. *Bidirectional LSTM for Relation Classification*

As Eir queries PubMed with a gene-trait pair, the text mining model only needs to classify whether the returned texts from PubMed can be seen as evidences to support that there is association between the queried gene-trait pair. Therefore, the text mining module can be conveniently regarded as a classification module. We use a Bidirectional LSTM[38] for classifying whether the text describes as association relationship between the gene and the trait the system queried. We choose this Bidirectional LSTM architecture mainly because we notice that it is empirically the best performing method among other neural architecture for our specific task. We first treat the sequence of concepts as words in text and created a 512-dimension vector of continuous values to represent each concept. Further, we feed in this concept-embedding, together with an one-hot representation of the category information, and the two binary values into the Bidirectional LSTM, which is trained through Adam.

3.5. *Algorithm*

Algorithm 1 describes the overall algorithm of the MDP process of Eir, where g and t stands for gene and trait respectively, a stands for action, s stands for state, and r stands for reward. "Agent" refers to DQN, which organizes the sequence of actions given states and reward. Details including the methodology of updating (s, r) has been discussed in previous sections.

3.6. *Implementation Specification*

The Deep Reinforcement Learning component of Eir is implemented as an extension of Narasimhan *et al*,[34] we also use a DQN consisting of two linear layers (20 hidden units each) followed by rectified linear units (ReLU), along with two separate output layers.

The web query component is built with a web crawling engine Scrapy[c] communicating with NCBI PubMed search engine. At this moment, we only query for the abstracts of the articles. We only work with abstracts for three reasons: 1) this allow us to conveniently access and scan a large amount literature, 2) we notice that a majority of articles disclose the most important findings in the abstract with a straightforward style of writing, 3) previous work notice that mining from full texts may lead to more false positives.[39]

The preprocessing module is built as a python script that runs MetaMap, which is a binary software that allows users to conveiently annotate words and phrases of texts with manually defined concepts in UMLS.

The sentences are truncated with max length of 300 concepts. We only consider the 30,000 most frequent concepts together with the specific defined 'SOS' (start of sentence), 'EOS' (end of sentence), 'UNK' (unknown) and 'PAD' (padding the sentences shorter than 300) concepts. We use a 2-layer Bidirectional LSTM with hidden dimension set to 1000, and feed 512 dimension concept embedding, one dimension gene ontology, one dimension disease ontology, and 136 dimension semantic type as the input of LSTM. The LSTM is trained jointly with the embedding matrix using Adam with step size set to 0.00004 and batch size set to 64.

[c]https://scrapy.org/

Then We train the Eir models for 10000 steps every epoch using the Maxent classifier as the base extractor, and evaluate on the entire test set every epoch. The final accuracy reported are averaged over three independent runs; each runs score is averaged over 5 epochs after 45 epochs of training. The penalty per step is set to -0.001. We used a replay memory of size 500k, and a discount γ of 0.8. We set the learning rate to 2.5E5. The ϵ in ϵ-greedy exploration is annealed from 1 to 0.1 over 500k transitions. The target-Q network is updated every 5k steps. The whole framework was trained to optimize the reward function.

We release our implementation[d] for the community to use our system or build more advanced text mining module into our system for better performance.

Algorithm 1 MDP framework of Eir

```
for epoch = 1, M do
    for g, t in query_list do
        Query the search engine with g and t.
        Update and send state (s, r) to agent
        Get action a from agent
        while a is not "stop" do
            if a is "select" then
                Update (s, r) with selection
            else if a is "reject" then
                Update (s, r) with rejection
            else
                Translate texts into sequence of concept embeddings.
                Relation classification with Bidirectional LSTM
                Update (s, r) with classified relation
            end if
            Send state (s, r) to agent
            Get action a from agent
        end while
    end for
end for
```

4. Experiments

In this section, we will verify the performance of Eir by showing that, with the same text mining module, the Eir system can help improve the performance of extracted associations. We will first discuss how we construct the experimental data sets then discuss the results.

4.1. *Data*

Within the scope of this paper, Eir focus on constructing the knowledge base for gene-trait association relationship of human. To enable Eir to learn the associations, we utilized the high

[d]https://github.com/lebronlambert/Eir

quality data set of 167k association relationship that is manually crafted for over ten years.[9]

In addition to the gold-standard information of gene-trait association relationship, another contribution of this data set is the collection of high quality publications that report these associations. Every entry in the database is grounded by the authentic source of scientific paper that originally publishes the relationship. These detailed information grants us the possibility of directly training Eir to discriminate the reliable papers out of the less favourable papers that were not selected by GAD curators for some reasons.

Despite that Eir is designed for extracting latest information online, in order to test the effectiveness of Eir, we need to run the core functions on a local collections of articles with manually labelled true associations. Therefore, we query the PubMed with 54,041 queries of gene-trait pairs through our API and download 913,939 results with 305,651 distinct medical articles. After removing some invalid records (e.g. articles with invalid PMID), there are roughly 133,548 records (44,592 distinct articles) appear in the GAD database, which will serve as the reliable articles. As the construction of GAD ceased in 2014, we regard the articles that are published before 2014 but not in the GAD database as less favorable articles. To balance the data set for performance evaluation, we sampled 140,361 less favorable records before 2014 for comparison. Note that, these less favorable articles are not collected randomly, but are returned from PubMed search engine when we query with a pair of gene and trait. Besides, we delete the articles whose titles and abstracts do not contain the queried gene and trait explicitly to remove obviously irrelevant articles. Then, we random split the whole data set to sample 80% records as training data, and the rest as testing data. The training set consists of 55k records, the testing set consists of 219k records.

4.2. *Evaluation*

In order to show the effectiveness of the Eir system, we compare the system's precision, recall, and F1 score with a conventional biomedical text mining strategy that scans all the documents nondiscriminatorily. As Eir uses the Bidirectional LSTM for text mining module, we use the same model as baseline method for fair comparison.

4.3. *Results*

4.3.1. *Improved Reliability*

We first train our baseline Bidirectional LSTM and the results are shown in the Table 1 (first row). The Bidirectional LSTM yields a precision of 91.25%, a recall of 96.55%, and an overall F1 of 93.80%. These numbers indicate that the Bidirectional LSTM is capable to capture the feature of authentic articles.

Table 1: Results of Reliability Comparison

	precision	recall	F1
Bidirectional LSTM	91.25%	96.55%	93.80%
Eir	91.4%	97.0%	94.1%

Further we add the Deep Reinforcement Learning component to train the overall Eir system. The results of Eir are shown as Table 1 (second row). We can see that the precision score is 91.4%, the recall score is 97.0%, the F1

score is 94.1%. Compared to the baseline model, our Eir framework is better at extracting the features of valuable articles and utilizing the information and can retrieve the authentic articles more efficiently by employing the Deep Reinforcement Learning module.

4.3.2. *Robustness in Real-world Situations*

Table 2: Results of Eir in real-world situations

	Full Data			20% Authentic Articles			10% Authentic Articles		
	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1
Bi-LSTM	91.25%	96.55%	93.80%	87.7%	95.7%	91.5%	86.9%	92.2%	89.4%
Eir	91.4%	97.0%	94.1%	87.9%	96.9%	92.2%	87.8%	96.9%	92.1%
Increment	0.16%	0.47%	0.32%	0.23%	1.25%	0.77%	1.04%	5.10%	3.02%

To better simulate the real-world situation that the researchers are in, we remove different percentage of authentic articles both in the training data set and in the testing data set, for the researchers get ample amount of less favorable articles. We randomly remove a certain percentage of authentic articles to do the ablation experiments. As the percentage of authentic articles decreases, the difficulty of our task increases. The results are shown in Table 2. We can see the Eir system is more robust than the baseline model under these situations. Eir reports higher precision, recall, and F1 score in all of these settings. More interestingly, we calculate the increments Eir achieves over baseline model. We notice that, as the difficulty increases, the increment also increases. Therefore, we believe Eir will be more helpful in the real-world situation when a large amount of articles are less favorable articles.

4.3.3. *Number of Articles Read*

Finally, we examine Eir's performance in the numbers of articles it needs to read to make a decision. Since Eir stops once it believes it has sufficient amount of information, we anticipate Eir will inspect less amount of articles than baseline models. To conduct this experiment, we exclude the gene-trait query pair with only one authentic articles. In the remaining data set, there is on average 2.54 articles for every query, and Eir reads only on average 2.46 articles. We further repeat this experiment with a data set that excludes all the articles with less than 4 articles per query, resulting in a data set with on average 6.23 articles for every query. Eir reads on average 6.10 articles.

5. Conclusions and Future Work

In this work, we introduced a system, namely Everlasting Iatric Reader (Eir), for biomedical text mining. A distinct difference between our system and previous biomedical text mining works is that our system is aimed to directly simulate the behaviors of scientists, including searching for scientific literature, examining the reliability of the manuscript, studying the paper for details, and continuing to search with suspicion of the learned knowledge.

In contrast to traditional biomedical text mining tools, the distinguishable advantage Eir has is the ability to discriminate reliable articles out of questionable articles and to shield the problems introduced by humans. This ability is particularly important in biomedical areas because in clinics, a falsely constructed knowledge may lead to fatal errors, while a missing piece of true knowledge will at most delay the cure of certain disease. Also, it is necessary to select trustworthy papers to read for information because it is known that there is a non-negligible number of publications with the troubles of reproducibility.

There are also limitations of the current Eir. For example, the action of Eir for evaluating the literature quality is trained supervisedly. The performance of our Eir can be greatly improved with a more cleaned data source, as now the false positives are introduced by some manuaaly crafted data that are labeled not correctly. Therefore, we will need a manually crafted data set first before we use Eir in some application. In this paper, we choose to construct the genetic association database because of the availability of GAD.[9] However, there are still a large number of manually curated databases with information about which paper these information comes from, such as GWAS Catalog[40] for SNP-phenotype association or UniProt[41] for protein function annotation.

Looking into the future, a direct extension of our work is to broaden Eir vision to ask investigate into more biomedical topics in addition to gene-trait association relationships. Our immediate next-step plan is to train Eir for SNP-phenotype association with GWAS Catalog, then we can integrate these databases into GenAMap,[42] a visual machine learning tool for GWAS[e], for validation purpose of GWAS results. On the method development side, we hope to upgrade the biomedical text mining module with state-of-the-art methods to improve the information extraction performance, so that Eir could serve the community better. As a long-term plan, we hope Eir could help the community to build the omini-biomedical knowledge base, therefore, we released the source code of Eir for others in the community to use.

6. Acknowledgement

This work is funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. This work is also supported by the National Institutes of Health grants R01-GM093156 and P30-DA035778.

References

1. P. O. Larsen and M. Von Ins, The rate of growth in scientific publication and the decline in coverage provided by science citation index, *Scientometrics* **84**, 575 (2010).
2. K. Raja, M. Patrick, Y. Gao, D. Madu, Y. Yang and L. C. Tsoi, A review of recent advancement in integrating omics data with literature mining towards biomedical discoveries, *International journal of genomics* **2017** (2017).
3. A. M. Cohen and W. R. Hersh, A survey of current work in biomedical text mining, *Briefings in bioinformatics* **6**, 57 (2005).

[e]http://www.genamap.org/

4. K. B. Cohen and D. Demner-Fushman, *Biomedical natural language processing* (John Benjamins Publishing Company, 2014).
5. G. Poste, Bring on the biomarkers, *Nature* **469**, 156 (2011).
6. K. Shoenbill, N. Fost, U. Tachinardi and E. A. Mendonca, Genetic data and electronic health records: a discussion of ethical, logistical and technological considerations, *Journal of the American Medical Informatics Association* **21**, 171 (2013).
7. H. Kilicoglu, Biomedical text mining for research rigor and integrity: tasks, challenges, directions, *Briefings in bioinformatics* (2017).
8. O. Bodenreider, The unified medical language system (umls): integrating biomedical terminology, *Nucleic acids research* **32**, D267 (2004).
9. K. G. Becker, K. C. Barnes, T. J. Bright and S. A. Wang, The genetic association database, *Nature genetics* **36**, 431 (2004).
10. B. M. Hollister, N. A. Restrepo, E. Farber-Eger, D. C. Crawford, M. C. Aldrich and A. Non, Development and performance of text-mining algorithms to extract socioeconomic status from de-identified electronic health records, in *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2017*, 2017.
11. B. K. Beaulieu-Jones, P. Orzechowski and J. H. Moore, Mapping patient trajectories using longitudinal extraction and deep learning in the mimic-iii critical care database, *bioRxiv* , p. 177428 (2017).
12. B. S. Glicksberg, R. Miotto, K. W. Johnson, K. Shameer, L. Li, R. Chen and J. T. Dudley, Automated disease cohort selection using word embeddings from electronic health records, in *Pac Symp Biocomput*, 2018.
13. S. Wang, J. Ma, M. K. Yu, F. Zheng, E. W. Huang, J. Han, J. Peng and T. Ideker, Annotating gene sets by mining large literature collections with protein networks, in *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 2018.
14. R. Frijters, M. Van Vugt, R. Smeets, R. Van Schaik, J. De Vlieg and W. Alkema, Literature mining for the discovery of hidden connections between drugs, genes and diseases, *PLoS computational biology* **6**, p. e1000943 (2010).
15. G. H. Gonzalez, T. Tahsin, B. C. Goodale, A. C. Greene and C. S. Greene, Recent advances and emerging applications in text and data mining for biomedical discovery, *Briefings in bioinformatics* **17**, 33 (2015).
16. F. Liu, J. Chen, A. Jagannatha and H. Yu, Learning for biomedical information extraction: Methodological review of recent advances, *arXiv preprint arXiv:1606.07993* (2016).
17. E. K. Mallory, C. Zhang, C. Ré and R. B. Altman, Large-scale extraction of gene interactions from full-text literature using deepdive, *Bioinformatics* **32**, 106 (2015).
18. F. Niu, C. Zhang, C. Ré and J. W. Shavlik, Deepdive: Web-scale knowledge-base construction using statistical learning and inference., *VLDS* **12**, 25 (2012).
19. H. Poon, C. Quirk, C. DeZiel and D. Heckerman, Literome: Pubmed-scale genomic knowledge base in the cloud, *Bioinformatics* **30**, 2840 (2014).
20. J. A. Lossio-Ventura, W. Hogan, F. Modave, Y. Guo, Z. He, A. Hicks and J. Bian, Oc-2-kb: A software pipeline to build an evidence-based obesity and cancer knowledge base, in *Bioinformatics and Biomedicine (BIBM), 2017 IEEE International Conference on*, 2017.
21. J. A. Lossio-Ventura, W. Hogan, F. Modave, Y. Guo, Z. He, X. Yang, H. Zhang and J. Bian, Oc-2-kb: integrating crowdsourcing into an obesity and cancer knowledge base curation system, *BMC Medical Informatics and Decision Making* **18**, p. 55 (2018).
22. S. Min, B. Lee and S. Yoon, Deep learning in bioinformatics, *Briefings in bioinformatics* **18**, 851 (2017).
23. H. Wang and B. Raj, On the origin of deep learning, *arXiv preprint arXiv:1702.07800* (2017).
24. T. Yue and H. Wang, Deep learning for genomics: A concise overview, *arXiv preprint*

arXiv:1802.00810 (2018).
25. F. Li, M. Zhang, G. Fu and D. Ji, A neural joint model for entity and relation extraction from biomedical text, *BMC bioinformatics* **18**, p. 198 (2017).
26. W. Zheng, H. Lin, L. Luo, Z. Zhao, Z. Li, Y. Zhang, Z. Yang and J. Wang, An attention-based effective neural model for drug-drug interactions extraction, *BMC bioinformatics* **18**, p. 445 (2017).
27. Z. Jiang, L. Li, D. Huang and L. Jin, Training word embeddings for deep learning in biomedical text mining tasks, in *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on*, 2015.
28. M. Habibi, L. Weber, M. Neves, D. L. Wiegandt and U. Leser, Deep learning with word embeddings improves biomedical named entity recognition, *Bioinformatics* **33**, i37 (2017).
29. A. Termehchy, A. Vakilian, Y. Chodpathumwan and M. Winslett, Which concepts are worth extracting?, in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, 2014.
30. R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta and D. Lin, Knowledge base completion via search-based question answering, in *Proceedings of the 23rd international conference on World wide web*, 2014.
31. M. Samadi, P. P. Talukdar, M. M. Veloso and T. M. Mitchell, Askworld: Budget-sensitive query evaluation for knowledge-on-demand., in *IJCAI*, 2015.
32. P. H. Kanani and A. K. McCallum, Selecting actions for resource-bounded information extraction using reinforcement learning, in *Proceedings of the fifth ACM international conference on Web search and data mining*, 2012.
33. E. Noriega-Atala, M. A. Valenzuela-Escárcega, C. T. Morrison and M. Surdeanu, Learning what to read: Focused machine reading, *arXiv preprint arXiv:1709.00149* (2017).
34. K. Narasimhan, A. Yala and R. Barzilay, Improving information extraction by acquiring external evidence with reinforcement learning, *arXiv preprint arXiv:1603.07954* (2016).
35. C. J. Watkins and P. Dayan, Q-learning, *Machine learning* **8**, 279 (1992).
36. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, Human-level control through deep reinforcement learning, *Nature* **518**, 529 (2015).
37. A. R. Aronson, Effective mapping of biomedical text to the umls metathesaurus: the metamap program., in *Proceedings of the AMIA Symposium*, 2001.
38. A. Graves and J. Schmidhuber, Framewise phoneme classification with bidirectional lstm and other neural network architectures, *Neural Networks* **18**, 602 (2005).
39. W. W. Fleuren and W. Alkema, Application of text mining in the biomedical domain, *Methods* **74**, 97 (2015).
40. D. Welter, J. MacArthur, J. Morales, T. Burdett, P. Hall, H. Junkins, A. Klemm, P. Flicek, T. Manolio, L. Hindorff *et al.*, The nhgri gwas catalog, a curated resource of snp-trait associations, *Nucleic acids research* **42**, D1001 (2013).
41. U. Consortium, Uniprot: the universal protein knowledgebase, *Nucleic acids research* **45**, D158 (2016).
42. H. Wang, B. J. Lengerich, M. K. Lee and E. P. Xing, Genamap on web: Visual machine learning for next-generation genome wide association studies, *in preparation* (2018).

Estimating classification accuracy in positive-unlabeled learning: characterization and correction strategies

Rashika Ramola, Shantanu Jain, Predrag Radivojac*
Northeastern University, Boston, Massachusetts, U.S.A.

Accurately estimating performance accuracy of machine learning classifiers is of fundamental importance in biomedical research with potentially societal consequences upon the deployment of best-performing tools in everyday life. Although classification has been extensively studied over the past decades, there remain understudied problems when the training data violate the main statistical assumptions relied upon for accurate learning and model characterization. This particularly holds true in the open world setting where observations of a phenomenon generally guarantee its presence but the absence of such evidence cannot be interpreted as the evidence of its absence. Learning from such data is often referred to as positive-unlabeled learning, a form of semi-supervised learning where all labeled data belong to one (say, positive) class. To improve the best practices in the field, we here study the quality of estimated performance in positive-unlabeled learning in the biomedical domain. We provide evidence that such estimates can be wildly inaccurate, depending on the fraction of positive examples in the unlabeled data and the fraction of negative examples mislabeled as positives in the labeled data. We then present correction methods for four such measures and demonstrate that the knowledge or accurate estimates of class priors in the unlabeled data and noise in the labeled data are sufficient for the recovery of true classification performance. We provide theoretical support as well as empirical evidence for the efficacy of the new performance estimation methods.

Keywords: Positive-unlabeled learning, AlphaMax, Matthews correlation, accuracy estimation.

1. Introduction

Machine learning-based prediction has become the cornerstone of modern computational biology and biomedical data science. Numerous approaches have been developed and applied in these fields, including those related to the function of biological macromolecules,[1,2] the effect of genomic variation,[3] precision medicine,[4,5] or computer-aided clinical decision making.[6] A significant part of this research considers binary classification where the learning algorithms have been extensively studied and characterized, both theoretically and empirically.[7] The objective in binary classification is to train (learn) a model (function) that can distinguish one type of objects from another; e.g., predicting the effect of single nucleotide variants as pathogenic or benign.[3] However, these algorithms have a broader value because multi-class, multi-label and even structured-output learning are often framed as extensions of binary classification, sometimes in a straightforward manner.[8]

In addition to learning, binary classification has also been extensively studied with respect to the performance evaluation of predictive models.[7] Typically, the prediction algorithm outputs a real-valued score for a given input example, after which a thresholding function is applied to map the prediction score into one of the elements of the output space (e.g., pathogenic vs. benign). In some cases, one first chooses the decision threshold and then computes the performance measures for the model on the binarized predictions. In others, calcu-

*The first two authors should be regarded as Joint First Authors.

lating the performance measures entails some form of aggregating over all decision thresholds. The first category of evaluation metrics includes classification accuracy, or the probability that a randomly selected, previously unseen, example from the population will be correctly classified. Other, more specialized measures, include the true positive rate (sensitivity, recall), true negative rate (specificity, 1 − false positive rate) or precision (positive predictive value, 1 − false discovery rate).[7] These measures may also be combined to compute derived quantities such as the balanced sample accuracy, F-measure[7] or Matthews correlation coefficient.[9] The second group of metrics include two-dimensional plots such as the Receiver Operating Characteristic (ROC) curve and the precision-recall curve that visualize the trade-offs between various quantities as a function of the decision threshold. These curves can be further summarized into a single quantity by computing the area under the curve. Alternatively, metrics such as F-measure can be computed for each decision threshold to report the maximum value over all thresholds; e.g., $F_{\max}$.[10] This allows each algorithm to select its own decision threshold and also comparisons between algorithms that binarize their outputs with those that do not. It is worth mentioning that cost-sensitive learning and evaluation,[11,12] as well as information-theoretic approaches[13,14] can also be considered in certain classification scenarios; however, these evaluation strategies are beyond the scope of this work.

Although binary classification has been extensively studied and is well understood,[7] there remain problems related to the open world setting that require attention. Open world refers to the framework in knowledge representation and artificial intelligence in which the observation of a phenomenon generally establishes its presence; however, the lack of the observation cannot be interpreted as the evidence of absence of the phenomenon. One such example is protein function assignment,[15] where an experimental assay can definitively establish, say, that a particular protein is an enzyme. High-throughput experiments can similarly establish the presence of the phenomenon, albeit with some error as in generating protein-protein interaction networks using yeast two-hybrid systems.[16] However, no protein has ever been experimentally assayed for all functions and, additionally, an unsuccessful experiment does not necessarily establish the lack of particular activity. This is because an absence of required molecular partners, an inadequate set of experimental conditions (e.g., pH, temperature[17]), or a human error can combine to result in a failed experiment.[b] When presented with such data, one is *de facto* given a set of positive examples (e.g., enzymes) and a set of unlabeled examples (e.g., a sample of all proteins) and the learning setting is referred to as positive-unlabeled learning.[18] Although the unlabeled set contains an unknown fraction of positive examples, the standard practice ignores this fact and considers all unlabeled examples to be negative. One then trains a prediction model (interestingly, this approach is optimal for a wide range of loss functions referred to as composite loss functions[19]) and estimates its performance, after which the predictor is deployed with a particular estimated quality. In other words, machine learning models in the positive-unlabeled setting are trained/evaluated on positive vs. unlabeled data, whereas the ideal predictor, certainly one expected by the downstream user, would be trained/evaluated on positive vs. negative data. Following Elkan and Noto,[20]

[b]Even with exhaustive experimentation and no human error, the "negative" findings are rarely published.

we will refer to the predictors trained on positive vs. negative data as traditional classifiers and models trained on positive vs. unlabeled data as non-traditional classifiers. Similarly, we will refer to the two different types of evaluation as traditional and non-traditional evaluation.

The primary objective of this work is to study non-traditional classifiers and the adverse effects of non-traditional performance evaluation when the intent is to carry out a traditional evaluation. We show that the traditional performance of these classifiers can be recovered with the knowledge or an accurate estimate of class priors (i.e., the fractions of the positive and negative examples in a representative unlabeled set) and the labeling noise (i.e., the fraction of negative examples in the labeled data set that have been mistakenly labeled as positive). We conduct extensive and systematic experiments to evaluate the proposed methods and draw conclusions pertaining to the best practices of performance evaluation in the field.

2. Methods

2.1. *Performance measures: definitions and estimation*

In this section, we give definitions of several widely used performance measures and their standard estimation formulas. To this end, we first describe the probabilistic framework used in the definitions. Consider a binary classification problem of mapping an input $x \in \mathcal{X}$ to its class label $y \in \mathcal{Y} = \{0, 1\}$. Assume that x and y come from an underlying, fixed but unknown joint distribution $h(x, y)$ over $\mathcal{X} \times \mathcal{Y}$.[c] Let $h(x)$ denote its marginal density over x. It follows that $h(x)$ can be expressed as a two-component mixture:

$$h(x) = \pi h_1(x) + (1 - \pi)h_0(x), \tag{1}$$

for all $x \in \mathcal{X}$, where h_1 and h_0 represent the distributions of the positive and negative examples (inputs), respectively, and $\pi \in (0, 1)$ is the proportion of positive examples in h, also referred to as the class prior for the positive class.

Next, we give definitions of the three most fundamental performance measures: (1) true positive rate (γ), the probability that a positive example is correctly classified, (2) false positive rate (η), the probability that a negative example is incorrectly classified as positive, and (3) precision (ρ), the probability that a positive prediction is correct. Mathematically, given a binary classifier $\hat{y} : \mathcal{X} \to \mathcal{Y}$, they are defined as

$$\gamma = \mathbb{E}_{h_1}[\hat{y}(x)], \qquad \eta = \mathbb{E}_{h_0}[\hat{y}(x)], \qquad \rho = \frac{\pi \mathbb{E}_{h_1}[\hat{y}(x)]}{\mathbb{E}_h[\hat{y}(x)]} = \frac{\pi\gamma}{\theta} \tag{2}$$

where $\mathbb{E}_h$ denotes expectations w.r.t. h and $\theta = \mathbb{E}_h[\hat{y}(x)]$ is the probability of a positive prediction. A classifier with a high γ and ρ, but low η is desirable. However, these measures are at odds with each other; i.e., typically, increasing a classifier's γ leads to a smaller ρ and a larger η. A classifier that always predicts either 0 or 1 can optimize them individually at the expense of others. Consequently, they are often used together to gauge a classifier's performance; for example, in an ROC curve analysis. Moreover, other performance measures combine them explicitly or implicitly in their formulation. Though θ itself is not widely used as a measure

[c]For convenience, we use terms density and distribution interchangeably.

	Predicted positive	Predicted negative
Positive	tp	fn
Negative	fp	tn

(a)

$$\hat{\gamma} = \frac{\text{tp}}{\text{tp+fn}} \quad \hat{\pi} = \frac{\text{tp+fn}}{\text{tp+fn+tn+fp}}$$

$$\hat{\eta} = \frac{\text{fp}}{\text{tn+fp}} \quad \hat{\theta} = \frac{\text{tp+fp}}{\text{tp+fn+tn+fp}}$$

(b)

Table 1: (a) Confusion matrix of $\hat{y}(x)$ on a labeled data set. (b) Standard estimation of γ, η, π and θ.

of classifier performance, it also appears in the expression of several important measures (a classifier for which $\theta > \pi$ is sometimes said to "overpredict"). A particularly useful expression of θ in terms of γ, η and π is derived as follows.

$$\theta = \mathbb{E}_h[\hat{y}(x)] = \pi \mathbb{E}_{h_1}[\hat{y}(x)] + (1-\pi)\mathbb{E}_{h_0}[\hat{y}(x)] = \pi\gamma + (1-\pi)\eta \tag{3}$$

In this paper, we focus on four performance measures that are widely used in biomedical research: (1) Accuracy (acc), the probability that a random example is correctly classified (2) Balanced accuracy (bacc), the average accuracy on the positive and negative examples, weighed equally, (3) F-measure (F), the harmonic mean of γ and ρ,[d] and (4) Matthews correlation coefficient (mcc), the correlation between the true and predicted class. Mathematically, they are defined as follows:

$$\text{acc} = \pi\gamma + (1-\pi)(1-\eta) \tag{4}$$

$$\text{bacc} = \frac{1+\gamma-\eta}{2} \tag{5}$$

$$F = \frac{1}{\frac{1}{2}\cdot\frac{1}{\gamma} + \frac{1}{2}\cdot\frac{1}{\rho}} = \frac{2\pi\gamma}{\pi+\theta} \tag{6}$$

$$\text{mcc} = \frac{\mathbb{E}_h[y\cdot\hat{y}(x)] - \mathbb{E}_h[y]\cdot\mathbb{E}_h[\hat{y}(x)]}{\sqrt{\mathbb{V}_h[y]\cdot\mathbb{V}_h[\hat{y}(x)]}} \tag{7}$$

where $\mathbb{V}_h$ in Eq. (7) denotes the variance operator w.r.t. distribution $h(x)$. Notice that, since $y \sim \text{Bernoulli}(\pi)$ under h, $\mathbb{E}_h[y] = \pi$ and $\mathbb{V}_h[y] = \pi(1-\pi)$; similarly, $\mathbb{V}_h[\hat{y}(x)] = \theta(1-\theta)$. Further, using the law of iterated expectations, $\mathbb{E}_h[y\cdot\hat{y}(x)] = \pi\mathbb{E}_{h_1}[\hat{y}(x)] = \pi\gamma$. Thus,

$$\text{mcc} = \sqrt{\frac{\pi}{(1-\pi)}}\frac{\gamma-\theta}{\sqrt{\theta(1-\theta)}} = \sqrt{\frac{\pi(1-\pi)}{\theta(1-\theta)}}\cdot(\gamma-\eta) \tag{8}$$

Using the estimates of γ, η, π and θ from Table 1, we give the standard formulas for acc, bacc, F and mcc estimation, in terms of the classifier's confusion matrix entries. For example, simple algebraic operations on Eq. (8) give

$$\widehat{\text{mcc}} = \frac{\hat{\pi}(1-\hat{\pi})(\hat{\gamma}\cdot(1-\hat{\eta}) - \hat{\eta}\cdot(1-\hat{\gamma}))}{\sqrt{\hat{\theta}\hat{\pi}(1-\hat{\pi})(1-\hat{\theta})}} = \frac{\text{tp}\cdot\text{tn} - \text{fp}\cdot\text{fn}}{\sqrt{(\text{tp}+\text{fp})(\text{tp}+\text{fn})(\text{tn}+\text{fp})(\text{tn}+\text{fn})}}$$

Similarly, the standard estimation formulas for acc, bacc and F can be easily derived as:

$$\widehat{\text{acc}} = \frac{\text{tp}+\text{tn}}{\text{tp}+\text{fn}+\text{tn}+\text{fp}}, \qquad \widehat{\text{bacc}} = \frac{1}{2}\frac{\text{tp}}{\text{tp}+\text{fn}} + \frac{1}{2}\frac{\text{tn}}{\text{tn}+\text{fp}}, \qquad \hat{F} = \frac{2\text{tp}}{2\text{tp}+\text{fn}+\text{fp}}.$$

[d] We only consider the F_1 score in the family of F-measures.

2.2. *Positive-unlabeled setting*

Let **D** represent a set of examples drawn from $h(x)$; at this stage, the class of an x in **D** is unknown. Consider a labeling procedure that selects some examples from **D** for labeling. As is the case in many domains, the procedure tests only for the class of interest, the positive class. The procedure is successful when it deems the example as positive with high confidence. The successfully labeled examples are collected in a labeled set **L**, whereas the rejected examples along with the examples not selected for labeling, in the first place, are collected in an unlabeled set **U**. In spite of being labeled as positive, some examples in **L** might, in fact, be negative, due to the errors in the labeling procedure.

The typical, positive-unlabeled assumption made about the labeler is that the examples from **D** are selected independently of x, given y and further, that the same assumptions apply to the success of labeling.[20,21] The assumptions ensure that the distributions of positives and negatives remain unchanged in **L** and **U** and only the class proportions are affected. Let $f(x, y)$ and $g(x, y)$ denote the underlying joint distribution of **U** and **L**, respectively. Note that y still denotes the true unobserved class and not class assigned by the labeler. For $f(x)$ and $g(x)$ denoting the marginals over x,

$$f(x) = \alpha h_1(x) + (1 - \alpha)h_0(x), \qquad g(x) = \beta h_1(x) + (1 - \beta)h_0(x), \tag{9}$$

for all $x \in \mathcal{X}$, where α and β denote the proportion of positives in the unlabeled and labeled set, respectively. By design, **L** has a higher concentration of positives than **D**; i.e., $\beta \in (\pi, 1]$. Similarly, **U** has a lower concentration of positives than **D**; i.e., $\alpha \in [0, \pi)$. When $\beta = 1$ we say that the labeled data is clean. When $\beta < 1$, the labeled data contains a fraction $(1 - \beta)$ of negatives that are mislabeled. We will refer to the latter scenario as the noisy positive setting and $1 - \beta$ as the noise proportion.

The relationship between h, f and g is further constrained, since **D** is partitioned by **L** and **U**. Precisely,

$$h(x) = cg(x) + (1 - c)f(x) = \big(c\beta + (1 - c)\alpha\big)h_1(x) + \big(1 - c\beta - (1 - c)\alpha\big)h_0(x), \tag{10}$$

for all $x \in \mathcal{X}$, where $c = \frac{|\mathbf{L}|}{|\mathbf{L}|+|\mathbf{U}|}$. Thus,

$$\pi = c\beta + (1 - c)\alpha. \tag{11}$$

To distinguish h from f and g, we refer to h as the true or the target distribution. We are primarily interested in a classifier's performance on the true distribution, which is reflected in our goal to obtain unbiased estimates of the performance measures w.r.t. the true distribution.

2.3. *Performance measure correction*

The absence of negative examples in positive-unlabeled learning is tackled by treating the unlabeled set as a surrogate for negatives. This is referred to as the non-traditional approach.[20] A non-traditional classifier trained on such data learns to discriminate the labeled-as-positive set from the unlabeled set. Surprisingly, an optimal non-traditional classifier has been shown to perform optimally in the traditional sense; i.e., as a discriminator between the positive and negative examples.[21] However, measuring a classifier's performance non-traditionally does not

reflect its performance in the traditional sense. Ref. 22 demonstrated the bias in the non-traditionally estimated γ, η and ρ and its implications towards the ROC and precision-recall analysis. They also provided techniques for bias correction using estimates of the class prior and the noise proportion.[22] We take a similar approach in this work and show that the standard estimators of acc, bacc, F and mcc, when used in a non-traditional framework, are biased. Then we give formulas to correct the bias by estimating the class prior and the noise proportion. To formalize the notion of a non-traditional labeled set, we introduce the pseudo class $\tilde{y}$, which is 1 for every example in $\mathbf{L}$ and 0 for those in $\mathbf{U}$. The non-traditional labeled set $\mathcal{L}^{\text{pu}}$ contains all examples from $\mathbf{L}$ and $\mathbf{U}$ along with their pseudo class labels. The standard approach (see Table 1) for estimating γ, η, π and θ presupposes that the examples in the labeled set are drawn randomly from $h(x, y)$ and more importantly, that tp, fn, tn and fp are counted w.r.t. the true class. However, when working with $\mathcal{L}^{\text{pu}}$, the counts are based on the pseudo class, which affects the quality of the standard estimates.

In particular, $\hat{\gamma}$ and $\hat{\eta}$ give biased estimates of γ and η, respectively. Instead, they give unbiased estimates of $\gamma^{\text{pu}} = \mathbb{E}_g[\hat{y}(x)]$ and $\eta^{\text{pu}} = \mathbb{E}_f[\hat{y}(x)]$; this is because g and f correspond to the distributions of the pseudo positives and the pseudo negatives, respectively. Moreover, $\hat{\pi}$ represents the proportion of the pseudo positives c, instead of π; that is, $\hat{\pi} = c$. However, $\hat{\theta}$ is still an unbiased estimator of θ, since θ only depends on the marginal distribution of x in $\mathcal{L}^{\text{pu}}$, which is the same as $h(x)$ as per Eq. (10). To summarize, we have

$$\hat{\gamma} \xrightarrow{estimates} \gamma^{\text{pu}} \neq \gamma, \qquad \hat{\eta} \xrightarrow{estimates} \eta^{\text{pu}} \neq \eta, \qquad \hat{\pi} = c \neq \pi, \qquad \hat{\theta} \xrightarrow{estimates} \theta.$$

The bias in $\hat{\gamma}$, $\hat{\eta}$ and $\hat{\pi}$ is also reflected in the standard estimates of acc, bacc, F and mcc. They give unbiased estimates of the following quantities instead.

$$\text{acc}^{\text{pu}} = c\gamma^{\text{pu}} + (1-c)(1-\eta^{\text{pu}}) \qquad \text{bacc}^{\text{pu}} = \frac{1 + \gamma^{\text{pu}} - \eta^{\text{pu}}}{2}$$

$$F^{\text{pu}} = \frac{2c\gamma^{\text{pu}}}{c + \theta} \qquad \text{mcc}^{\text{pu}} = \sqrt{\frac{c(1-c)}{\theta(1-\theta)}} \cdot (\gamma^{\text{pu}} - \eta^{\text{pu}})$$

Next, we give the relationship between γ, η, γ^{pu} and η^{pu} which are then used for bias correction.

$$\gamma = \frac{(1-\alpha)\gamma^{\text{pu}} - (1-\beta)\eta^{\text{pu}}}{\beta - \alpha} \qquad \text{obtained by solving}$$

$$\eta = \frac{\beta\eta^{\text{pu}} - \alpha\gamma^{\text{pu}}}{\beta - \alpha} \qquad \gamma^{\text{pu}} = \mathbb{E}_g[\hat{y}(x)] = \beta\gamma + (1-\beta)\eta$$

$$\eta^{\text{pu}} = \mathbb{E}_f[\hat{y}(x)] = \alpha\gamma + (1-\alpha)\eta$$

We derive the bias-corrected estimates of acc, bacc, F and mcc by correcting for γ, η and π:

$$\widehat{\text{acc}}_{cr} = \hat{\pi}_{cr}\hat{\gamma}_{cr} + (1-\hat{\pi}_{cr})(1-\hat{\eta}_{cr}) \qquad (12) \qquad \widehat{\text{bacc}}_{cr} = \frac{1 + \hat{\gamma}_{cr} - \hat{\eta}_{cr}}{2} \qquad (13)$$

$$\hat{F}_{cr} = \frac{2\hat{\pi}_{cr}\hat{\gamma}_{cr}}{\hat{\pi}_{cr} + \hat{\theta}} \qquad (14) \qquad \widehat{\text{mcc}}_{cr} = \sqrt{\frac{\hat{\pi}_{cr}(1-\hat{\pi}_{cr})}{\hat{\theta}\left(1-\hat{\theta}\right)}}(\hat{\gamma}_{cr} - \hat{\eta}_{cr}), \qquad (15)$$

where $\hat{\gamma}_{cr}$, $\hat{\eta}_{cr}$ and $\hat{\pi}_{cr}$ are estimated using estimates of α and β as follows:

$$\hat{\gamma}_{cr} = (\hat{\beta} - \hat{\alpha})^{-1}((1-\hat{\alpha})\hat{\gamma} - (1-\hat{\beta})\hat{\eta}), \qquad \hat{\eta}_{cr} = (\hat{\beta} - \hat{\alpha})^{-1}(\hat{\beta}\hat{\eta} - \hat{\alpha}\hat{\gamma}), \qquad \hat{\pi}_{cr} = c\hat{\beta} + (1-c)\hat{\alpha}.$$

Theorem 2.1 shows that unbiased bacc and mcc estimates can also be directly recovered from bacc^{pu} and mcc^{pu} estimates, requiring only estimation of classifier-independent quantities π, α and β (the class proportions in $\mathbf{D}$, $\mathbf{U}$ and $\mathbf{L}$); i.e., γ and η do not need to be corrected as an intermediate step. Furthermore, the relationship between bacc (mcc) and its positive-unlabeled counterpart is monotonic, which is a desirable property when constructing a classifier by thresholding a score function. It ensures that the threshold obtained with the positive-unlabeled data by optimizing the non-traditional measure also maximizes the traditional measure. The inequalities derived in the theorem demonstrate that the non-traditionally evaluated bacc and mcc underestimate the traditional performance, provided the non-traditional classifier performs better than random.

Theorem 2.1. *The following equations hold true.*

$$\text{bacc} = \frac{2\text{bacc}^{\text{pu}} - 1}{2(\beta - \alpha)} + \frac{1}{2}, \quad \textit{and} \quad \text{mcc} = \frac{1}{\beta - \alpha}\sqrt{\frac{\pi(1-\pi)}{c(1-c)}} \cdot \text{mcc}^{\text{pu}}$$

Moreover,

$$\text{sign}(\text{mcc})(\text{mcc} - \text{mcc}^{\text{pu}}) \geq 0, \quad \textit{and} \quad \text{bacc} - \text{bacc}^{\text{pu}} \geq 0, \ \textit{when } \text{bacc}^{\text{pu}} \geq 1/2.$$

Proof. The proof of the two equalities follow by observing $\gamma^{\text{pu}} - \eta^{\text{pu}} = (\beta - \alpha)(\gamma - \eta)$ and using it in the expressions of bacc^{pu} and mcc^{pu}, thereby obtaining a conversion to bacc and mcc (Eqs. (5) and (8)). Now, $\text{mcc} - \text{mcc}^{\text{pu}} = \text{mcc}^{\text{pu}}\Big(\frac{1}{\beta-\alpha}\sqrt{\frac{\pi(1-\pi)}{c(1-c)}} - 1\Big)$. The mcc inequality follows since $\sqrt{\frac{\pi}{c(\beta-\alpha)}} \cdot \sqrt{\frac{1-\pi}{(1-c)(\beta-\alpha)}} \geq 1$ because $\pi - c(\beta-\alpha) = \alpha \geq 0$ and $1 - \pi - (1-c)(\beta-\alpha) = 1 - \beta \geq 0$. The bacc inequality follows since $\beta - \alpha \geq 0$ and consequently, $2\text{bacc} - 2\text{bacc}^{\text{pu}} = \frac{2\text{bacc}^{\text{pu}}-1}{\beta-\alpha} - (2\text{bacc}^{\text{pu}} - 1) \geq 0$, provided $\text{bacc}^{\text{pu}} \geq 1/2$. □

3. Experiments and Results

3.1. *A case study*

We first demonstrate the problem with non-traditional evaluation in a situation where the positive and negative conditional distributions, h_1 and h_0, are univariate Gaussians with $\mathbb{E}_{h_1}[x] > \mathbb{E}_{h_0}[x]$ and $\mathbb{V}_{h_1}[x] = \mathbb{V}_{h_0}[x]$. Knowing the underlying distributions allows us to make exact computations of performance measures, instead of estimating them from data. As per Section 2, let $h(x) = \pi h_1(x) + (1-\pi)h_0(x)$, $f(x) = \alpha h_1(x) + (1-\alpha)h_0(x)$ and $g(x) = \beta h_1(x) + (1-\beta)h_0(x)$ be the true, labeled and unlabeled data distributions, respectively. Values of α, β and c will be fixed, from which $\pi = c\beta + (1-c)\alpha$ will be computed. We will consider a simple linear classifier $\hat{y}(x) = 1(x \geq \tau)$, where $1(\cdot)$ is the indicator function and $\tau \in \mathbb{R}$ is the decision threshold. This thresholding function predicts a 0 for inputs below τ; otherwise, it predicts a 1.

In the traditional setting, the true positive rate (γ) and false positive rate (η) can be straightforwardly computed as $\gamma = 1 - \text{cdf}_{h_1}(\tau)$ and $\eta = 1 - \text{cdf}_{h_0}(\tau)$, where cdf_f is the cumulative distribution function corresponding to the density f. On the other hand, when evaluated in the non-traditional setting, these quantities can be expressed as $\gamma^{\text{pu}} = 1 - \text{cdf}_g(\tau)$ and

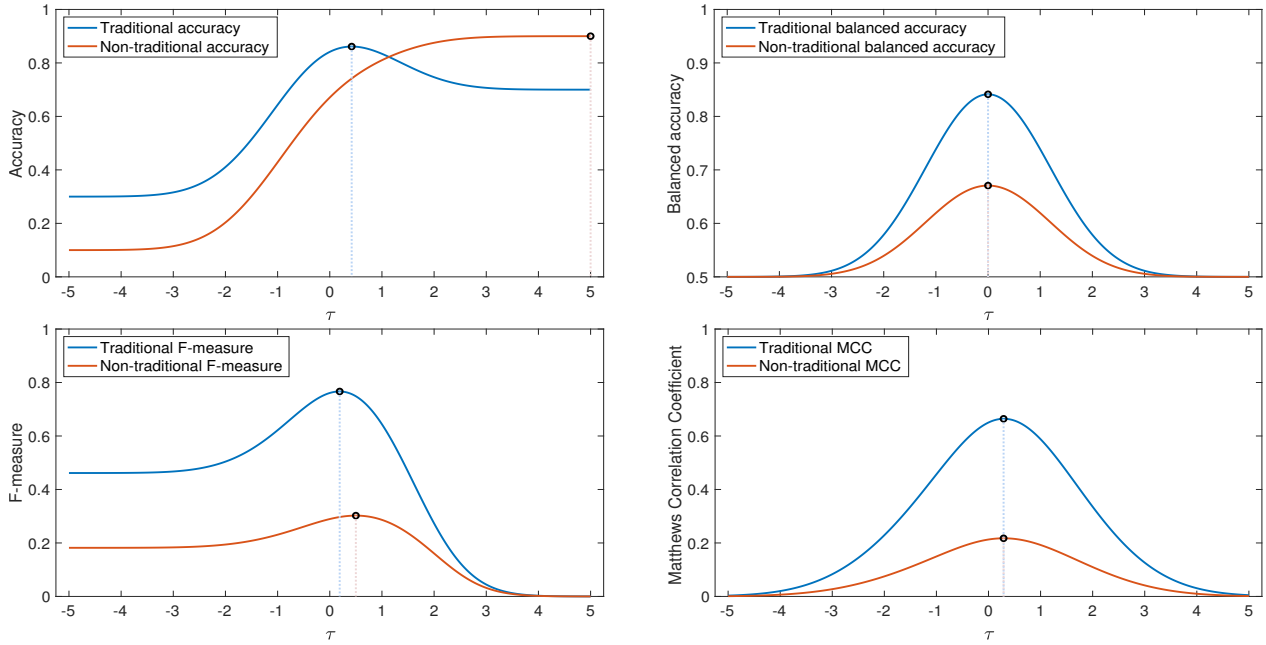


Fig. 1: Traditional vs. non-traditional performance accuracy as a function of decision threshold τ. The circles and vertical lines in all four panels indicate the threshold values and the corresponding best performances in both traditional and non-traditional setting. (Upper left) Classification accuracy: top traditional performance $\text{acc}_{\max} = 0.86$ is reached at the threshold value $\tau = 0.42$, whereas the top non-traditional performance $\text{acc}^{\text{pu}}_{\max} = 0.90$ is reached at $\tau = 5$; (Upper right) Balanced accuracy: top traditional performance $\text{bacc}_{\max} = 0.84$ and non-traditional performance $\text{bacc}^{\text{pu}}_{\max} = 0.67$ are both reached at $\tau = 0$; (Lower left) F-measure: top traditional performance $F_{\max} = 0.77$ is reached at $\tau = 0.19$, whereas the top non-traditional performance $F^{\text{pu}}_{\max} = 0.30$ is reached at $\tau = 0.50$; (Lower right) Matthews Correlation Coefficient: top traditional performance $\text{mcc}_{\max} = 0.66$ and non-traditional performance $\text{mcc}^{\text{pu}}_{\max} = 0.22$ are both reached at $\tau = 0.29$.

$\eta^{\text{pu}} = 1 - \text{cdf}_f(\tau)$. The probability of positive prediction θ is computed using Eq. (3). Of course, $g = h_1$ when $\beta = 1$ and $f = h_0$ when $\alpha = 0$, but this case corresponds to the standard supervised learning problem and is not of interest.

Let us now be concrete and consider that $h_0 = \mathcal{N}(-1, 1)$, $h_1 = \mathcal{N}(1, 1)$, $\alpha = 1/4$, $\beta = 3/4$ and $c = 1/10$; thus, $\pi = 3/10$. In Figure 1, we plot the values of the accuracy, balanced accuracy, F-measure and Matthews correlation coefficient in the traditional and non-traditional setting for each value of $\tau \in (-5, 5)$, where acc, acc^{pu}, bacc, bacc^{pu}, F, F^{pu}, mcc and mcc^{pu} are calculated from γ, η, θ, h, f, g, and c, as shown in Section 2. As a reminder, c represents the proportion of labeled examples in the training set consisting of all labeled and unlabeled examples; however, a data set is not generated here. It is important to point out the large differences between all traditional and non-traditional estimates, which provide evidence that the non-traditional estimates can be far from accurate, as in this example. As proved in Section 2, the maximum values for $\text{bacc}_{\max}$ vs. $\text{bacc}^{\text{pu}}_{\max}$ and $\text{mcc}_{\max}$ vs. $\text{mcc}^{\text{pu}}_{\max}$ are observed at the same score thresholds τ, respectively. This is desirable as one can establish the best decision threshold using positive-unlabeled data and secure the best predictor performance even without the precise knowledge of what that performance is. On the other hand, $\text{acc}_{\max}$ vs. $\text{acc}^{\text{pu}}_{\max}$ as well as $F_{\max}$ vs. $F^{\text{pu}}_{\max}$ do not occur at the same decision thresholds, which presents a

problem for method benchmarking. The F-measure is further interesting as a simple predictor ($\tau = -5$) that gives positive predictions on (almost) all inputs can achieve a high-scoring F, which may be misinterpreted in practice as good performance. Similarly, in terms of accuracy, an inability to "beat" a trivial classifier (the one always predicting the majority class) might be incorrectly interpreted as inability to develop a good classifier.

3.2. *Data sets*

The empirical evaluation was carried out on 14 data sets from the UCI Machine Learning repository. The selected data sets span various biomedical problems, such as recognizing splice-junction boundaries from the DNA sequence,[23] predicting the physical activity of an individual based on their smartphone[24] or sensor[25] data, and predicting hospital re-admissions by using a patient's demographics, medical diagnoses and lab test results.[26] Where necessary, the data sets were converted to binary classification problems by considering one of the classes as positive and the other(s) as negative or by converting regression problems to classification by introducing appropriate thresholds on the target variable. The following data sets were used: Covertype, Activity recognition with healthy older people using a batteryless wearable sensor (two experiments), Epileptic Seizure Recognition, Smartphone-Based Recognition of Human Activities and Postural Transitions, Mushroom, Thyroid Disease, Anuran Calls, Wilt, Abalone, HIV-1 protease cleavage, Splice-junction Gene Sequences, Parkinsons Telemonitoring, and Physicochemical Properties of Protein Tertiary Structure.

3.3. *Experimental protocols*

The experiments were designed to simulate the construction of non-traditional classifiers in the positive-unlabeled setting and assess the quality of performance estimation both in the non-traditional and traditional mode. Labeled and unlabeled data sets, with n_l and n_u examples, respectively, were first created by sampling an appropriate number of positive/negative examples as follows. After fixing the value of β from $\{1, 0.9, 0.8, 0.7\}$, $\beta \cdot n_l$ points were sampled from the positive set and $(1 - \beta) \cdot n_l$ from the negative set to make the labeled data set. This process determined the true value of α as the ratio of the remaining positive points and the remaining negative points from the original data set. Unlabeled data set was then formed by selecting $\alpha \cdot n_u$ points from the remaining positive points and $(1 - \alpha) \cdot n_u$ points from the remaining negative points. The number of unlabeled examples n_u was set to 10,000 in all data sets with sufficient size. Otherwise, it was set to 5000, 2000 or 1000. The size of the labeled data set n_l was picked so as to fix the ratio of labeled vs. unlabeled data to 1:10. That is, if $n_u = 1000$, n_l would be set to 100. This ratio mimics a typical situation in which one is presented with larger unlabeled data compared to the labeled data. A non-linear classification model was trained on each non-traditional data set. Its performance was evaluated in both non-traditional and traditional setting. This experiment was repeated 50 times for different random selections of labeled and unlabeled data sets, each of which was considered for four different values of β.

One-hundred bagged two-layer neural networks, each with 7 hidden neurons, were used as a non-traditional classifier in all experiments. The networks were trained using the RPROP

algorithm[27] with a validation (25% of the training set) stop or at most 5,000 epochs. Out-of-bag performance evaluation was carried out in all experiments. At the end of each run, we calculated four performance measures: the maximum classification accuracy ($\text{acc}_{\max}$), the maximum balanced accuracy ($\text{bacc}_{\max}$), the maximum F-measure ($F_{\max}$) and the maximum MCC ($\text{mcc}_{\max}$), in four different scenarios: (1) the non-traditional (PU) estimates, where the labeled data was considered to be positive and unlabeled data negative; (2) the traditional (true) performance estimates, where the actual class labels instead of the PU labels were used; (3) the recovery setting proposed in Section 2 with actual (α, β) values; and (4) the recovery setting proposed in Section 2 with estimated (α, β) values, referred to as $(\hat{\alpha}, \hat{\beta})$. The non-traditional estimates provide the performance that a practitioner would report by ignoring noise and assuming that the unlabeled set was negative. The traditional performance estimates represent the estimated true performance of these models that a practitioner would not be aware of. The third and fourth scenario represent the traditional estimates after the correction. They were designed to explore the effects of incorrectly estimating (α, β), instead of knowing their true values. The AlphaMax algorithm[21,28] was used to obtain $(\hat{\alpha}, \hat{\beta})$.

3.4. *Results*

We measured the difference between non-traditional and corrected performance against the traditional performance in each run. The traditional performance was considered to be "true"; it could be estimated because the positive-unlabeled setting was simulated on data sets where both positives and negatives were available. The corrected performance was presented twice: first with known (α, β) that were used to construct positive-unlabeled data sets and, second, with (α, β) themselves estimated from the positive-unlabeled data. The experimental results, summarized in a single box plot over all 14 data sets and all 50 runs, are shown in Figure 2. Non-traditionally estimated (without correction) $\text{bacc}_{\max}$, $F_{\max}$ and $\text{mcc}_{\max}$ significantly underestimate the traditional performance, whereasd $\text{acc}_{\max}$ significantly overestimates it. The errors generally deteriorate with the increasing level of noise $(1 - \beta)$.

The corrected estimates attained much smaller error. While using the true values of α and β provided a near perfect recovery of the traditional performance, the estimated values generally resulted in a slightly overestimated traditional performance. We note however that we did not perform any model selection and parameter optimization during class prior and noise level estimation and, therefore, one could expect to observe an improved recovery after these steps. Manual inspection of the likelihood curves outputted by AlphaMax would also be recommended to increase confidence in the recovered performance estimates.

4. Conclusions

Estimating the performance of machine learning models is one of the critical yet understudied research directions in the biomedical sciences. Incorrect evaluation might have severe negative effects upon the deployment of machine learning tools and the perception of their usefulness in the nearby future, including in genetic counseling, precision medicine, clinical decision support, etc.[3–6] This work therefore investigated the quality of performance evaluation in binary classification when training data best fits the positive-unlabeled setting.[18] However,

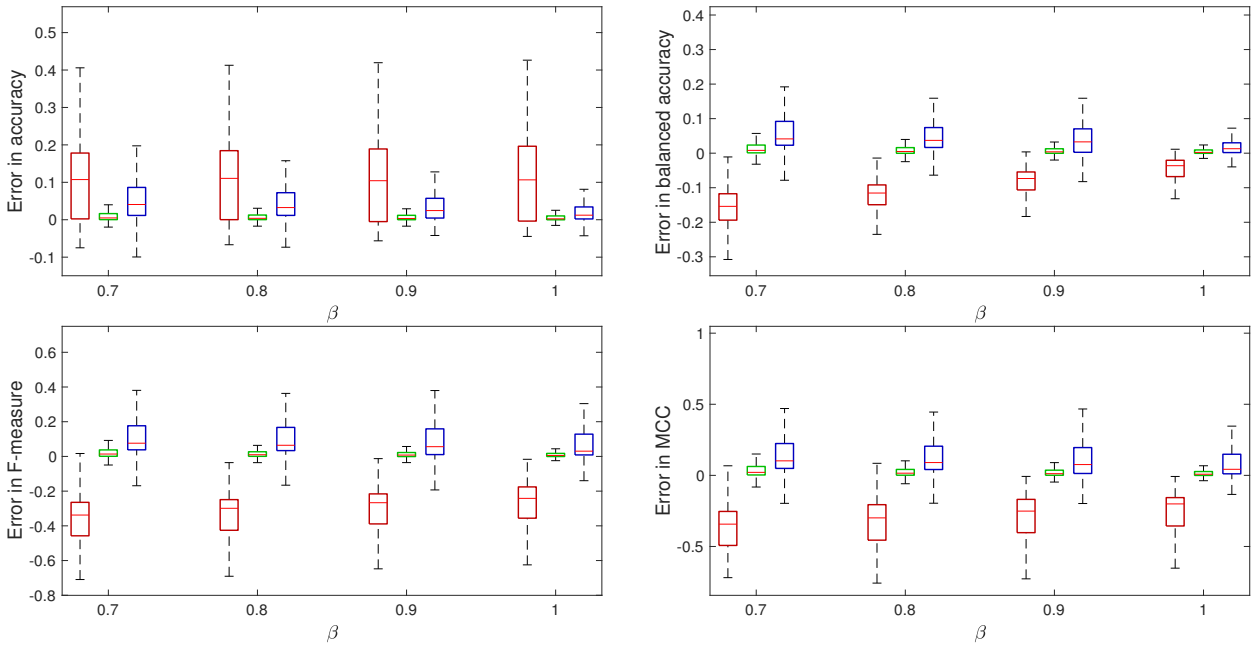


Fig. 2: Error in the non-traditionally evaluated performance measures before and after correction for 14 biomedical data sets. PU represents the estimates on the **P**ositive **U**nlabeled data without bias-correction. CR and CE represent the bias-**C**orrected estimates with the **R**eal and **E**stimated values of α and β. In each run, the optimal decision threshold was selected first, to maximize the performance, and then the resulting performance was compared with the true performance at that same threshold. (Upper left) Classification accuracy: Eq. (12) was used for correction. All estimates were clipped between 0 and 1; (Upper right) Balanced accuracy: Eq. (13) was used for correction. All estimates were clipped between $1/2$ and 1; (Lower left) F-measure: Eq. (14) was used for correction. All estimates were clipped between 0 and 1; (Lower right) Matthews Correlation Coefficient: the formula from Theorem 2.1 was used for a direct correction from the mcc^{pu} estimate. All estimates were clipped between -1 and 1. The x-axis is the true value of β, according to which the box plots were grouped.

the generality of our methods is provided by the equivalence between training from noisy positive vs. unlabeled data and the so-called corrupt binary classification model, where it is assumed that both positive and negative examples are given, but that each data set is corrupted by a (potentially) different amount of label noise.

To characterize performance evaluation problems, we built on the previous work in machine learning[22,29] to evaluate the quality of four estimated measures: accuracy, balanced accuracy, F-measure, and Matthews correlation coefficient. We found that the balanced accuracy and Matthews correlation coefficient are well-behaved, meaning that they provide certain important guarantees to the practitioner even when applied in the positive-unlabeled setting. For example, the optimal decision threshold for maximizing the performance does not change when the evaluation is shifted from the non-traditional to the traditional setting; furthermore, the performance in the traditional setting is always better than non-traditionally estimated. On the other hand, classification accuracy and F-measure provide fewer guarantees and require sophisticated understanding when deployed in practice.

To mitigate the problems associated with any of the above-mentioned performance estimation strategies, we first showed that the true (traditional) classification performance can be recovered with the knowledge of (1) the class priors in the unlabeled data and (2) the propor-

tion of noise in the labeled data. We then used the AlphaMax algorithm[21,28] to estimate both of these quantities in a nonparametric fashion and showed that the performance estimation process is significantly improved. As a practical guideline, we suggest that the deployment of machine learning models should be accompanied with both non-traditional and recovered traditional performance estimates along with the estimated values of α and β.

Acknowledgements

The authors acknowledge the support by the NIH grant R01 MH105524, NSF grant DBI-1458477 and the Precision Health Initiative of Indiana University where the study started.

References

1. R. Rentzsch and C. A. Orengo, *Trends Biotechnology* **27**, 210 (2009).
2. F. Xin and P. Radivojac, *Curr Protein Pept Sci* **12**, 456 (2011).
3. T. A. Peterson *et al.*, *J Mol Biol* **425**, 4047 (2013).
4. G. H. Fernald *et al.*, *Bioinformatics* **27**, 1741 (2011).
5. B. Rost *et al.*, *FEBS Lett* **590**, 2327 (2016).
6. B. Middleton *et al.*, *Yearb Med Inform* **25**, S103 (2016).
7. T. Hastie *et al.*, *The elements of statistical learning* (Springer Verlag, New York, NY, 2001).
8. R. Rifkin and A. Klautau, *J Mach Learn Res* **5**, 101 (2004).
9. B. W. Matthews, *Biochim Biophys Acta* **405**, 442 (1975).
10. P. Radivojac *et al.*, *Nat Methods* **10**, 221 (2013).
11. A. D. Whalen, *Detection of signals in noise* (Academic Press, New York, NY, 1971).
12. C. Elkan, The foundations of cost-sensitive learning, in *IJCAI*, 2001.
13. W. T. Clark and P. Radivojac, *Bioinformatics* **29**, i53 (2013).
14. Y. Jiang *et al.*, *Bioinformatics* **30**, i609 (2014).
15. C. Dessimoz *et al.*, *Trends Genet* **29**, 609 (2013).
16. S. Fields and O. Song, *Nature* **340**, 245 (1989).
17. A. Mohan *et al.*, *PLoS Comput Biol* **5**, p. e1000497 (2009).
18. F. Denis *et al.*, *Theor Comput Sci* **348**, 70 (2005).
19. M. D. Reid and R. C. Williamson, *J Mach Learn Res* **11**, 2387 (2010).
20. C. Elkan and K. Noto, Learning classifiers from only positive and unlabeled data, in *KDD*, 2008.
21. S. Jain *et al.*, Estimating the class prior and posterior from noisy positives and unlabeled data, in *NIPS*, 2016.
22. S. Jain *et al.*, Recovering true classifier performance in positive-unlabeled learning, in *AAAI*, 2017.
23. M. O. Noordewier *et al.*, Training knowledge-based neural networks to recognize genes in DNA sequences, in *NIPS*, 1990.
24. J. L. Reyes-Ortiz *et al.*, *Neurocomputing* **171**, 754 (2016).
25. R. L. S. Torres *et al.*, Sensor enabled wearable RFID technology for mitigating the risk of falls near beds, in *IEEE RFID*, 2013.
26. B. Strack *et al.*, *Biomed Res Int* **2014**, p. 781670 (2014).
27. M. Riedmiller and H. Braun, A direct adaptive method for faster backpropagation learning: the RPROP algorithm, in *IEEE ICNN*, 1993.
28. S. Jain *et al.*, *arXiv:1601.01944* (2016).
29. A. K. Menon *et al.*, Learning from corrupted binary labels via class-probability estimation, in *ICML*, 2015.

PLATYPUS: A Multiple–View Learning Predictive Framework for Cancer Drug Sensitivity Prediction

Kiley Graim*, Verena Friedl, Kathleen E. Houlahan† and Joshua M. Stuart∓

Dept. of Biomolecular Engineering, University of California, Santa Cruz, CA 95064, USA,
∓*E-mail: jstuart@ucsc.edu*

Cancer is a complex collection of diseases that are to some degree unique to each patient. Precision oncology aims to identify the best drug treatment regime using molecular data on tumor samples. While omics-level data is becoming more widely available for tumor specimens, the datasets upon which computational learning methods can be trained vary in coverage from sample to sample and from data type to data type. Methods that can 'connect the dots' to leverage more of the information provided by these studies could offer major advantages for maximizing predictive potential. We introduce a multi-view machine-learning strategy called PLATYPUS that builds 'views' from multiple data sources that are all used as features for predicting patient outcomes. We show that a learning strategy that finds agreement across the views on unlabeled data increases the performance of the learning methods over any single view. We illustrate the power of the approach by deriving signatures for drug sensitivity in a large cancer cell line database. Code and additional information are available from the PLATYPUS website https://sysbiowiki.soe.ucsc.edu/platypus.

Keywords: Pattern Recognition; Machine Learning; Multiple View Learning; Cancer; Drug Sensitivity; Incompleteness; Unlabeled Data; Semi-Supervised; Co-Training; Integrative Genomics; Systems Biology; Multidimensional; Multi-Omic

1. Introduction

Predicting whether a tumor will respond to a particular treatment strategy remains a challenging and important task. However, the availability and cost of screening compound libraries for a tumor sample remains prohibitive. At the same time, the use of genomic assays, such as DNA and RNA sequencing, for clinical decision making are on the rise. As the costs for these high-throughput assays drop, applying 'genomic signatures' from machine-learning trained on external data in place of the more expensive direct drug assay becomes an option.

One obstacle to achieving this goal is the ability to find training sets for machine-learning classifiers for which comprehensive clinical outcomes are available, *e.g.* survival or drug sensitivity. Non–uniformity of large composite datasets such as The Cancer Genome Atlas (TCGA, *cancergenome.nih.gov*) forces many existing approaches to ignore data unless it is available for all samples. At the same time, many studies have samples that would be useful to analyze

*Currently at the Flatiron Institute & Princeton University
†Currently at the Ontario Institute of Cancer Research

beyond their original purpose, yet cannot be included because they lack outcome data.

The large number of variables compared to far fewer samples can often result in biologically irrelevant solutions.[12] However, issues related to the over-determined nature of the problem sets can be minimized by using prior knowledge to inform feature selection techniques. Incorporating this information can guide learning methods to both more generalizable and interpretable solutions. For example, several approaches that include database-mined gene–gene interaction information have shown promise for interpreting cancer genomics data and utilizing it to predict outcomes.[1,10,16,18] In addition, ensembles can reduce error caused by small sample sizes.[17]

We present a multiple view learning (MVL) framework called PLATYPUS (**P**rogressive **LA**bel **T**raining b**Y** **P**redicting **U**nlabeled **S**amples) that combines the advantages of the knowledge-driven and ensemble approaches. 'Views' are feature extractions of particular data platforms that encode specific prior knowledge and are each allowed to vote on the predicted outcome, providing a more complete and diverse glimpse into the underlying biology. The framework infers outcome labels for unlabeled samples by maximizing prediction agreement between multiple views, thus including more of the data in the classifiers. It reduces overfitting caused by small sample sizes both by predicting labels for unlabeled samples and by incorporating prior knowledge.[8]

A typical approach in machine learning is to train classifiers on a subset of samples containing all of the data, impute missing data, or train ensembles based on data availability, but are generally restricted to samples with the majority of the data for each sample.[20] The semi–supervised MVL approach learns missing patient outcome labels, thus allowing the use of all available labeled and unlabeled datasets. PLATYPUS trains on one or more views and then co-trains on the unlabeled samples. By doing this, PLATYPUS can make predictions on any patient regardless of data availability. This increases overall classifier accuracy while also finding solutions that generalize to the entire population– which has proven extremely difficult in high–feature, low–sample problems.[2] A comparison of PLATYPUS to other related methods is provided in Supplemental Section S1.

2. System and methods

2.1. *Data*

At the time of download the Cancer Cell Line Encyclopedia (CCLE) contained genomic, phenotype, clinical, and other annotation data for 1,037 cancer cell lines,[7] described in Section S2. Of these, drug sensitivity data was available for 504 cell lines and 24 drugs. Drug response was converted to a binary label in order to transform the regression problem into a classification problem. For each compound, cell lines were divided into quartiles ranked by ActArea; The bottom 25% were assigned to the 'non–sensitive' class and the top 25% to the 'sensitive' class. Cell lines lying in the middle were marked with 'intermediate' and considered unlabeled in this test (Fig. S2). Note that these samples are often the most difficult to classify as they represent those with a range of sensitivities that may span orders of magnitude where the growth inhibition curve has its steepest changes as a function of drug concentration. Thus, the ability to input a binary designation for the growth inhibition using a co-training strategy could in

itself have advantages over approaches that identify cutoffs in the drug response curves that are more-or-less arbitrary, without the use of a clear optimization criteria, and without the ability to make use of genomic signatures.

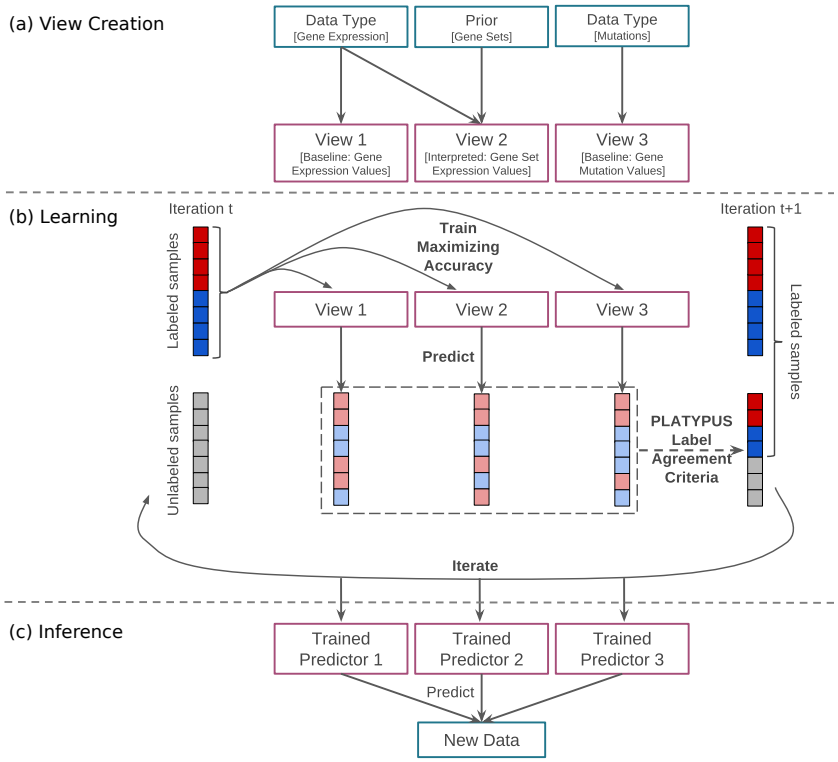


Fig. 1. PLATYPUS framework illustrated with three views. (a) Creation of single views using sample data and optional prior knowledge. (b) Iterative Learning: Each view maximizes prediction accuracy on the labeled samples; unlabeled samples predicted with high confidence are added to the known sample set; repeat until no new samples are labeled. (c) Models from the final iteration of PLATYPUS training applied to new data.

2.2. *Single views and co-training*

PLATYPUS uses co–training (Fig. 1) between single views to learn labels for unlabeled samples. Single views are based on different feature sets. Genomic or clinical features can be used directly (baseline views), or transformed using a biological prior (interpreted views). We built four baseline views from the CCLE data: expression, CNV, mutation, Sample- and Patient-Specific (SPS) information; and many interpreted views (Section S3). Each view can be set up with the best suited machine learning algorithm and optimized parameters for its task, e.g. a random forest or an elastic net (Section S5.1).

Co–training works by training a separate classifier using each view as a separate feature set to make independent predictions, then incorporating disagreement into the loss function. Each view trains on the labeled data then predicts labels for the common unlabeled set. High confidence labels are passed as truth in the next iteration. Co–training methods iterate until

either convergence, some threshold (a minimal change in label definition on the unlabeled samples) is attained, or a maximum number of iterations is reached.

After co–training, each view can be used as a standalone classifier that incorporates learning from one or more data platforms without relying solely on that data platform. Since views are trained in conjunction, the trained models will incorporate the perspectives of all views. This also provides a measure of influence from all views when applying any of the classifiers to new data, without requiring data for those views when making predictions.

2.3. *Maximizing agreement across views through label assignment*

The key step in the PLATYPUS approach is the inference of outcome labels for a set of unlabeled data. Each training iteration seeks to improve the agreement of the assignments given to the unlabeled data across all views. Views are first created by applying machine learning methods using either the features directly, or from gene set summaries or subsetting (Section S3). Fig. 1 shows an overview of PLATYPUS using three views. Any number of views may be used– in this paper, up to 10 views are used per experiment.

PLATYPUS searches iteratively for a label assignment that improves the agreement on unlabeled data (Fig. 1(b)). At each iteration t, the views are trained on labeled data and the labels for unlabeled samples are inferred. Because the set of labels can change across iterations, we denote the training data with sensitive labels as $T^+(t)$ and those with non–sensitive labels as $T^-(t)$ at iteration t. $T^+(0)$ and $T^-(0)$ are the given sets of sensitive and non–sensitive training samples before learning labels, respectively. The set of unlabeled samples is denoted $U(t)$, with all unlabeled samples before learning labels as $U(0)$.

V is the set of views used in the PLATYPUS run. In iteration t, each view $v \in V$ is trained to maximize its prediction accuracy on the labeled samples $T^+(t)$ and $T^-(t)$. The accuracy of view v at iteration t is determined using cross–validation of the training samples and is written here as $a(v,t)$, where $a(v,0)$ is the single view accuracy before learning labels. A prediction is then made by the trained models for each unlabeled sample s. Let $l(v,s,t)$ be the prediction of sample s by view v in iteration t where it is 1 if predicted sensitive and 0 otherwise. The single view votes are summarized to a sensitive ensemble vote $L^+(s,t)$ and non–sensitive ensemble vote $L^-(s,t)$ for each sample (Eq. 1 and 2).

$$L^+(s,t) = \sum_{v \in V^s} w(v,t)\, l(v,s,t) \qquad (1)$$

$$L^-(s,t) = \sum_{v \in V^s} w(v,t)(1-l(v,s,t)) \qquad (2)$$

Only views with data to predict sample s are taken into account: $V^s = \{v \in V : v \text{ has data for } s\}$; and the different views are weighted by $w(v,t)$ (Eq. 3). View accuracies within $[0.5, 1]$ are rescaled to $[0, 1]$ and log–scaled. Views with an accuracy lower than 0.5 are given a weight of 0 since it indicates worse than random predictions.

$$w(v,t) = \begin{cases} -log(1 - \frac{a(v,t)-0.5}{0.5}) \text{ if } a(v,t) \geq 0.5 \\ 0 \qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \qquad (3)$$

To determine, which unlabeled samples are added to the training data for the next iteration, we define $L^{\max}(t)$, the strongest vote found between all samples in iteration t (Eq. 4), and

$\Psi(t)$, the set of samples reaching the strongest vote (Eq. 5).

$$L^{\max}(t) = \max_{s \in U(t)} \{\max\{L^+(s,t), L^-(s,t)\}\} \tag{4}$$

$$\Psi(t) = \{s \in U(t) : \max\{L^+(s,t), L^-(s,t)\} = L^{\max}(t)\} \tag{5}$$

In order to favor missing data for a sample over conflicting predictions, we define $L^{\min}(t)$ as $\min_{s \in \Psi(t)}\{\min\{L^+(s,t), L^-(s,t)\}\}$, the weakest contrary vote that is found between all samples in $\Psi(t)$.

All samples meeting both the strongest vote and the weakest contrary vote conditions (Label Agreement Criteria) build the set of new training samples $\mathcal{T}(t)$, which are added to $T^+(t)$ and $T^-(t)$ for the next iteration's training data:

$$\mathcal{T}(t) = \{s \in \Psi(t) : \min\{L^+(s,t), L^-(s,t)\} = L^{\min}(t)\} \tag{6}$$

$$T^+(t+1) = T^+(t) \cup \{s \in \mathcal{T}(t) : L^+(s,t) > L^-(s,t)\} \tag{7}$$

$$T^-(t+1) = T^-(t) \cup \{s \in \mathcal{T}(t) : L^+(s,t) < L^-(s,t)\} \tag{8}$$

To avoid adding predictions with low confidence, $L^{\max}(t)$ needs to stay above a certain value, otherwise no labels are added to the training data in iteration t. This can be adjusted by the learning threshold λ, which represents the fraction of the maximal reachable vote, *i.e.* when all views agree. By default λ is 75%.

The training process continues until a convergence criterion is met: either all labels have been learned, no new labels have been learned in the last iteration, or a maximum number of iterations has been reached. After termination of the learning process, the trained single–view predictors can be used independently or as an ensemble via PLATYPUS (Fig. 1(c)).

3. Results

3.1. *Preliminary experiments to optimize PLATYPUS performance*

We ran 120 different PLATYPUS variants to predict drug sensitivity in the CCLE cell lines to identify the best way to combine the views for this application. As mentioned in the Data Section (Section S2), samples with intermediate levels of sensitivity for a particular drug were treated as unlabeled and used by the co-training to maximize agreement across views. The conversion of this regression problem into a classification problem in which drug sensitivities arbitrarily are discretized into sensitive versus insensitive (top and bottom 25%), reflects the reality of the clinical setting in which a decision must be made to either treat or not treat a particular patient. The test measures the co-training strategy's ability to infer sensitivities for cell lines that are the most difficult to classify.

We first asked whether the interpretive views that use gene set information provide benefit over using only the baseline views (Section 2.2). We then determined a weighting scheme for the ensemble to achieve better performance. We ran PLATYPUS using the 4 baseline views and the 3, 5, 7, and 10 best–performing single views for each of the 24 CCLE drugs at a $\lambda = 75\%$ learning threshold, for a total of 120 different PLATYPUS variants (5 per drug). Fig. 2(a) shows the highest accuracy PLATYPUS models as well as each of the single view scores. In almost all cases PLATYPUS significantly outperforms single view models, most notably for

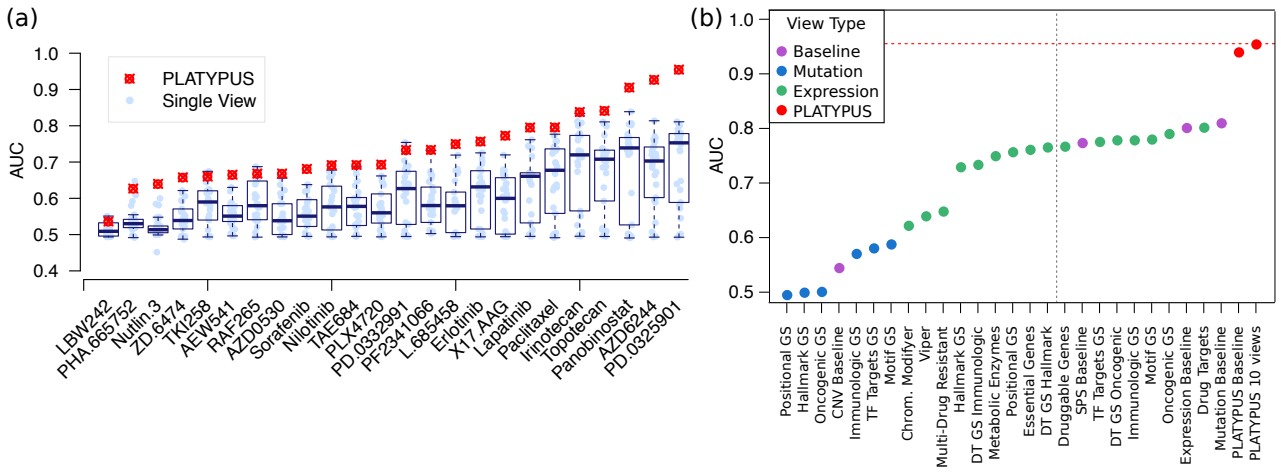


Fig. 2. PLATYPUS Performance. (a) Boxplot showing performance (in AUC) sorted by PLATYPUS score, of all single views and the best PLATYPUS score. PLATYPUS score for each drug is the highest from the 3,5,7, and 10 view runs. (b) AUC for PD-0325901 sensitivity predictions for each single view, colored by view type. The 10 views to the right of the gray line are used in the PLATYPUS ensemble. See Fig. S3 for single view AUCs for all drugs. DT = Drug Target; GS = Gene Set.

the MEK inhibitors AZD6244 and PD-0325901, and HDAC inhibitor Panobinostat. Adding interpreted views to PLATYPUS increased PD-0325901 AUC from 0.94 to 0.99 (Fig. 2(b)), motivating their continued inclusion in PLATYPUS models. Furthermore, within 10 iterations, most PLATYPUS runs added 90% or more of the unlabeled cell lines to the labeled set, effectively doubling the number of samples on which the models trained. We look more closely at the results from the best overall performing PLATYPUS model, PD-0325901, as well as important features from each of its models, in Section 3.2.

We next investigated how to combine the ensemble of different views to improve the PLATYPUS method's accuracy. Previous studies show that combining multiple weak but independent models will result in much higher model accuracy.[17,20] Similarly, previous work has shown that using biological priors can reduce the influence of noise present in biological data.[10,11,18] However, it is not clear how models can be combined in an ensemble to achieve the best results. First, we tested a weighting scheme where each view contributed equally to the final prediction, however this made the model sensitive to information–poor views (data not shown). We then tested an AUC–weighted voting scheme, which derives view weights for the current iteration based on the AUC obtained from the previous iteration (Eq. 3). Doing so allows the PLATYPUS ensemble to incorporate a large number of views, without the need for a pre-selection step, where each view has the opportunity to either become more accurate, and contribute more to the prediction outcome, during label learning, or is effectively ignored if it never reaches a high accuracy.

Figs. S5 and S6 show the effectiveness of label learning validation (LLV) for each of the 24 drugs in CCLE. Most of the drug models learn labels correctly, however model AUC decreases once a model starts to learn labels incorrectly. Over many iterations this can lead to a model where the majority of labels are learned incorrectly (e.g. Nutlin-3, Fig. S6). We found that this

risk can be minimized by setting a high confidence threshold for label learning and by using many information–independent views. In our experiments, LLV consistently helps identify optimal parameters to run PLATYPUS on a given dataset.

Without missing data, PLATYPUS is equivalent to a classic ensemble classifier and often outperforms any single view model. In order to understand the benefits of using additional unlabeled data, we compared the 'ensemble' (first) iteration of PLATYPUS to the final and the 'best' iterations. We define 'best' as the iteration with the highest AUC. Interestingly, in almost all cases, the PLATYPUS AUC is higher than the ensemble AUC (Fig. S4). The use of more samples by PLATYPUS helps ensure a more generalizable model. For the experiments in this paper, we intentionally set a high number for maximum iterations to show how label learning can degrade over time, and therefore the final iteration often scores poorly. Label learning degradation is avoidable by using high label learning thresholds and an appropriate number of iterations.

3.2. *Predicting drug sensitivity in cell lines*

Our analysis focuses on the full CCLE dataset, composed of 36 tumor types. For most drugs, the Sample- and Patient-Specific (SPS) view has the highest starting view performance with AUCs ranging from 0.6 to 0.8, and expression baseline views often performed similarly. The mutation view is effective for some drugs (e.g. MEK inhibitors). Three of the four baseline views are top performers for predicting cancer cell line sensitivity to PD-0325901 (Fig. 2(b)), a MEK1/2 inhibitor. CNV view performance was never high enough to warrant inclusion in PLATYPUS models except as the 'aggregated copy number changes' feature in the SPS view.

Interpreted views often outperform the SPS view (Fig. S3). We found several examples in which a biological prior view outperformed the data–specific view, e.g. Metabolic Enzymes, Drug Targets, and Chromatin Modifying Enzymes are better at predicting Lapatinib sensitivity than the baseline expression predictor. The Drug Target Gene Set Hallmark view outperforms data–specific views in Irinotecan and Panobinostat sensitivity predictions. Such examples can be found for all compounds except for the MEK inhibitors, for which the baseline mutations view is always the top performer.

In general, views incorporating expression data have high accuracy (Fig. S3), whereas mutation views are comparable to a random prediction in most cases. This could be due to the presence of many passenger mutations that have little bearing on cell fitness and drug response. In one notable exception, AZD6244, the Drug Target Mutation view is more accurate than the Drug Target Expression view. Generally, interpreted mutation views outperform their baseline counterpart. For example, the Drug Target Mutation view is more accurate than the baseline mutation view in both Irinotecan and Topotecan. Furthermore, the Drug Target Mutation view trained on PD-0325901 increases the relative feature weights for RAS genes, suggesting that it identifies the exclusivity of RAS/BRAF mutations described in Section S6. However overall, mutation views have low accuracy despite mutations being key to drug sensitivity, indicating that other representations that increase the signal-to-noise ratio of this data should be explored in future work.

The Drug Target Gene Set views created from Molecular Signatures Database (MSigDB)

gene set collections perform well overall, especially on Irinotecan, Topotecan, and Panobinostat (Fig. S3). For most compounds the Drug Target Gene Set Hallmark is more accurate than the Oncogenic and Immunologic. A possible reason is that these gene sets are from the Hallmark collection, which are re–occurring, highly reliable gene sets built from combinations of other gene set collections. Their similar performance could also be due to overlap in the gene sets. We recommend that users test for and subsequently remove highly correlated views before running label learning, and intend to incorporate this into future versions of PLATYPUS. One approach to handling correlated views is to extend the ensemble vote step to use stacked learning instead of the current agreement formula. By training a model on the predictions from each view, PLATYPUS may be better able to handle correlated views by treating them with less weight than more independent views.

In addition to the MSigDB gene set views, master regulator–based predictors via Virtual Inference of Protein activity by Enriched Regulon analysis (VIPER)[13] were tested but are not among the top performing ones for any drug. This could be due to use of a generic regulon as VIPER input rather than tissue–specific versions for each cell line.[13]

The PLATYPUS model for the drug PD-0325901 achieved the highest accuracy of all experiments, with a near perfect AUC. We therefore chose to further investigate the results of this drug to identify the nature by which the MVL approach finds an improved classification. PD-0325901 was initially tested in papillary thyroid carcinoma cell lines and is known to be especially effective in cell lines with BRAF mutations.[14] Since these are frequent in the CCLE data, the high accuracy of the single view models is expected. Fig. 3 shows changes from the ensemble to the 'best' PLATYPUS PD-0325901 models. Single view AUCs mostly increase after several iterations, and feature weights within the models also shift to varying degrees. In the baseline mutations view, RAS gene mutations have higher Gini coefficient changes in the PLATYPUS model than in the ensemble (Fig. 3(c)), indicating increased model importance of those genes. Past studies of the CCLE data[7] and our analysis (Section S6 and Table S3) have found RAS and BRAF mutations in the data tend to be mutually exclusive, both of which are linked to PD-0325901 sensitivity (Fig. S1). Thus, PLATYPUS is better able to identify the dual importance of RAS/BRAF mutations than the single view and ensemble models.

We also chose to look at a case where PLATYPUS failed to achieve an improvement. LBW242 is one such case. The single views for this drug all have near random scores. However, instead of identifying an improvement through view combination as is the usual case in our experiments (e.g. PHA-665752 and Nutlin-3), the PLATYPUS models also achieved near random performance (Fig. 2(a)). Further investigation reveals that the performance may not be the fault of PLATYPUS. Instead, little signal may be available in the drug sensitivity labels for this case due to our quantization strategy (i.e. using the upper- and lower-quartiles for the resistant and sensitive classes). The dose-response curve for LBW242 shows very few of the CCLE cell lines may be truly sensitive. While our approach creates balanced class sizes and ensures continuity between experiments, finding a more nuanced per–drug cutoff would likely improve model performance. Suboptimal label cutoffs lead to a low signal–to–noise ratio in the labels for a few of the drugs, which in general leads to low classifier performance.[19] It is also possible that the metric for drug sensitivity for some drugs is ineffective. Traditional

methods to quantify sensitivity are dependent on population growth and thus slow–growing cell lines may appear to be resistant to all drugs.[6]

These results are consistent with previous findings that have shown sensitivity to some compounds is easier to predict than others.[9] For example, the two MEK inhibitors (PD-0325901, AZD6244) and Panobinostat have higher overall accuracy in the single view models (Fig. S3). Interestingly, in the case of Panobinostat, the 'Chromatin Modifiers' and 'Positional Gene Set' PLATYPUS views have higher single view accuracy than the baseline expression view, which could indicate that there is an epigenetic effect from chromatin modifiers. We postulate that a small region of the genome has been unwound, lending sensitivity to Panobinostat. PLATYPUS captures this interaction, whereas single view models do not.

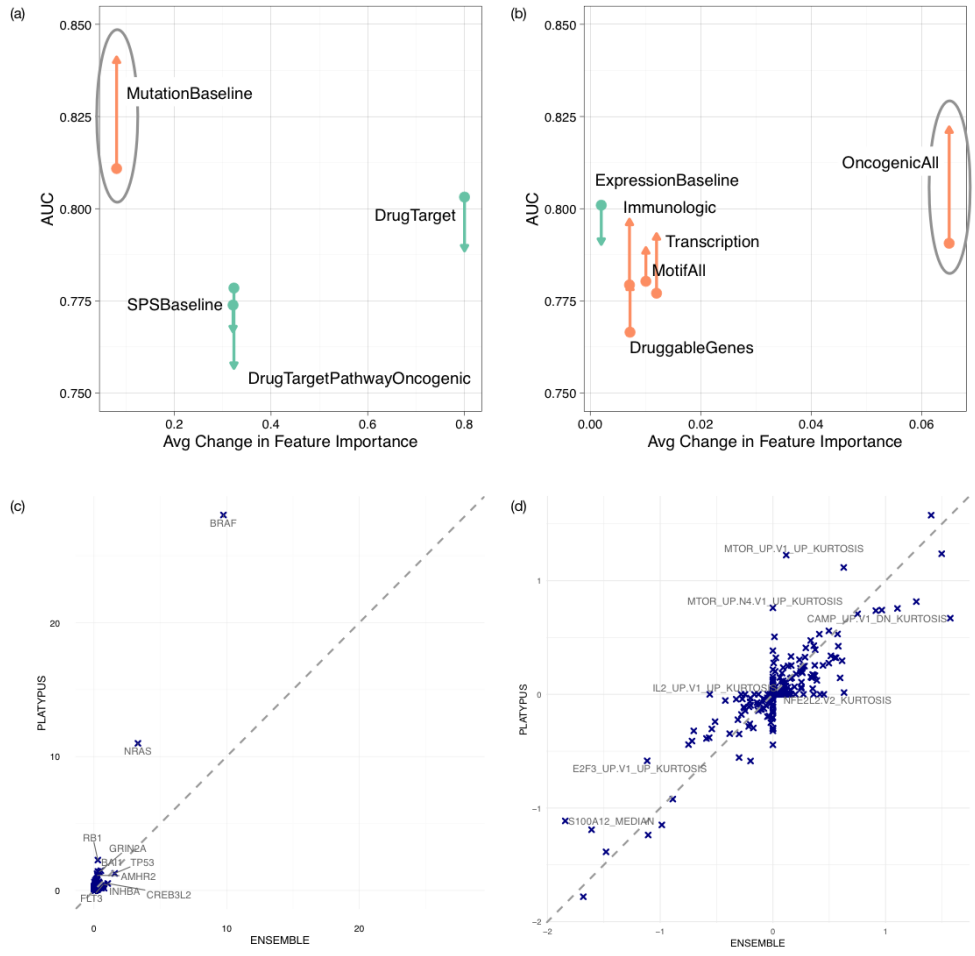


Fig. 3. Performance and feature weight changes for single views between ensemble and PLATYPUS in predicting sensitivity to PD-0325901. (a) For each random forest view, the average Gini change for all features between the ensemble and the best PLATYPUS iteration, plotted against the view AUC for the ensemble (arrow tail) and PLATYPUS (arrow head). Circled view is shown in detail in (c). (b) Same as (a), but showing the elastic net views and their average change in feature weights. (c) Scatter plot where each point is a feature in the Baseline Mutations view. Plot shows the ensemble feature weight versus the PLATYPUS feature weight. (d) Same as (c), but showing feature weight changes in the Oncogenic (OncogenicAll in (b)) view.

3.3. *Key features from PLATYPUS models*

Each machine-learning algorithm used by a view has its own internal feature selection. We extracted features from these models to evaluate the most informative features. Fig. 3(a-b) show changes in single view model performance and average feature importance within those models, before and after PLATYPUS training. Fig. 3(c-d) show feature changes and enrichment of those features within one of the views. Fig. 3(c) highlights how PLATYPUS is able to remove feature weights of spurious correlations between cell line mutations and the true mutation features of importance, NRAS and BRAF. While the overall feature weights in the single view model do not have large changes from the ensemble to PLATYPUS frameworks, there is a large shift in 2 key features which are known to be significantly associated with sensitivity to this particular drug. PLATYPUS is able to avoid overfitting the model whereas the ensemble is unable to draw from external information. In Fig. 3(d), the model has significantly changed both in AUC and in feature weights between the ensemble and PLATYPUS experiments.

Fig. S8 shows a closeup of the changes within the Fig. 3(d) view between PLATYPUS and a general ensemble. It focuses on one feature from the view, MTOR_up_V1_up kurtosis, which had the biggest increase in feature weight from ensemble to PLATYPUS. At a glance, this gene set is not associated with cancer– it describes genes that are regulated by an inhibitor used to prevent graft rejection by blocking cell proliferation signals via mTOR. However, the gene set kurtosis correlates with ActArea and with our binary drug sensitivity labels (Fig. S8(a-b)). A closer look shows that this is because of gene-gene correlations within the gene set. Kurtosis features are intended to capture large changes within the gene set. Mean and median gene set correlation values do not capture cell line differences in the co-correlated gene clusters, whereas kurtosis highlights extreme values. No one gene expression correlates strongly with the kurtosis of the whole set (Fig. S8(c,e)), and so the set cannot be replaced with a single gene expression value. Clusters within the gene set are linked to EGFR signaling (cluster IV, genes marked E), metastasis and Basal vs Mesenchymal BRCA (cluster V, genes marked M and B respectively), and resistance to several cancer drugs (clusters II and V, genes marked R). Gene-gene correlations shown in Fig. S8(d) combine to form the overall kurtosis score. As shown in Fig. S8(e), many genes related to cancer processes are the driving force in the gene set kurtosis score. This highlights how small overall changes combine to improve PLATYPUS accuracy over the ensemble.

Many of the highly ranked features from other models (Fig. S7 shows expression view for PD-0325901, other data not shown) are known oncogenes, for example ETV4 was previously found to be correlated with MEK inhibitor sensitivity.[15] SPRY2, a kinase inhibitor, correlates with BRAF mutation status, both of which are predictive of sensitivity to PD-0325901, AZD6244, and PLX4720. DUSP6 has been named as a marker of FGFR inhibitor sensitivity[4] and a previous study shows a weak inverse correlation between DUSP6 expression and sensitivity to MEK1/2 inhibitors.[3] Thus PLATYPUS recapitulates several known markers of drug sensitivity.

4. Conclusions

When compared to a traditional ensemble and to single view predictors, PLATYPUS often has higher AUC (Fig. 2). The multi-view approach uses the set of unlabeled samples as links between different views to find agreement in the different feature spaces. Since label learning validation shows that labels are learned correctly in most cases, the increase in improve model performance may be due to doubling the number of samples that can be considered while training. In 96% of our experiments, PLATYPUS outperforms an ensemble (Fig. S4). Furthermore, PLATYPUS outperforms 85% of the single views and has higher AUC than *all* of the single views for 17 of the 24 drugs. No one single view consistently outperforms any of the PLATYPUS models In order to retain such high performance without PLATYPUS, a user would need to test all single view models.

Important features from PLATYPUS views (both baseline and interpreted) have previously been linked to drug sensitivity. The approach generally improves AUC while incorporating significantly more data and allowing uncertainty– a necessity in medical research. By combining extracted features from each of the MVL model views, the user is provided a clearer picture of the key facets of sensitivity to each drug. We also investigated the generality of PLATYPUS by applying it to the prediction of an aggressive subtype of prostate cancer and found it generalized to an external validation set not used during training (see Supplemental Section S7). Overall, PLATYPUS enables the use of samples with missing data, benefits from views without high correlation, and is a flexible form of MVL amenable to biological problems.

The PLATYPUS co-training approach has several important advantages. First, it is ideal when samples have missing data, a common scenario in bioinformatics. Imagine a new patient entering a clinic for whom not all of the same data is available as was collected for a large drug trial. A PLATYPUS model trained on the drug trial data is able to predict drug response for this patient without retraining, simply by restricting to views for which there is patient data. For example, a sample with only expression data could be provided predictions using the expression–based views. Predicted label confidence for that sample will be much lower since there are no scores from the missing views, ensuring that labels for samples with complete data will be inferred in earlier iterations than those with missing data. PLATYPUS automatically sets weights for view predictions, implicitly accounting for missing data, and ensuring future predictions are not constrained by limited data. Second, co-training allows for the use of different classification methods for each data type, capturing the strengths of each data type and increasing flexibility in the framework. Third, PLATYPUS is effective when using information–divergent views. Fourth, co–training combines predictions at a later stage in the algorithm, so that views are trained independently. This is ideal for ensemble learning, which has shown to be highly effective when models/views are independent, even with low individual model accuracy.[5,17]

It is worth mentioning some distinct limitations of the approach as a pointer toward future work. First, if missing data correspond to cases that are more difficult to classify, rather than missing at random, the poorer performance of individual views may result in appreciably lower agreement, and thus little benefit in combining views. Second, combining multiple views introduces the need for setting additional parameters (e.g. the agreement threshold). This

requires a user to gain familiarity with the performance of newly incorporated views in test runs before final results can be obtained. Finally, highly correlated views can inflate the agreement voting and down-weigh other, uncorrelated views. A future adjustment could incorporate prediction correlation on the labeled samples for the voting of unlabeled samples.

Acknowledgments

We thank Evan Paull, Dan Carlin, Yulia Newton, Pablo Cordero, Anya Tsalenko, Robert Kincaid, and Artem Sokolov for feedback during PLATYPUS implementation. K.G. was supported by an Agilent Fellowship. V.F. was supported by a PROMOS scholarship of the Technical University of Munich (TUM). J.S. was supported by grants from NCI (U24-CA143858, 1R01CA180778, NHGRI (5U54HG006097), and NIGMS (5R01GM109031).

References

1. T. Turki and Z. Wei, *BMC systems biology* **11**, p. 94 (2017).
2. A. Airola, T. Pahikkala, W. Waegeman, B. De Baets and T. Salakoski, in *Machine learning in systems biology*, 2009.
3. S. Gupta, K. Chaudhary, R. Kumar, A. Gautam, J. S. Nanda, S. K. Dhanda, S. K. Brahmachari and G. P. Raghava, *Scientific reports* **6**, p. 23857 (2016).
4. Y. Nakanishi, H. Mizuno, H. Sase, T. Fujii, K. Sakata, N. Akiyama, Y. Aoki, M. Aoki and N. Ishii, *Molecular cancer therapeutics* , molcanther (2015).
5. A. Bagnall, J. Lines, A. Bostrom, J. Large and E. Keogh, *Data Mining and Knowledge Discovery* **31**, 606 (2017).
6. M. Hafner, M. Niepel, M. Chung and P. K. Sorger, *Nature methods* **13**, p. 521 (2016).
7. J. Barretina, G. Caponigro, N. Stransky, K. Venkatesan, A. A. Margolin, S. Kim, C. J. Wilson, J. Lehár, G. V. Kryukov, D. Sonkin *et al.*, *Nature* **483**, p. 603 (2012).
8. W.-Y. Cheng, T.-H. O. Yang and D. Anastassiou, *PLoS computational biology* **9**, p. e1002920 (2013).
9. J. C. Costello, L. M. Heiser, E. Georgii, M. Gönen, M. P. Menden, N. J. Wang, M. Bansal, P. Hintsanen, S. A. Khan, J.-P. Mpindi *et al.*, *Nature biotechnology* **32**, p. 1202 (2014).
10. J. A. Seoane, I. N. Day, T. R. Gaunt and C. Campbell, *Bioinformatics* **30**, 838 (2013).
11. C. J. Vaske, S. C. Benz, J. Z. Sanborn, D. Earl, C. Szeto, J. Zhu, D. Haussler and J. M. Stuart, *Bioinformatics* **26**, i237 (2010).
12. D. Venet, J. E. Dumont and V. Detours, *PLoS computational biology* **7**, p. e1002240 (2011).
13. M. J. Alvarez, Y. Shen, F. M. Giorgi, A. Lachmann, B. B. Ding, B. H. Ye and A. Califano, *Nature genetics* **48**, p. 838 (2016).
14. Y. C. Henderson, Y. Chen, M. J. Frederick, S. Y. Lai and G. L. Clayman, *Molecular cancer therapeutics* , 1535 (2010).
15. C. C.-Y. Leow, S. Gerondakis and A. Spencer, *Blood cancer journal* **3**, p. e105 (2013).
16. I. S. Jang, R. Dienstmann, A. A. Margolin and J. Guinney, in *Pacific Symposium on Biocomputing Co-Chairs*, 2014.
17. L. Rokach, *Artificial Intelligence Review* **33**, 1 (2010).
18. F. Iorio, T. A. Knijnenburg, D. J. Vis, G. R. Bignell, M. P. Menden, M. Schubert, N. Aben, E. Gonçalves, S. Barthorpe, H. Lightfoot *et al.*, *Cell* **166**, 740 (2016).
19. B. Frénay and M. Verleysen, *IEEE transactions on neural networks and learning systems* **25**, 845 (2014).
20. H.-P. Kriegel and A. Zimek, *Proceedings of MultiClustKDD* (2010).

Computational KIR copy number discovery reveals interaction between inhibitory receptor burden and survival

Rachel M. Pyke
School of Medicine, University of California, San Diego, 9500 Gilman Dr.
San Diego, CA 92093, USA
Email: ramarty@ucsd.edu

Raphael Genolet, Alexandre Harari, George Coukos and David Gfeller
Ludwig Institute for Cancer Research, University of Lausanne, Chemin des Boveresses 155
Epalinges, VD, CH, 1066
Email: raphael.genolet@unil.ch, a*lexandre.harari@chuv.ch, george.coukos@chuv.ch, david.gfeller@unil.ch*

Hannah Carter*
School of Medicine, University of California, San Diego, 9500 Gilman Dr.
San Diego, CA 92093, USA
Email: hkcarter@ucsd.edu

Natural killer (NK) cells have increasingly become a target of interest for immunotherapies[1]. NK cells express killer immunoglobulin-like receptors (KIRs), which play a vital role in immune response to tumors by detecting cellular abnormalities. The genomic region encoding the 16 KIR genes displays high polymorphic variability in human populations, making it difficult to resolve individual genotypes based on next generation sequencing data. As a result, the impact of polymorphic KIR variation on cancer phenotypes has been understudied. Currently, labor-intensive, experimental techniques are used to determine an individual's KIR gene copy number profile. Here, we develop an algorithm to determine the germline copy number of KIR genes from whole exome sequencing data and apply it to a cohort of nearly 5000 cancer patients. We use a k-mer based approach to capture sequences unique to specific genes, count their occurrences in the set of reads derived from an individual and compare the individual's k-mer distribution to that of the population. Copy number results demonstrate high concordance with population copy number expectations. Our method reveals that the burden of inhibitory KIR genes is associated with survival in two tumor types, highlighting the potential importance of KIR variation in understanding tumor development and response to immunotherapy.

Keywords: Killer immunoglobulin-like receptors, KIR, cancer, immunology, MHC, copy number

1. Introduction

Killer Immunoglobulin-like receptors (KIRs) are cell-surface receptors expressed by Natural Killer (NK) cells and some T cells. KIRs bind to other naturally occurring immune receptors, including Major Histocompatibility Complexes (MHCs), to inhibit or activate immune

cell activity[2]. MHC molecules, which are expressed on nearly all nucleated cells, can present pathogenic or tumorigenic peptides on the cell surface for recognition by T cells. In order to evade the immune system, malignant cells often down regulate expression of MHC molecules[3]. However, KIR on NK cells are able to respond with an immune attack if they can recognize that the expression of MHC deviates from normal[4]. This dual system allows "no way out" for cancerous cells -- either the MHC presents the neo-peptides or the MHC is downregulated and NK cells attack the cell[5]. However, the efficiency of this process depends greatly on the ability of the KIR expressed on NK cells to bind to the MHC receptors.

The impact of these NK cell mechanisms in response to malignancies has been validated through the several associations found between KIR genotype and cancer phenotypes. The presence of certain KIR genes can predict response to immunotherapy treatment and survival outcomes in chronic myeloid leukemia and acute myeloid leukemia[6,7]. Associations have also been found between specific KIR genes and susceptibility to several cancers (malignant melanoma, leukemia, nasopharyngeal carcinoma, and cervical cancer)[5,8–1011]. Furthermore, the strength of HLA-KIR interactions plays a functional role and can influence disease susceptibility[12].

However, all of these studies have been performed on cohorts of low sample size due to the difficulty of studying the highly variable KIR region. KIRs are encoded by a cluster of genes on chromosome 19q13.4. Individuals vary widely in the number of KIR genes they carry and in the allelic variation within those genes. The region can contain up to 16 genes but sometimes has as few as four gene, each one with up to 100 known allelic variants.

The highly homologous nature of the KIR genes hampers usage of conventional, computational copy number technologies for short read Next Generation Sequencing (NGS) data. However, the interesting immune implications of the region have led to the development of several experimentally based techniques. One approach uses polymerase chain reaction to amplify the sequences and sequence specific primers to detect particular alleles[13]. Another uses sequence specific oligonucleotides as a first pass and then sequences specific exons to identify allelic variation[14]. Sanger sequencing can also provide long enough reads to cover several genes at a high resolution[14,15]. However, all of these techniques require KIR specific techniques in the data gathering stage. Only two computational alternatives exist that do not require KIR specific techniques in the data gathering stage. KIR*IMP imputes the KIR region from SNP genotype data[16] and PING predicts KIR copy number from NGS data[17]. However, KIR*IMP cannot be applied to large exome datasets and PING requires time consuming read mapping, a potentially biased normalization and manual curation step.

To achieve the computational speed and accuracy required for inferring the KIR types of nearly six thousand cancer patients in order to study tumor phenotypes, we implemented an unsupervised, k-mer based algorithm that leverages large populations to determine copy number (**Figure 1)**. Using this cancer cohort, we discovered that patients in uterine and cervical cancer

survive longer when they have fewer inhibitory KIR genes as compared to patients that have more inhibitory genes.

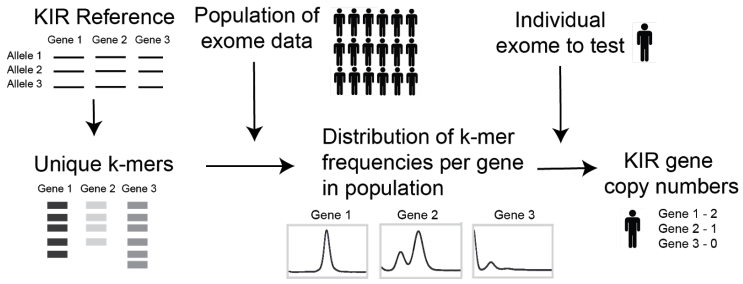


Figure 1. Schematic of copy number calling pipeline. Unique k-mers are derived from a KIR reference library. The exome data for thousands of individuals is searched for these unique k-mers to find distributions of frequencies in the population. The copy number for a specific individual can be deduced from where their frequency falls in the distribution.

2. Materials and Methods

2.1 Data collection

Exome sequencing, transcriptome sequencing and clinical data from The Cancer Genome Atlas was downloaded from the National Cancer Institute's Genomic Data Commons on August 3rd, 2018. All disease types were obtained. KIR alleles were downloaded from the Immuno Polymorphism Database on October 6th, 2016[18]. Population KIR allele frequencies were obtained from The Allele Frequency Net Database on February 22, 2017[19].

2.2 K-mer selection

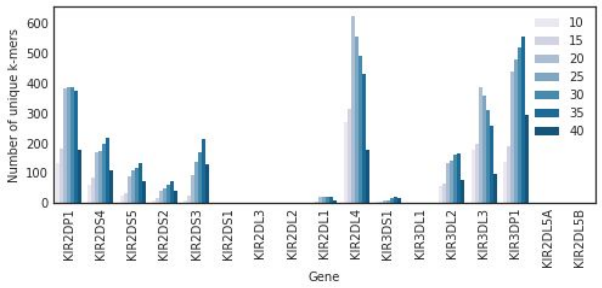


Figure 2. Unique k-mer counts. The number of unique k-mers found in each KIR gene across a spectrum of k.

A set of k-mers were selected to represent each KIR gene -- these k-mers are referred to as unique k-mers. The criteria for the unique k-mers are as follows: a unique k-mer, or its reverse

complement, must appear in (1) every allele of a specific KIR gene and (2) no alleles of any other KIR gene. Unique k-mers of lengths 10, 15, 20, 25, 30, 35 and 40 were collected based on the KIR reference from the Immuno Polymorphism Database (IPD)[18]. The number of unique k-mers for each gene is shown in **Figure 2.** In addition, only one length of k-mer, 30, was collected in 100 random genes from throughout the genome.

2.3 NGS pipeline and k-mer extraction

The genomic region encoding the KIR locus (GRCh38:chr19:54025634-55084318) and the regions encoding the 100 random genes were extracted from the exome sequencing bam files from the TCGA. The unmapped reads of the exome sequencing bam files were also pulled from the exome sequencing bam files. All of these genomic regions were merged together into a single bam file. Then, the reads were stripped into a fastq file and realigned using Bowtie2[20] to a reference that is constructed of all the KIR alleles for each KIR gene from IPD and each of the 100 random reference genes. All reads that mapped in the reference at least once are again stripped and then searched for the set of unique k-mers and occurrence counts are stored for each k-mer. The pipeline concludes with each patient having a vector of occurrence counts for every unique k-mer.

2.4 Data cleaning

To identify substructure in the dataset that might indicate problematic samples, the k-mer frequency for each of a set of 100 random genes for all patients in TCGA are visualised with a t-SNE plot[21]. To further understand the relationship between sequencing depth and clusters of samples, we plotted the distribution of k-mer counts in the set of 100 random genes and also k-mer counts in the KIR region. To reduce noise from outliers, only the samples from the largest cluster of the t-SNE (Agilent Sureselect capture kit) were selected and all samples with < 40,000 k-mer coverage in the set of 100 random genes and < 20,000 k-mer coverage in the set of KIR genes were excluded. After applying these filters, a total of 4,717 samples remained.

2.5 Normalization of k-mer frequencies

Since every sample will have different sequencing depth, the k-mer counts must be normalized before being compared between samples. Furthermore, there are several lengths of k to choose between. We evaluated normalization methods and lengths of k based on reduction in variance of k-mer counts associated with KIR3DL3 which is known to be almost universally diploid. We tested each length of k (15, 20, 25, 30, 35, 40) against each of the following normalization approaches: (1) the mean of the number of k-mers mapped to the set of 100 random genes, (2) the mean of the number of reads with at least one k-mer mapping to the set of 100 random genes, (3) the median of the number of k-mers mapped to the set of 100 random genes and (4) the

median of the number of reads with at least one k-mer mapping to the set of 100 random genes. In the end, we used a k of 30 and normalized with option (1) for the remainder of the analysis.

2.6 Copy number segregation and cutoff selection

KIR genes have varying numbers of unique k-mers (**Figure 1**). After collecting 30-mer occurrences for each gene and normalizing them to the mean of the number of k-mers mapped to the set of KIR genes, we plotted the values for all individuals across the population with a histogram. Kernel density plots show the distribution of unique k-mer counts for each gene (**Figure 3**).

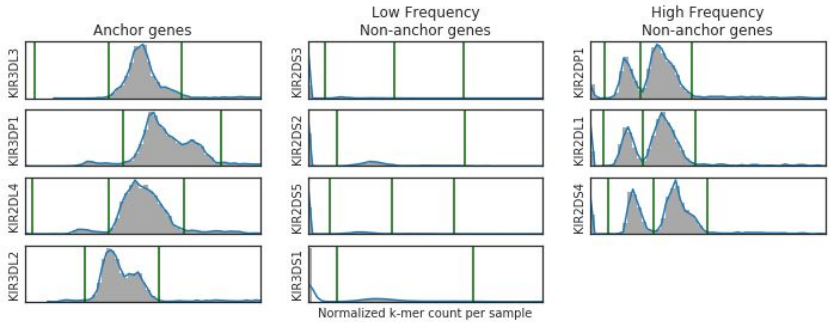


Figure 3. K-mer frequency distribution and copy number thresholds. The distribution of k-mer frequencies across patients in TCGA for anchor genes, high frequency non-anchor genes and low frequency non-anchor genes. The green lines denote copy number thresholds.

These kernel density plots can be used to assign gene copy numbers in an unsupervised manner. First, the genes are divided into three categories based on the documented ploidy of the gene: anchor genes that are present in two copies for most individuals (KIR3DL3, KIR3DP1, KIR2DL2 and KIR3DL2), high frequency non-anchor genes that are present at least once in most individuals (KIR2DP1, KIR2DL1, KIR2DS4 and KIR2DL5) and low frequency non-anchor genes that are present less than once in most individuals (KIR2DS3, KIR2DS2, KIR2DS5 and KIR3DS1). Second, peaks and valleys are called for each kernel density plot by finding local minima of the second derivative. Third, we map the highest peak to the most common ploidy based on the documented copy number variant frequency in the population and determine cutoffs by selecting the valleys surrounding that peak. For anchor genes, the highest peak is determined to be two copy numbers. Samples beyond either edge of the peak (as determined by a second derivative close to 0) are assigned a copy number of 1 or 3+. Instead of looking for subsequent minima, we used the width of the highest peak to create a new threshold for samples with 0 copies to the left of the region for 1 copy. For high frequency non-anchor genes, three peaks are usually observed and thresholds are defined as the valleys between them. The left-most and shortest peak corresponds to 0 copies, the middle peak to 1 copy and the right-most and highest peak to 2 copies. All samples beyond the the third peak correspond to 3+ copies. For low frequency non-anchor genes, typically only two peaks are observed. The

left-most and highest peak is assigned 0 copies and the second peak is assigned 1 copy. The distance between these peaks is used to denote thresholds for the samples that had 2 copies or 3+ copies. Each sample was assigned copy numbers at each KIR gene according to where their k-mer count fell in the distribution. However, patients that fell very close to the cutoff boundaries for a gene (the value of the boundary that splits one copy from two copies divided by 50) were excluded for that gene. All of the genes that do not have any unique k-mers are known to co-segregate with other KIR genes. Thus, we inferred copy number for these genes from the copy number of the co-segregating gene as follows: individuals typically have as many copies of KIR2DS1 as they do KIR3DS1, KIR2DL2 as KIR2DS2, KIR3DL1 as KIR2DS4 and KIR2DL5A as the combined total of KIR2DS3 and KIR2DS5. Furthermore, individuals typically have an inverse number of KIR2DL3 as KIR2DS2 (e.g. 0 KIR2DL3 and 2 KIR2DS2, 1 KIR2DL3 and 1 KIR2DS2 or 2 KIR2DL3 and 0 KIR2DS2).

2.7 Validation of copy number

KIR gene counts for TCGA patients of a specific ancestry are expected to follow the documented distribution of the corresponding population. To validate this assertion, KIR gene frequencies for a European ancestry population from IPD were compared to predicted KIR gene frequencies for the European ancestry patients in TCGA. The correlation between individual gene frequencies was determined using a Pearson correlation.

2.8 Survival analysis

For each tumor type, we divided patients into two sets: those that had the median number of inhibitory genes or fewer and those who had greater than the median number of inhibitory genes. We calculated the survival difference between the two cohorts using the Kaplan Meier and the log rank test as implemented by the lifelines python library. P-values were adjusted with Bonferroni correction. The two tumor types with different survival outcomes, cervical squamous cell carcinoma (CESC) and uterine carcinosarcoma (UCS), were combined because of their similar physical location, immune infiltration profiles and rates in order to increase statistical power.

2.9 Additional immune analysis

We used RNA-seq data from TCGA to obtain immune infiltration predictions with EPIC[22]. Then, we checked the relationship between inhibitor gene count with infiltration of $CD8^+$ T cells and NK cells for the tumor types where significant survival differences were found. P-values were calculated with a Mann-whitney U test between the patient set with high and low inhibitory gene counts. Furthermore, we calculated MHC-I PHBR scores (which represent the ability of a patient to present a specific mutation to the immune system based on their specific HLA alleles) for

each patient's observed driver mutations as outlined in Marty et al.[23] and compared the PHBR scores for CESC and UCS patients with all other patients using a Mann-whitney U test.

3. Results and Discussions

3.1 Establishing unique k-mers

The key challenge for determining KIR gene copy number is the high frequency of reads mapping to multiple places across the homologous region. To address this challenge, we developed an algorithm that capitalizes on distinct k-mers to successfully determine the sequencing coverage of the gene from which each k-mer was derived. To construct our algorithm, we began by building a library of unique k-mers for all KIR genes. A unique k-mer is then defined as a string of length k that appears in *all* alleles of a specific gene but in no alleles of any other gene. The IPD contains all observed alleles of each KIR gene. Using this reference, we searched each gene for unique k-mers and found that all KIR genes either have unique k-mers (**Figure 2**) or are co-inherited with other KIR genes that have unique k-mers[19].

3.2 Varying coverage of KIR region by exome capture kit

Next, we explored The Cancer Genome Atlas (TCGA), a large set of cancer patients (~10,000 individuals) with germline exome sequencing to learn the relationship between k-mer counts and gene copy number. We first evaluated the implication of technical covariates for our analysis. The majority of patients in TCGA had their exome captured with an Agilent capture kit; however, there were several other capture kits used for subsets of patients (**Figure 4A**). We selected 100 random genes in the genome and chose up to 100 unique k-mers from each gene. For each individual, we counted all observations of each k-mer and then normalized each k-mer

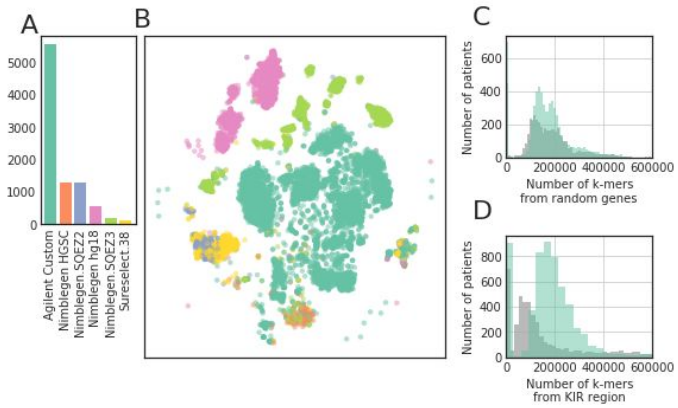


Figure 4. Patient exome data substructure. (A) A bar plot representing the number of patients whose exome data was captured with each exome capture kit. (B) A t-SNE plot representing the clustering of patients based on their k-mer frequency for 100 random genes in the genome. Each sample is colored by their exome capture kit. (C-D) Histograms showing the sequencing coverage of the patients with an Agilent capture kit versus the sequencing coverage of all other patients for (C) 100 random genes in the genome and (D) the KIR genes.

count by the total number of observed k-mers across all 100 random genes found in that individual, resulting in a frequency for each k-mer. Using a t-SNE clustering approach, we discovered that the patients clustered by exome capture kit (**Figure 4B**), suggesting that capture kit could confound k-mer frequency analysis. Among capture kits, the Agilent kit was both the most frequently used kit in TCGA and the kit with the highest coverage of the KIR region. Thus we restricted our analysis to individuals sequenced with this capture kit. Furthermore, we eliminated all patients with low coverage of the 100 random genes or of the KIR region, leaving us with 4,717 high quality individuals.

3.3 Inference of KIR copy number

Next, we searched the reads for each patient mapping to the KIR reference for unique k-mers. Since every patient will have a different sequencing depth, we had to normalize the k-mer counts before comparing them among individuals. Furthermore, we gathered k-mer counts for several lengths of k and wanted to choose the optimal value. Thus, we swept the parameter space, evaluating several normalization techniques and several values for k (**Figure 5A**). We evaluated each approach by determining the variance of frequency for k-mers specific to KIR3DL3 (**Figure 5B**), an anchor gene that is known to be present at two copies in nearly all individuals, under the assumption that lower variance across the population would mean better normalization for sequencing depth differences. We found the optimal normalization technique to be the average k-mer count of the k-mers from the 100 random genes. Though a k of 20 performed the best, we chose to k to be 30 because its performance was very close to optimal and it has higher k-mer coverage of low frequency KIR genes than a k of 20.

Figure 5. Evaluation of optimal normalization. (A) A heatmap representing the variance of k-mer frequency of KIR3DL3 anchor gene across Agilent captured TCGA patients. Several lengths of k and normalization techniques are tested. (B) A histogram showing the k-mer frequency of KIR3DL3 anchor gene with the optimal normalization technique.

After establishing the normalization technique, we calculated the normalized k-mer count over all of the unique k-mers for every KIR gene of each patient. The frequencies were combined across the population to construct density curves showing the proportion of individuals

with similar frequencies. Each KIR gene shows a smooth density curve with peaks that correspond to gene copy number. Anchor genes that are present in all patients have a single peak while the non-anchor genes that are present mostly at 0, 1 or 2 copies have three peaks (**Figure 3**). From the peaks, we determine a cutoff based around the local minima of the population densities. To determine the copy number of a specific individual, we follow the same alignment and k-mer searching approach, followed by the assignment of gene copy number depending on the individual's placement on the curve of each gene. We applied our algorithm to 4,717 individuals in TCGA to assess the copy number of each KIR gene. For most genes, we observed good agreement to copy number calls with PING; however, on genes where the methods disagreed, our method predicted closer to the expected caucasian frequency (**Figure S1A**). Furthermore, our method ran four times as fast as PING on the same hardware (**Figure S1B**).

3.4 Population variation of the KIR region

As anticipated, the distributions of copy number per KIR gene across the population are highly variable (**Figure 6A**). The anchor genes have two copies for nearly all individuals while non-anchor genes have a mixture of copy numbers. To validate our method computationally, we assessed correlation between known KIR copy number frequency against our algorithm. The results were very promising; there was a high correlation (R^2 = .999) between ancestry-matched population frequencies of KIR haplotypes in TCGA and a recent study that used an experimental approach for typing[24] (**Figure 6B**). This finding also suggests little or no germline KIR-based cancer predisposition; however, more comparisons with non-cancer populations will be required to make a definitive assertion.

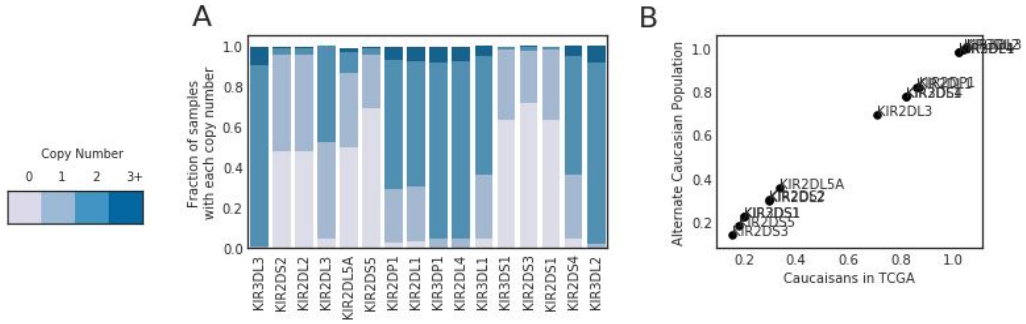


Figure 6. TCGA KIR copy number distribution and validation. (A) A stacked bar chart showing the fraction of patients with each copy number across all KIR genes. (B) A dot plot showing the comparison in gene frequency (average gene copy number per haplotype) within the European ancestry population of TCGA and an experimentally typed European ancestry population.

3.5 KIR inhibitory gene burden correlates with survival in cervical and uterine cancer

KIR genes are divided into two functional categories: activating genes and inhibitory genes. Inhibitory genes bind to specific MHC-I ligands to inhibit the NK cell from attacking the MHC-I expressing cell[12,25]. Often in cancer, cells will down regulate their MHC-I molecules to avoid immune presentation of neoantigens. When this happens, there is no inhibition of the NK

cells by the KIR, and NK cells attack. Activating genes have remained more elusive with their ligands and function mainly unknown[12]. They are believed to have evolved after the inhibitory genes and are non-essential to proper immune functioning. Since inhibitory genes are variable in copy number across individuals, we tested survival differences within tumor types for patients with high and low numbers of inhibitory gene copies. We found two tumor types, cervical squamous cell carcinoma (CESC) and uterine carcinosarcoma (UCS), with unadjusted p-values of less than 0.05 (P=0.000182 and P=0.0113, respectively). In both of these tumor types, patients with high numbers of inhibitory genes had lower survival rates, suggesting that NK cells were unable to defend against the tumor in these patients. Since these tumor types are physically co-localized and have similar immune infiltration profiles and survival rates (**Figure S2**), we analyzed these cohorts together to increase sample sizes (adj P=0.00612, **Figure 7A**).

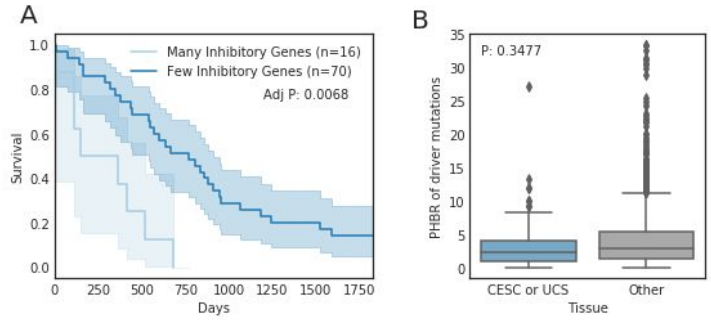


Figure 7. The impact of KIR copy number on tumor development phenotypes in CESC and UCS. (A) Kaplan-meier survival curves denoting the difference in survival between patients with more inhibitory genes than average and less inhibitory genes than average. (B) A boxplot showing the difference in MHC-I presentation of driver mutations between CESC and UCS.

To investigate why we found a significant survival difference in these two tumor types as compared to others, we explored the ability of their MHC-I to present observed driver mutations for recognition by the immune system[23]. Patients with CESC and UCS had better presentation of observed driver mutations to the immune system than other tumors (P=0.0034, **Figure 7B**), suggesting that the CESC and UCS tumors have immunosuppressive mechanisms at play. One possible mechanism for this immunosuppression is impaired antigen presentation, potentially via mutation[3] or loss of heterozygosity in the HLA region[26], allowing perpetuation of the tumor despite high affinity of observed drivers for the MHC-I. If MHC-I presentation on the cell surface is altered and T cells become less relevant, we expect that individuals with higher inhibitory KIR gene counts would have less ability to initiate an NK based attack against the tumor. These observations suggest that when NK cells are called to action, patients with higher NK cell inhibition may be less able to attack the cancer cells, resulting in a shorter survival time.

5. Conclusions

Though natural killer cells are increasingly being considered as targets for immunotherapy, little is understood about the role of KIR, their main receptor family, on tumor

development. Here, we describe our effort to evaluate the copy number of KIR genes in a large cancer cohort to learn about their influence in relationship with MHC on tumor development. We demonstrate the value of algorithmically learning KIR copy number in a large population by uncovering a survival difference in CESC and USC based in the number of inhibitory genes carried by an individual. Due to batch effects in exome sequencing, the current method must be retrained on each new cohort of individuals. This limitation leaves us unable to validate many of our methods experimentally. Furthermore, our method does not provide allele calls and cannot be used to determine the copy number of small cohorts or individual patients. However, our analysis highlights the importance of KIR variability to tumor development and warrants further study of this complex locus.

6. Acknowledgements

We would like to thank Alexandra Buckley for the exome capture kit assignments and the Gfeller Lab and Carter Lab for scientific discussion. Furthermore, we would like to acknowledge the TCGA research network for providing data used in the analyses. This work was supported by a NSF graduate fellowship #2015205295 to R.M., NIH grants DP5-OD017937, RO1 CA220009 and a CIFAR fellowship to H.C., P41-GM103504 for the computing resources, as well as the Cancer Cell Map Initiative U54CA209891 supported by the Fred Luddy Family Foundation.

7. Supplementary Material

http://carter.ucsd.edu/papers/pyke2019/Supplementary%20information.pdf

References

1. Hofer, E. & Koehl, U. Natural Killer Cell-Based Cancer Immunotherapies: From Immune Evasion to Promising Targeted Cellular Therapies. *Front. Immunol.* **8,** 745 (2017).
2. Yeung, D. T. *et al.* KIR2DL5B genotype predicts outcomes in CML patients treated with response-directed sequential imatinib/nilotinib strategy. *Blood* **126,** 2720–2723 (2015).
3. Shukla, S. A. *et al.* Comprehensive analysis of cancer-associated somatic mutations in class I HLA genes. *Nat. Biotechnol.* **33,** 1152–1158 (2015).
4. Martner, A. *et al.* NK cell expression of natural cytotoxicity receptors may determine relapse risk in older AML patients undergoing immunotherapy for remission maintenance. *Oncotarget* **6,** 42569–42574 (2015).
5. Naumova, E. *et al.* Genetic polymorphism of NK receptors and their ligands in melanoma patients: prevalence of inhibitory over activating signals. *Cancer Immunol. Immunother.* **54,** 172–178 (2005).
6. Verheyden, S., Bernier, M. & Demanet, C. Identification of natural killer cell receptor phenotypes associated with leukemia. *Leukemia* **18,** 2002–2007 (2004).
7. Butsch Kovacic, M. *et al.* Variation of the killer cell immunoglobulin-like receptors and HLA-C genes in nasopharyngeal carcinoma. *Cancer Epidemiol. Biomarkers Prev.* **14,** 2673–2677 (2005).
8. Carrington, M. *et al.* Hierarchy of resistance to cervical neoplasia mediated by

combinations of killer immunoglobulin-like receptor and human leukocyte antigen loci. *J. Exp. Med.* **201,** 1069–1075 (2005).
9. Bessoles, S. *et al.* Adaptations of Natural Killer Cells to Self-MHC Class I. *Front. Immunol.* **5,** (2014).
10. Bubeník, J. MHC class I down-regulation: tumour escape from immune surveillance? (review). *Int. J. Oncol.* **25,** 487–491 (2004).
11. Kulkarni, S., Martin, M. P. & Carrington, M. The Yin and Yang of HLA and KIR in human disease. *Semin. Immunol.* **20,** 343–352 (2008).
12. Rajagopalan, S. & Long, E. O. Understanding how combinations of HLA and KIR genes influence disease. *J. Exp. Med.* **201,** 1025–1029 (2005).
13. Ordóñez, D., Moraru, M., Gómez-Lozano, N., Cisneros, E. & Vilches, C. KIR typing by non-sequencing methods: polymerase-chain reaction with sequence-specific primers. *Methods Mol. Biol.* **882,** 415–430 (2012).
14. Lebedeva, T. V., Ohashi, M., Zannelli, G., Cullen, R. & Yu, N. Comprehensive approach to high-resolution KIR typing. *Hum. Immunol.* **68,** 789–796 (2007).
15. Hou, L. *et al.* Killer cell immunoglobulin-like receptors (KIR) typing by DNA sequencing. *Methods Mol. Biol.* **882,** 431–468 (2012).
16. Vukcevic, D. *et al.* Imputation of KIR Types from SNP Variation Data. *Am. J. Hum. Genet.* **97,** 593–607 (2015).
17. Norman, P. J. *et al.* Defining KIR and HLA Class I Genotypes at Highest Resolution via High-Throughput Sequencing. *Am. J. Hum. Genet.* **99,** 375–391 (2016).
18. Robinson, J., Halliwell, J. A., McWilliam, H., Lopez, R. & Marsh, S. G. E. IPD--the Immuno Polymorphism Database. *Nucleic Acids Res.* **41,** D1234–D1240 (2012).
19. González-Galarza, F. F. *et al.* Allele frequency net 2015 update: new features for HLA epitopes, KIR and disease and HLA adverse drug reaction associations. *Nucleic Acids Res.* **43,** D784–8 (2015).
20. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nat. Methods* **9,** 357–359 (2012).
21. Maaten, L. van der & Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **9,** 2579–2605 (2008).
22. Racle, J., de Jonge, K., Baumgaertner, P., Speiser, D. E. & Gfeller, D. Simultaneous enumeration of cancer and immune cell types from bulk tumor gene expression data. *Elife* **6,** (2017).
23. Marty, R. *et al.* MHC-I Genotype Restricts the Oncogenic Mutational Landscape. *Cell* **171,** 1272–1283.e15 (2017).
24. Hou, L. *et al.* Killer cell immunoglobulin-like receptors (KIR) typing by DNA sequencing. *Methods Mol. Biol.* **882,** 431–468 (2012).
25. Hilton, H. G. *et al.* Polymorphic HLA-C Receptors Balance the Functional Characteristics of KIR Haplotypes. *J. Immunol.* **195,** 3160–3170 (2015).
26. McGranahan, N. *et al.* Allele-Specific HLA Loss and Immune Escape in Lung Cancer Evolution. *Cell* **171,** 1259–1271.e11 (2017).

Exploring microRNA Regulation of Cancer with Context-Aware Deep Cancer Classifier

Blake Pyman†, Alireza Sedghi, MSc, Shekoofeh Azizi, PhD, Kathrin Tyryshkin, PhD, Neil Renwick, MD, PhD, Parvin Mousavi, PhD

School of Computing, Queen's University,
Kingston, Ontario K7L 3N6, Canada
†*E-mail: pyman@cs.queensu.ca*
http://www.queensu.ca/

Background: MicroRNAs (miRNAs) are small, non-coding RNA that regulate gene expression through post-transcriptional silencing. Differential expression observed in miRNAs, combined with advancements in deep learning (DL), have the potential to improve cancer classification by modelling non-linear miRNA-phenotype associations. We propose a novel miRNA-based deep cancer classifier (DCC) incorporating genomic and hierarchical tissue annotation, capable of accurately predicting the presence of cancer in wide range of human tissues.

Methods: miRNA expression profiles were analyzed for 1746 neoplastic and 3871 normal samples, across 26 types of cancer involving six organ sub-structures and 68 cell types. miRNAs were ranked and filtered using a specificity score representing their information content in relation to neoplasticity, incorporating 3 levels of hierarchical biological annotation. A DL architecture composed of stacked autoencoders (AE) and a multi-layer perceptron (MLP) was trained to predict neoplasticity using 497 abundant and informative miRNAs. Additional DCCs were trained using expression of miRNA cistrons and sequence families, and combined as a diagnostic ensemble. Important miRNAs were identified using backpropagation, and analyzed in Cytoscape using iCTNet and BiNGO.

Results: Nested four-fold cross-validation was used to assess the performance of the DL model. The model achieved an accuracy, AUC/ROC, sensitivity, and specificity of 94.73%, 98.6%, 95.1%, and 94.3%, respectively.

Conclusion: Deep autoencoder networks are a powerful tool for modelling complex miRNA-phenotype associations in cancer. The proposed DCC improves classification accuracy by learning from the biological context of both samples and miRNAs, using anatomical and genomic annotation. Analyzing the deep structure of DCCs with backpropagation can also facilitate biological discovery, by performing gene ontology searches on the most highly significant features.

Keywords: Deep learning; miRNA; Autoencoder; Cancer classification; PSB

1. Introduction

Following rapid advances in biotechnology (RNA-Seq) and machine learning, mining of high-resolution transcriptomic data has become a promising tool for the discovery of potential RNA cancer biomarkers.[1] However, the ability to use this high-dimensional data to predict cancer is limited by the tendency of large models to overfit available data, known as the curse of dimensionality.[2] This problem can be mitigated by filtering variables, and reducing the dimensionality of input, techniques that can be incorporated in machine learning algorithms.

Deep learning (DL) describes a family of machine learning algorithms designed to model non-linear features at various levels of abstraction, by processing training data over multiple connected layers. DL models are a type of artificial neural network (ANN) which learn by calibrating the weights of connections between nodes by backpropagation of the error gradient. Applying backpropagation to deeper networks can be ineffective due to the problem of "vanishing gradients", but this problem was solved in 2006 by Hinton and Salakhutdinov, who devised a procedure for pre-training hidden layers.[3] Hinton's original formulation used stochastic, binary networks with one hidden layer and symmetrical weights, known as Restricted Boltzmann Machines (RBMs). RBMs were pre-trained such that the hidden layer of one RBM formed the input of the next. After pre-training, the entire model (named a Deep Belief Network, or DBN) could be fine-tuned with supervised learning. Because each layer encodes features based on the previous layer, the higher layers contain increasingly abstract feature sets. In addition, the non-linearity of deep learning results in highly generalizable models that are less dependent on preprocessing and normalization. Both of these characteristics - complex internal structure, and insensitivity to input variance - makes DL models well-suited to transcriptomic applications. DBNs have been used with microarray data to cluster breast cancers[4] and glioblastomas into prognostically relevant subtypes.[5] Deep Boltzmann Machines, a related architecture, have also been used to classify human colorectal carcinomas by subtype.[6] In each of these studies, training data was limited to a single cancer type, permitting subtype discrimination but limiting the scope for transfer learning between cancers. Deep neural nets have also been used to predict cancer type based on genetic data (somatic point mutations) albeit with poorer accuracy than RNA-based models.[7]

The pre-training method applied to DBNs can be generalized for layers with continuous outputs,[8] known as autoencoders, which recreate their input using a single real-valued hidden layer and non-symmetrical weights. Autoencoders can be stacked and pre-trained in a manner analogous to DBNs, and the resulting stacked (or deep) autoencoders (SAE/DAE) can be fine-tuned using supervised learning. It is possible to limit overfitting by imposing a constraint on the sparsity (number of active nodes) of autoencoders in an SAE. Recently, stacked sparse autoencoders have shown promising results classifying cancer sub-types[9,10]

Most studies applying deep learning to RNA-based cancer prediction have focused on the familiar protein-coding variety, mRNA. However, at least 15 types of non-coding RNA are also produced (accounting for approx. 98% of nuclear output), including potentially valuable biomarkers such as microRNAs.[11] microRNA (or miRNA) are a small non-coding class of RNA responsible for post-transcriptional repression of mRNAs. In contrast to over 22,000 mRNA in the human genome, the high-confidence set of miRNAs is limited to just over one thousand.

This relatively smaller input space mitigates the effects of the curse of dimensionality in DL models. In addition, miRNAs individually display much greater tissue- and tumour-specificity than mRNAs, perhaps due to their role as upstream regulators of RNA activity.[12]

To date, most miRNA DL applications have focused on diagnostic or prognostic classification of tumour subtypes. In one study, DBNs were used to select miRNAs to classify six tumour types.[13] Other projects have examined so-called multimodal architectures integrating miRNA expression with other data sources. One study fed a combination of miRNA, mRNA and gene methylation data to a DBN to cluster ovarian and breast cancer samples.[14] Another study combined the same inputs using a 3-layer stacked autoencoder to predict survival time in liver cancer.[15] Combining data from multiple sources may improve results, at the cost of increasing complexity and the risk of overfitting. Instead, the proposed DCC is supplied with concise data situating both samples and miRNAs in a biological context enabling comparisons between related samples and miRNAs.

In this paper, we propose a DL model to predict the presence of cancer based on miRNA sequencing across over 30 human tissues, from approximately 3600 patients, using an ensemble of deep autoencoders. The proposed model leverages the biological context of both samples and sequences. First, hierarchical anatomical annotation was used to score and filter miRNAs based on their information content. In addition, annotation of miRNA cistrons and sequence families were used to create variants of input data, used to train ensembles of classifiers with superior performance than any single component. The DCC also goes beyond cancer classification, by identifying significant miRNAs with backpropagation, and exploring with network visualization. Finally, targets of selected miRNAs were analyzed using gene ontology, to provide insight into the biological nature of the selected miRNAs' association with cancer.

2. Data

Samples were sequenced between September 2008 and December 2015 at Rockefeller University using the Illumina HiSeq. Samples were richly annotated by expert clinicians using over 30 features, including the type of biological material, disease state and anatomical site, as well as expression of 1187 miRNAs. Our original data included samples from body tissues, body fluids and cultured or sorted cells, from a wide array of anatomical sites (Fig. 2). All subsequent analyses were confined to tissue samples, due to greater average sequencing depth and balanced subclasses. Of tissue samples, 2026 (56%) were neoplastic, and 1606 (44%) were either normal or affected by an unrelated diseased. Site-of-origin was described at three different levels, namely organ, organ substructure, and cell type, organized into a 3-level anatomical hierarchy. The dataset includes samples from 26 organs, 6 organ sub-structures and 68 cell types. *Sequence families*: are sets of miRNAs defined on the basis of sequence similarities and represent miRNAs which are likely to share targets - due to the pleiotropy of miRNA targeting, they are likely to have overlapping sets of targeted mRNAs. *Precursor clusters*: are defined to include miRNAs that either share an identical mature form, or are clustered closely in the genome, and are likely to be co-expressed due to shared promoters. Because of this fact, they may be up- or down-regulated in concert, which may indicate involvement in shared (patho-)physiological pathways.[16]

Table 1. Organs represented by the largest number of samples, with number of samples from each.

Source organ	Number
C, SC and other soft tissues	913
Breast	762
Thyroid gland	333
Brain	292
Skin	269
Kidney	244
Hematopoietic and RC system	173
Heart, mediastinum, and pleura	141

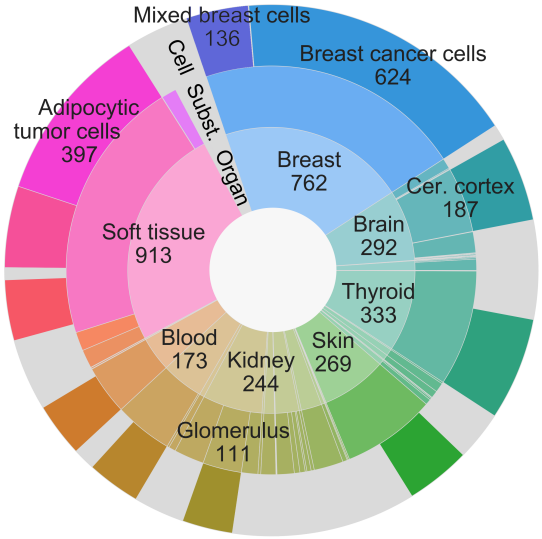


Fig. 2. Left: Eight organs representing the greatest number of tissue samples. Right: Sunburst diagram depicting tissue samples at three anatomical levels: (from the centre outwards) organ, organ substructure and cell type. Only cell types with greater than 100 samples are shown. C = connective, SC = subcutaneous, RC = reticuloenothelial.

2.1. *Preprocessing*

Outlier Removal: We used the inter-quartile range (IQR) to label and subsequently remove outliers and batch effects .[17] Upper and lower bounds were established at a certain distance below the first quartile and above the third quartile of the data and we measured the distance of outlier points beyond the bulk of the data. The distance is usually set at a multiple of the IQR; 1.5×IQR was suggested in previous work and that is the value used herein.[17]

Batch effects were identified using median Spearman coefficient and the bounds established by the IQR method. Batches were removed if at least half their samples were flagged by the IQR method. Following the removal of batches, the Spearman correlations of the remaining points were recalculated. Removal of samples with extreme Spearman values results in tighter bounds, which may enable the detection of further outliers and batch effects. This process was performed iteratively until batch effects could no longer be identified.

Filtering: The initial set of 1187 human miRNAs was filtered based on abundance of expression, eliminating the lowest-expressed miRNAs. An expression threshold was set at 1.41 x 10^{-5}, which corresponds to 14 counts in one million. Any miRNA that was not expressed at or above this level in at least 1% of samples was removed. 397 miRNAs were filtered out on account of low-expression. The remaining miRNAs were also filtered on the basis of information content or cancer specificity.

The specificity of an miRNA, m, for a given tissue, t, ($G_{m,t}$) is a measure of the relative expression level of m in t, compared to other tissues.$G_{m,t}$ can be understood as the proportion of total m expression in all samples that would occur in t, if all classes were sampled from

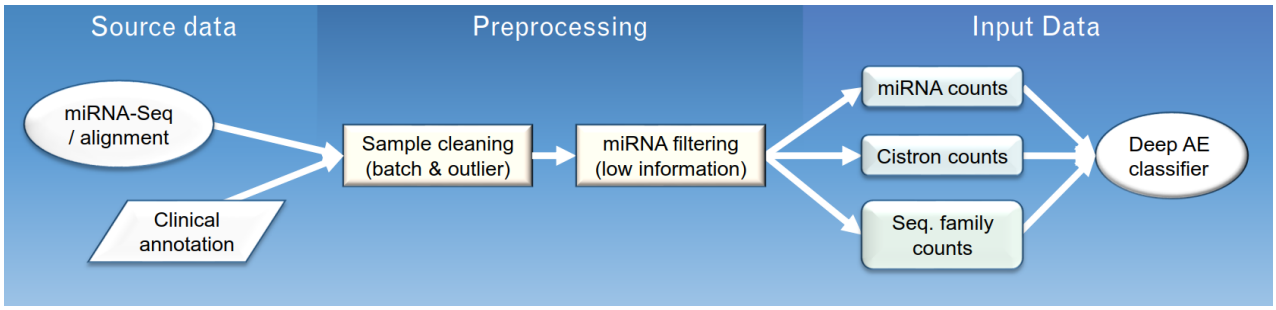


Fig. 1. Flowchart illustrating the datasets used (rounded boxes) and the various transformations and preprocessing steps (rectangular boxes) applied to it, before it is fed as input in the the deep autoencoder model.

equally. $G_{m,t}$ is a value ranging from 0 to 1, representing the specificity of miRNA m for tissue t. If $G_{m,t}$ is close to 1, this indicates that m is expressed much more in t than in other tissues. The specificity of m is determined by the distribution of $G_{m,t}$ for different tissues. If $G_{m,t}$ were the same for all values of t, then the specificity of m would be 0. Once $G_{m,t}$ is determined for all m and t, the total information content of a miRNA, s_m, can be calculated using all $G_{m,t*}$ terms of m for all tissues $t*$.

$$s_m = \log_2(\#oftissues) + \sum_t (G_{m,t} * \log_2(G_{m,t})) \quad (1)$$

Therefore, the maximum possible specificity (i.e. a miRNA expressed solely in one class) is $\log_2(\#ofclasses)$, and cancer specificity of miRNAs ranges from 0 to 1. miRNAs that did not meet a minimum information content threshold ($s_m > 0.01$) were excluded. The remaining 497 miRNAs were inputs to the DCC to predict the presence of cancer (Fig. 1).

Normalization: We used Total Counts Scaling (TCS), in which read counts are divided by the total number of sequenced counts, known as the sequencing depth for normalization.[18] TCS was preferred to more complex methods due to its widespread use and ease of interpretation.[18]

3. Deep Cancer Classifier

The proposed deep cancer classifier (DCC) merges stacked autoencoders with a multilayer feedforward network to accurately classify cancer using miRNA in a range of human tissues (Fig. 2). Data (miRNA, cistron and sequence family expression) was presented to DCC via the input layer. Each successive autoencoder layer is smaller than the last, the layer sizes forming a geometric series. By training each autoencoder in the usual unsupervised manner (minimizing mean squared error with respect to input) it is possible to represent abstract, latent features in the data. These latent features were repeatedly transformed and compressed to 20 in the third AE layer. Following pre-training, the weights of each AE layer were initialized with the weights of the corresponding hidden layers, the AE layers were joined together, and a feed-forward MLP was added. After this step the DCC undergoes supervised learning to boost its classification performance. The model now uses the complex latent features learned in the first stage to predict the presence of cancer. Weights throughout the entire network were fine-tuned through backpropagation to minimize cross-entropy loss of predictions. The proposed

multi-modal architecture of DCC allows for learning multiple layers of latent features from miRNA expression while integrating expert clinical annotation. After developing the model using miRNA profiles using training data, the DCC's performance was tested on left-out samples, and benchmarked against other popular ML methods.

3.1. *Training & testing*

Of the initial 2518 tissue samples, 40% were used as a development set to tune model parameters, while the other 60% were set aside to provide an unbiased measure of the model's performance. This selection was stratified with respect to both cancer status and organ type. Five-fold cross-validation (CV) was performed on the development set, requiring each fifth of the data to provide validation for a model trained on the other four fifths. Once the model's hyperparameters were tuned, the previously unseen test set was used to assess its performance.

An ensemble of development models was used to maximize test set performance (Fig. 2). Each model predicts cancer status as a probability and the output probabilities were averaged over models from different cross-validations. Model variants based on the three input sources (miRNA, cistron and sequence family) were combined in the same way, so each test set prediction was based on an ensemble of $3 \times 5 = 15$ classifiers. These values were finally rounded to 0 (non-neoplastic) or 1 (neoplastic) to establish the number of true and false predictions for each type, and by extension the model's accuracy, sensitivity and specificity.

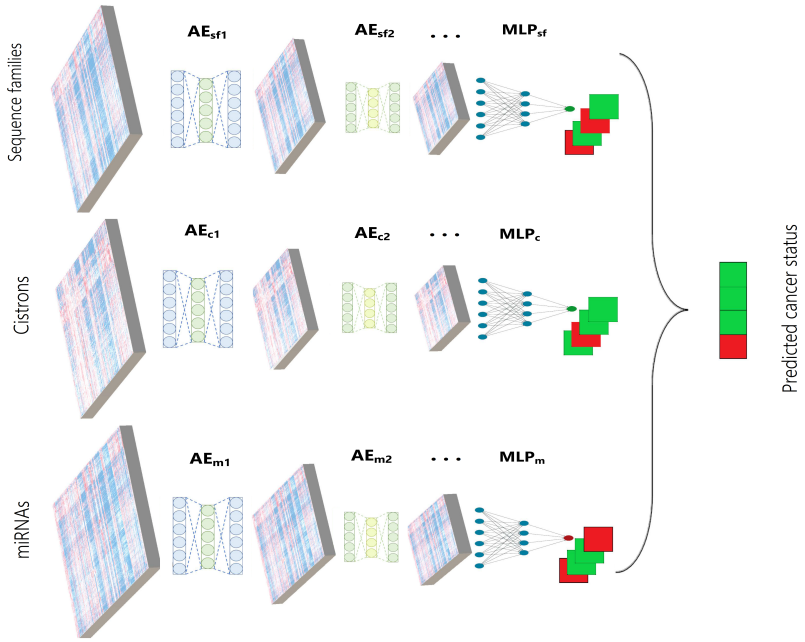


Fig. 2. Simplified schematic of the network topology with three input types (expression of individual miRNAs; miRNA cistron profiles; miRNA sequence family profiles) and main phases in the model's operation. First the AEs undergo unsupervised pre-training, each layer recreating the hidden layer activations of the last, following which the entire model is fine-tuned to classify cancer.

3.2. *Parameter tuning*

Deep learning models are characterized by a large number of configurable parameters, including the overall layout (topology) of the network, initial weights for connections between nodes, and various functions determining the way the network learns (i.e. how weights are updated). By using CV on the development set, different model configurations can be compared, allowing select parameters to be optimized. Even when working exclusively on the development set, there is a risk of overfitting the model if too many parameters are tuned to the development set, reducing the test performance. Therefore, a small number of significant parameters were selected for optimization. Namely, the size the latent features produced by the deep autoencoders, the number of stacked AE layers, the optimizer function to training the autoencoder, and an analogous optimizer for training the classifier.

The size of the smallest AE layer, also called the encoding size, determines the level of compression the input must undergo, and the amount of information available to the classifier. The size of the other AE layers were chosen to form a geometric series, the size of each layer decreasing by a constant factor between the input and final compressed form. The number of AE layers affects the amount of information learned in a different way. Each layer tends to represent different latent features in the data, so deeper networks can capture a greater number of more complex features. The downsides to increased network depth include the risk of overfitting, as well as potentially long training times.[19] Optimizers are algorithms which control the way weights are updated during training, and may control parameters such as (initial) learning rate, momentum and others. It is typically easier and more effective to use these thoroughly tested configurations, instead of varying these parameters independently. Because the performance of any single run is affected by random occurrences (e.g. the splitting of samples, random initialization of weights), the CV optimization procedure was repeated 30 times for each value of each parameter. Based on the distribution of scores for each configuration, the *Kruskal-Wallis (K-W)* test was applied to detect significant differences between groups. If a difference was detected, pairs of samples were compared using the *Mann-Whitney U* test to determine which samples were involved. As the *K-W* and *Mann-Whitney U* tests are non-parametric, no assumptions were made about the normality of the underlying distribution.

3.3. *Feature importance*

The importance of individual miRNAs (or cistrons, etc.) for cancer classification can be estimated using backpropagation, the same algorithm used to train models in supervised learning. However, rather than the gradient of error, we calculated the gradient of activation across input nodes.[20] Signed activation gradients can be computed for every edge between nodes. By taking the sum of the absolute values of activation gradients for all edges connected to a given node, the "contribution" of input features to the activation of higher nodes was determined. A distinct activation is produced in response to each batch of samples presented as input. To calculate the average activation gradient across input features, five DCC variants were trained using 5-fold cross validation. Then, the test set was presented to each variant in 16-sample batches, and the input activation gradients were recorded for each batch. Finally, the gradients were averaged across batches and CV variants, producing a single score for each input feature

(e.g. miRNA) representing its relative importance for classifying cancer.

The miRNAs with the greatest putative cancer association were validated using a network analysis tool called the Integrated Complex Traits Network (iCTNet2).[21] iCTNet2 links numerous biological databases (miRNA, gene, protein, disease, etc.) allowing visualization of indirect associations. Having made a list of miRNAs known to regulate cancer-related genes, one may expect a degree of overlap between this set of cancer-related miRNAs and those returned by the backpropagation method described above. Comparing the number of miRNAs found in both sets to the number expected by chance alone will provide an estimate of level of "cancer enrichment" in the miRNA set produced by the analysis of feature contribution.

Targets of selected miRNAs can also be investigated using gene ontology (GO). BiNGO is a cytoscape app that illustrates gene oncologies as hierarchical networks, with nodes (representing processes) coloured to illustrate their level of enrichment.[22] Enrichment is calculated as over-representation relative to entire GO annotation. This is measured by a p-value, adjusted using the Benjamini & Hochberg correction.

4. Results and Discussion

4.1. *Model selection*

Classification accuracy of the DCC was strongly associated with minimum AE size at the lower end of the tested range. Increasing the encoded size from 5 to 20 caused a clear benefit, although further increases had a null or negative effect (Fig. 3). While compressing miRNA profiles to just five features was clearly sub-optimal, it was still sufficient to classify samples with 93.7% accuracy. A similar trend was observed in relation to layer number. Validation accuracy was greatest when using three stacked autoencoders. Additional AE layers seemed to increase training times, without any significant performance increases. The choice of model optimizers had a strong effect on performance for supervised learning, but pre-training appeared to be relatively insensitive to optimizer choice (at least between the 5 tested configurations). Adagrad exhibited marginally superior performance for pre-training, while Adam was the most effective algorithm for supervised classification via backpropagation (Accuracy = 0.948).

Table 3. Summary of key parameters, with optimal values

Parameter	Optimal value (Range)	Accuracy range
Encoding size	20 (5-60)	0.937 - 0.948
AE layers	3 (1-5)	0.933 - 0.949
AE Optimizer	Adagrad (*)	0.948 - 0.949
MLP Optimizer	Adam (*)	0.844 - 0.948

* Tested optimizers: SGD, RMSprop, Adagrad, Adadelta, Adam

4.2. *Classifier performance*

The most common and intuitive way of assessing the performance of a classifier is its accuracy, given by the number of true predictions over the total number of predictions. The true pre-

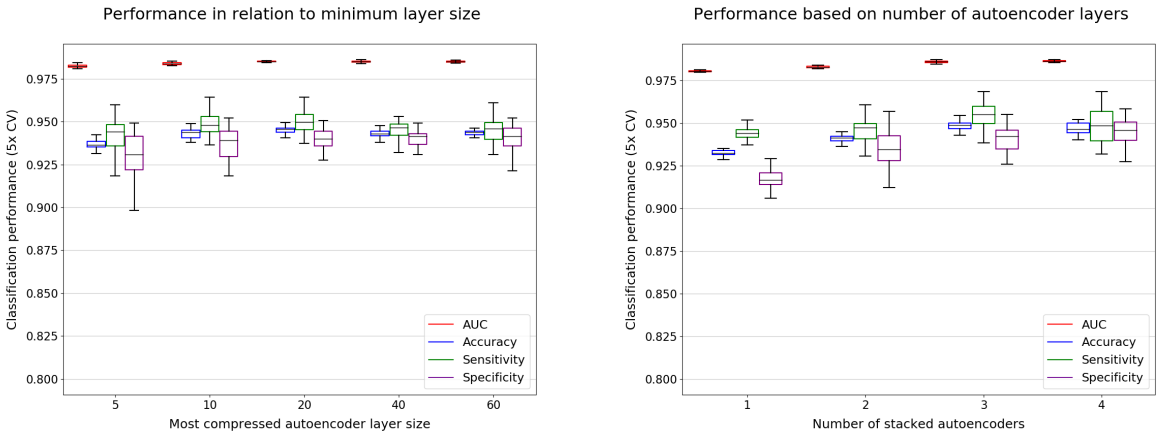


Fig. 3. Left: Test set AUC/ROC, accuracy, sensitivity and specificity in relation to the size of the latent feature representation produced by the stacked autoencoder (with 3 layers). Right: Test set performance (min. encoding size 20), with various numbers of stacked autoencoder layers.

dictions are the sum of the true positive (TP) and true negative (TN) predictions. Accuracy is a suitable metric for problems with similarly-sized target classes, but for highly imbalanced datasets, the success rate for positive and negative samples can be measured using sensitivity or specificity, respectively. Out of 1511 samples, the DCC was able to correctly classify 1421 of them (94.8%). The model had slightly better sensitivity (0.95) than specificity (0.94). The Receiver Operating Characteristic (ROC) illustrates the trade-off between Type I and Type II errors. The Area Under the Curve (AUC) was 0.985.

4.3. *Comparison with other methods*

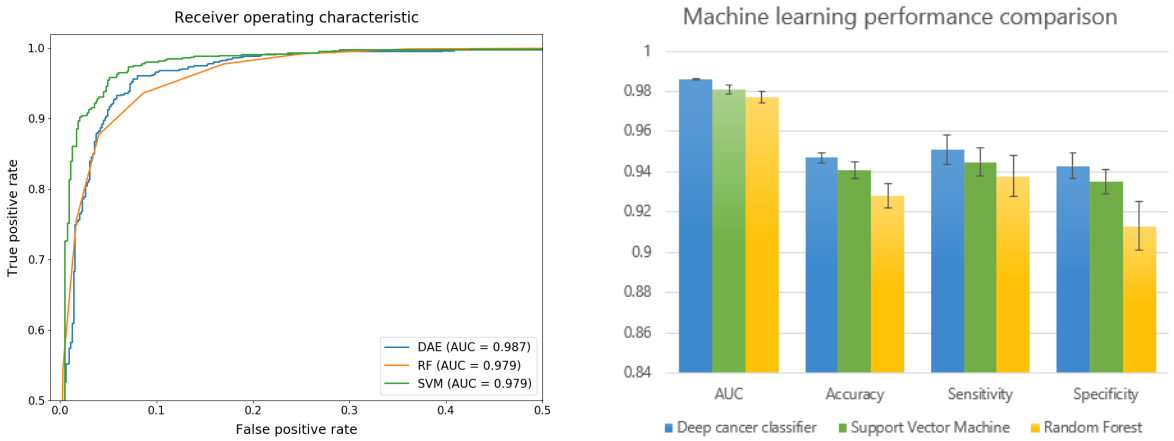


Fig. 4. Left: Receiver Operating Characteristic (ROC) graph showing performance at different thresholds, compared to random forests (RF) and support vector machines (SVM). Right: DCC performance compared with RF and SVM. Error bars show std. dev. for 30 trials, with 5xCV.

The model's performance on test set of 1510 samples was compared to the that of two well-known machine learning methods, support vector machines (SVM) and random forests (RF). The proposed DL model significantly surpassed the performance of a SVM with a linear kernal, and C=1. The example ROC on the following page shows that DCC outperforms SVM

and RF at every error threshold, with an average Area Under the Curve (AUC) of 0.9854.

4.4. *Feature importance*

Backpropagating the activation of the output node to the input nodes enables an estimation of the contribution each feature makes to the model's output. The top 20 miRNAs by output contribution are shown below (Fig. 5). The distribution of activation gradients across miRNAs was highly skewed; while the maximum gradient (for miR-21) was 22.9, only 26 of the 1187 features had an average activation gradient greater than 1. It would appear most of the information required to classify samples is concentrated in a small number of sequences.

iCTNet was used to link miRNAs, genes(/proteins) and human cancers, outputing a list of 61 miRNAs linked to cancer-associated genes (Fig. 5). iCTNet uses a different miRNA reference library (miRCat); after converting miRNAs to a common form and collapsing duplicates, the iCTNet network was reduced to 46 miRNAs, of which 44 were present in our reduced set of 582 miRNAs. Of the top 20 miRNAs by average output activation gradient, 8 were found in the list of cancer-linked miRNAs. Since 7.6% of the miRNAs in the larger list are present in the cancer-associated network, the expected number of matches (based on the binomial distribution) is just 1.5. Therefore, the backpropagation method returned a set of miRNAs with a cancer enrichment of $8/1.5 = 5.3$.

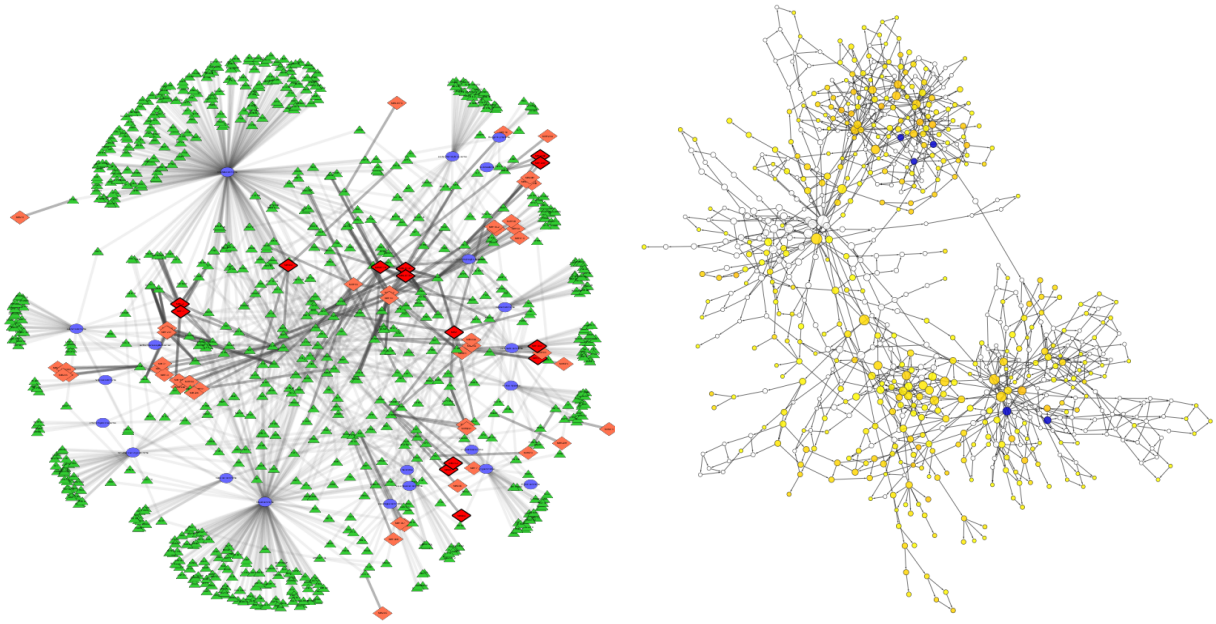


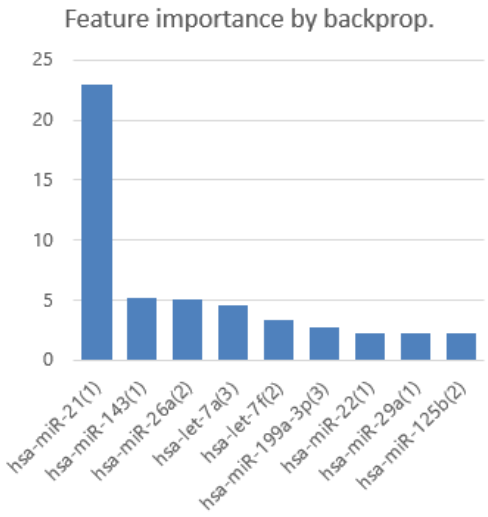
Fig. 5. Left: Network graph showing links between cancers (blue), genes/proteins (green) and miRNAs (red). Of the 44 cancer-linked miRNAs shown, 8 ranked in the top 20 by output activation gradient (bright red, outlined). Right: Gene ontology graph for genes targeted by selected miRNAs. Node color represents p-value of category over-representation. Five most over-represented categories highlighted in blue.

Sixteen miRNAs from the ICTNet graph are among the top 50 miRNAs by activation gradient. Gene ontology analysis was performed on 49 genes linked to these miRNAs (Fig. 5).

The graph displays a hierarchical GO network, containing at least two major clusters of related enriched biological processes. The bottom-right cluster stems from "Regulation of cellular process" and "Regulation of biological process", and contains "Positive regulation of cellular process" and "Regulation of cell proliferation". The other 3 of the top 5 most highly-enriched GO categories (Table 4) are found in the uppermost cluster, descended from "Developmental process", and "Multicellular organismal development".

Table 4. Most over-represented gene ontology categories linked to selected miRNAs

Gene ontology category	Adj. p-value	# of genes
Positive regul'n of cellular process	4.42E-5	22/49
Regulation of cell proliferation	3.60E-5	15/49
Epithelium development	4.42E-5	10/49
Tissue morphogenesis	1.84E-5	10/49
Morphogenesis of an epithelium	1.84E-5	9/49



5. Conclusion

The proposed deep cancer classifier is capable of diagnosing cancer in a wide range of human samples with almost 95% accuracy, which represents an improvement on conventional machine learning algorithms random forests and support vector machines. The model's performance is enhanced by exploiting two forms of contextual information, namely anatomical annotation of samples, and sequence annotation linking miRNAs to cistrons and sequence families. Once trained, the deep structure of the DCC can be interrogated for insights into the links between miRNAs and cancer. In particular, this enables the identification of miRNAs that may play serve as biomarkers or mediate the effects of cancers across diverse tissue types. The absolute activation gradient reveals a highly skewed distribution of feature importance, led by miR-21, a ubiquitous miRNA known to be disregulated in cancer.[23] This highly skewed feature importance distribution suggests the possibility of creating diagnostic arrays using small numbers of miRNAs. Gene ontology analysis of cancer-linked miRNAs identified multiple highly-enriched processes, some of which bare an obvious relationship to cancer (e.g. regulation of cellular proliferation) while others may indicate possible directions for future research.

References

1. I. Guyon, J. Weston, S. Barnhill and V. Vapnik, *Machine Learning* (2002).
2. Y. Bengio, *Foundations and Trends® in Machine Learning* (2009).
3. G. E. Hinton and R. R. Salakhutdinov, *Science* (2006).
4. M. Khademi and N. S. Nedialkov, *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)* , 727 (2015).
5. J. D. Young, C. Cai and X. Lu, *BMC Bioinformatics* **18** (2017).
6. A. F. Syafiandini, I. Wasito, S. Yazid, A. Fitriawan and M. Amien, *2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA)* , 108 (2016).
7. Y. Yuan, Y. Shi, C. Li, J. Kim, W. Cai, Z. Han and D. D. Feng, *BMC Bioinformatics* **17** (2016).
8. Y. Bengio, P. Lamblin, D. Popovici and H. Larochelle, Greedy Layer-Wise Training of Deep Networks, in *Proceedings of Neural Information Processing Systems, NIPS 2006*, (1)2006.
9. R. Fakoor, F. Ladhak, A. Nazi and M. Huber, *Proceeding of the 30th international conference on machine learning Atlanta, Georgia,USA* **28** (2013).
10. V. Singh, N. Baranwal, R. K. Sevakula, N. K. Verma and Y. Cui, *Proceedings - 2016 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2016* , 1542 (2016).
11. J. S. Mattick, *EMBO Reports* **2**, 986 (2001).
12. N. Rosenfeld, R. Aharonov, E. Meiri, S. Rosenwald, Y. Spector, M. Zepeniuk, H. Benjamin, N. Shabes, S. Tabak, A. Levy, D. Lebanony, Y. Goren, E. Silberschein, N. Targan, A. Ben-Ari, S. Gilad, N. Sion-Vardy, A. Tobar, M. Feinmesser, O. Kharenko, O. Nativ, D. Nass, M. Perelman, A. Yosepovich, B. Shalmon, S. Polak-Charcon, E. Fridman, A. Avniel, I. Bentwich, Z. Bentwich, D. Cohen, A. Chajut and I. Barshack, *Nature Biotechnology* **26**, 462 (2008).
13. R. Ibrahim, N. A. Yousri, M. A. Ismail and N. M. El-Makky, *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* , 3957 (2014).
14. M. Liang, Z. Li, T. Chen and J. Zeng, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **12**, 928 (2015).
15. K. Chaudhary, O. B. Poirion, L. Lu and L. X. Garmire, *Clinical Cancer Research* , p. clincanres.0853.2017 (2017).
16. P. Landgraf, M. Rusu, R. Sheridan, A. Sewer, N. Iovino, A. Aravin, S. Pfeffer, A. Rice, A. O. Kamphorst, M. Landthaler, C. Lin, N. D. Socci, L. Hermida, V. Fulci, S. Chiaretti, R. Foà, J. Schliwka, U. Fuchs, A. Novosel, R. U. Müller, B. Schermer, U. Bissels, J. Inman, Q. Phan, M. Chien, D. B. Weir, R. Choksi, G. De Vita, D. Frezzetti, H. I. Trompeter, V. Hornung, G. Teng, G. Hartmann, M. Palkovits, R. Di Lauro, P. Wernet, G. Macino, C. E. Rogler, J. W. Nagle, J. Ju, F. N. Papavasiliou, T. Benzing, P. Lichter, W. Tam, M. J. Brownstein, A. Bosio, A. Borkhardt, J. J. Russo, C. Sander, M. Zavolan and T. Tuschl, *Cell* **129**, 1401 (2007).
17. J. W. Tukey, *Exploratory Data Analysis* 1977.
18. S. Tam, M. S. Tsao and J. D. McPherson, *Briefings in Bioinformatics* **16**, 950 (2015).
19. C. Angermueller, T. Pärnamaa, L. Parts and O. Stegle, *Molecular Systems Biology* **12**, p. 878 (2016).
20. S. Azizi, F. Imani, S. Ghavidel, A. Tahmasebi, J. T. Kwak, S. Xu, B. Turkbey, P. Choyke, P. Pinto, B. Wood, P. Mousavi and P. Abolmaesumi, *International Journal of Computer Assisted Radiology and Surgery* **11**, 947 (2016).
21. L. Wang, D. S. Himmelstein, A. Santaniello, M. Parvin and S. E. Baranzini, *F1000Research* , 1 (2015).
22. S. Maere, K. Heymans and M. Kuiper, *Bioinformatics* **21**, 3448 (2005).
23. T. A. Farazi, J. I. Hoell, P. Morozov and T. Tuschl, *Advances in Experimental Medicine and Biology* (2013).

Implementing and evaluating a Gaussian mixture framework for identifying gene function from TnSeq data

Kevin Li
Department of Mathematics, Columbia University, New York, NY 10027, USA
Email: kl2918@columbia.edu

Rachel Chen
Department of Statistics, North Carolina State University, Raleigh, NC 27695, USA
Email: rschen@ncsu.edu

William Lindsey
Department of Mathematics and Statistics, Dordt College, Sioux Center, IA 51250, USA
Email: William.Lindsey@dordt.edu

Aaron Best
Department of Biology, Hope College, Holland, MI 49423, USA
Email: best@hope.edu

Matthew DeJongh
Department of Computer Science, Hope College, Holland, MI 49423, USA
Email: dejongh@hope.edu

Christopher Henry
Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL 60439, USA
Email: chrisshenry@gmail.com

Nathan Tintle
Department of Mathematics and Statistics, Dordt College, Sioux Center, IA 51250, USA
Email: Nathan.Tintle@dordt.edu

The rapid acceleration of microbial genome sequencing increases opportunities to understand bacterial gene function. Unfortunately, only a small proportion of genes have been studied. Recently, TnSeq has been proposed as a cost-effective, highly reliable approach to predict gene functions as a response to changes in a cell's fitness before-after genomic changes. However, major questions remain about how to best determine whether an observed quantitative change in fitness represents a meaningful change. To address the limitation, we develop a Gaussian mixture model framework for classifying gene function from TnSeq experiments. In order to implement the mixture model, we present the Expectation-Maximization algorithm and a hierarchical Bayesian model sampled using Stan's Hamiltonian Monte-Carlo sampler. We compare these implementations against the frequentist method used in current TnSeq literature. From simulations and real data produced by E.coli TnSeq experiments, we show that the Bayesian implementation of the Gaussian mixture framework provides the most consistent classification results.

Keywords: Bayesian; bacteria; genetics.

1. Introduction

1.1. *TnSeq Motivation and Background*

Understanding of bacterial gene function has not kept pace with the rapid acceleration of microbial genome sequencing. Only a small proportion of genes have had their functions experimentally examined and function estimates for unexamined genes have proven inaccurate.[1] Transposon mutagenesis with next generation Sequencing (TnSeq) is a recent method that alleviates this shortcoming in the study of gene function by allowing the simultaneous examination of a wide array of microbial genes.

In TnSeq, a transposon inserts itself into bacterial genes, creating mutants and potentially disrupting bacterial functions. In a library of mutants, DNA is isolated from a section of the bacterial pool as a control group. The remaining section can then be subjected to a test condition. Bacteria whose disrupted genes are essential for growth should decrease in frequency after exposure to the condition. PCR amplifies the DNA sequences bordering the insertions, which are then sequenced and map back to the genome. The change in a gene's fitness can be quantified by comparing the abundance of mutants before and after the test condition. Based on this change, we can then examine the effect of the disrupted genes in specific test conditions.[2] The test conditions under which the mutants suffer fitness penalties are then used to infer gene function.

1.2. *Motivation and New Methods*

The data produced by TnSeq poses classic statistical challenges. First, TnSeq allows researchers to produce fitness measurements for thousands of poorly understood genes across hundreds of experimental conditions.[3] This increase in scale from traditional experimental methods complicates attempts to create a universal decision rule for identifying a gene insertion's fitness condition. The inflated number of experiments also increases the frequency of outliers and edge cases. Furthermore, the magnitude of fitness change varies between gene insertions and experimental noise can be unpredictable. Current practice implements a frequentist statistical significance framework that does not incorporate assumptions inherent in TnSeq and ignores inter-gene information for classification. These shortcomings lead to overly conservative predictions due to overestimates of variance given the unique nature of TnSeq data. The frequentist framework also requires tuning to control the false-positive rate.[1] Finally, the current frequentist framework does not produce an easily interpretable uncertainty estimate for its classifications.

In this paper, we propose modeling the fitness measurements for gene insertions as two-component Gaussian mixture models. We use simulations to show that this framework increases sensitivity to fitness changes while controlling the false discovery rate at acceptable levels. We also provide two distinct methods for fitting these mixture models. The Expectation-Maximization algorithm is a widely accepted method for fitting such models. We also propose a hierarchical Bayesian approach in which we model the parameters of our Gaussian mixture as random variables with prior distributions. This strategy allows us to incorporate inter-gene information and prior knowledge of the TnSeq method as soft constraints on our estimates. We will ultimately compare the performance of these methods against the current frequentist framework.

2. Methods

2.1. *TnSeq Experimental Data*

We present a model of transposon sequencing in which only one strain of each gene insertion is counted. A control count is first obtained for each gene insertion by examining its growth under a condition known to have no effect on bacterial survival. Given n insertions, and m experimental conditions, TnSeq then produces an n x m matrix where each row represents an insertion, and each column contains fitness counts for an experimental condition. Thus if we denote this matrix $\boldsymbol{C}$, the matrix element $C_{i,j}$ represents the fitness counts for gene insertion i under experimental condition j. The final fitness measurement for each insertion under each experimental condition is calculated via the equation:

$$f = \log(n_1 + 1) \ - \log(n_0 + 1)^1 \tag{1}$$

where n_1 is the cell is count under the experimental condition and n_0 is the cell count under the control condition. The total variance of the gene's fitness value is calculated via:

$$V = \frac{\frac{1}{1+n_1} + \frac{1}{1+n_0}}{\ln(2)^2} \tag{2}$$

This variance assumes Poisson noise and is later used for calculating a t-like statistic for the frequentist method.[3]

2.2. *Mixture framework*

We apply our novel Gaussian mixture framework to the n x m matrix representing the fitness measurements of each insertion. We denote this matrix E. The matrix element $E_{i,j}$ represents the fitness measurement of the ith insertion under the jth experimental condition. We wish to identify each $E_{i,j}$ as the result of a neutral or deleterious experimental condition. Fitness measurements under deleterious experiments indicate that the mutant's disrupted gene is relevant to some function. Note that whether an experiment is neutral or deleterious depends on the mutant. To evaluate the likelihood of our label estimate, we propose modeling each row of E as a two-component Gaussian mixture. We would like the first mixture component to capture experiments in which fitness is unaffected such that $E_{i,j} | \ unaffected \sim \mathrm{N}(\mu_{i,0}, \sigma_i)$. The second component captures experiments in which fitness is affected such that $E_{i,j} | \ affected \sim \mathrm{N}(\mu_{i,1}, \sigma_i)$. Due to the nature of TnSeq data, we expect $\mu_{i,0}$ to be close to 0 and $\mu_{i,1}$ to be negative. This second component mixture exists because groups of experiments deliberately test similar bacterial functions and therefore produce similar fitness changes. This aspect of TnSeq also allows us to assume variances for the mixtures. We therefore define the likelihood of row i of the matrix as:

$$E_j \sim \theta\phi(\mu_{i,0}, \sigma_i) + (1 - \theta)\phi(\mu_{i,1}, \sigma_i) \tag{3}$$

where ϕ is the pdf of a normal distribution, and θ is the proportion of experiments in which the mutant is unaffected.

This framework can generate a probability that any fitness measurement is the product of a deleterious experiment. This probability that fitness measurement $E_{i,j}$ is produced by a deleterious experiment is defined as:

$$a_{i,j} = \frac{\phi(E_{i,j}|\mu_{i,1},\sigma_i)}{\phi(E_{i,j}|\mu_{i,1},\sigma_i) + \phi(E_{i,j}|\mu_{i,0},\sigma_i)} \quad (4)$$

This value is simply the density of the fitness-affected mixture divided by the total density. We classify the $E_{i,j}$ as the result of a deleterious experiment if $a_{i,j}$ is greater than .5.

2.3. *Classification methods*

2.3.1. *Novel method – EM*

An accepted statistical method for estimating unobserved labels under a Gaussian mixture likelihood is the Expectation-Maximization (EM) algorithm.[4] The EM algorithm iteratively fits a Gaussian mixture model by constructing a monotonically increasing sequence of lower bounds for the log likelihood function. We allow the mixture that is closest to zero represent the experiments that do not affect mutant fitness. The selection of a two-component mixture model as opposed to classifying all experiments as neutral is based upon the commonly used Bayesian Information Criterion (BIC).[4] We fit a two-component mixture model if it has the lower BIC compared to a simple Gaussian model. Otherwise we assume the insertion's fitness values are all produced from neutral experiments. We make this assumption as it is biologically improbable that all or even most experiments will harm fitness. We implement the algorithm through the R package Mclust.[5]

2.3.2. *Current method – t-statistic*

The current method in TnSeq literature leverages the estimated variance of fitness measurements to calculate the statistical significance of fitness changes.[1,3] It calculates a t-like statistic:

$$t = \frac{f}{\sqrt{.1+V}} \quad (5)$$

where .1 is a small regularizing constant, and V is the variance estimate for the insertion's fitness measurements as described in section 2.1. An experiment is considered deleterious if $|t| > 4$ and $|f| > .5$. This statistic is assumed to have a standard normal distribution[3].

The frequentist approach does not provide an easily interpretable probability for label estimates. For the sake of comparison, we define $a_{i,j}$ for the t-statistic classifier as:

$$a_{i,j} = 1 - \phi(\mathrm{t}) \quad (6)$$

where $\phi(\mathrm{t})$ represents a standard normal cdf. This expression is simply one minus the probability that we obtain a statistic as extreme as t under the assumption of no fitness change. This $a_{i,j}$ can be interpreted as the confidence of the classification.

2.3.3. *Bayesian hierarchical model*

We finally adopt a Bayesian hierarchical modeling framework for fitting a Gaussian mixture model. The hierarchical approach assumes that model estimates for individual insertions are conditional on some unobserved parameters shared across all insertions. We denote these parameters as hyper-parameters. The hyper-parameters have their own hyper-prior distributions which are estimated from all insertions in the data set. This strategy of conditioning estimates for individual genes on these sample-wide hyper-priors achieves a pseudo pooling effect. The hyper-prior distributions leverage across-gene information to weaken the influence of outliers and increase sensitivity to small mixture probabilities.[6]

We fit our hierarchical Bayesian model in the R interface to the probabilistic programming language, Stan.[7] Stan allows fast, out-of-the-box fitting of Bayesian models without the computation of the conditional parameter distributions or tuning variables.[8] We later provide strategies for partitioning our data set in order to speed computations and allow parallelization.

We use the following priors in our Bayesian model. We give $\mu_{i,0}$ prior distribution $N(0,\delta)$. The location of the prior is fixed at 0 to reflect the experiments' null effect on fitness. The scale of the prior is modeled by hyper-parameter δ with a $InverseGamma(20,1)$ prior. The parameters of the prior and hyper-prior reflect our strong belief that neutral experimental conditions should consistently produce fitness measurements close to zero plus or minus some error common to the mutants in the sample. The hierarchical structure on δ estimates this error from the mutants in sample. We default to the Inverse Gamma distribution for its conjugacy properties.

We constrain $\mu_{i,1}$ to be negative by the assumptions of transposon sequencing[3]. We give $\mu_{i,1}$ prior distribution $N(-3,\lambda)$. The mean of the prior is fixed at a negative real to prevent degenerate label switching with the first mixture. We choose –3 because it represents a moderate change in fitness.[3] The choice of –3 specifically as compared to any other reasonably small negative real is unimportant due to the choice of the uninformative scale prior λ, which has a prior distribution that is uniform across all positive real numbers. The uninformative prior allows λ to become arbitrarily large as the data demands.[6] The data dominates the value of λ in this the model and reflects our lack of prior information of the true distribution of the fitness measurements. We model λ as a hierarchical parameter to prevent outliers from overly affecting $\mu_{i,1}$ estimates and to increase sensitivity to departures from zero. Although $\lambda's$ prior is not a proper distribution, the joint distribution of $\mu_{i,1}$ and λ is proportional to an inverse gamma distribution, which ensures that the integral of the posterior distribution is finite.[6]

We give θ_i a beta prior with symmetric uniform hyper-priors for its flexibility over the $[0,1]$ interval as well as by the methods of Disselkoen 2016.[10] The hierarchical structure on theta resists outliers and prevents overfitting on single mutants.

We give σ_i a $Cauchy(0,5)$ prior. The prior is weakly informative by allowing for large values in the heavy tails of the distribution. This reflects our weak confidence that most variances should be reasonably small with a few exceptions. We select the Cauchy distribution by recommendation of Gelman 2006.[6]

2.3.4. *Data partitioning for the Bayesian model*

Markov Chain Monte Carlo sampling methods are computationally intensive for large data sets and sensitive to the true parameter diversity of the data. Therefore, we propose fitting the Bayesian model separately on partitions of the data that maximize within-partition similarity. Partitioning the data speeds sampling and makes the computations easily parallelizable. To maximize the similarity of genes within the partitions, we use the k-means clustering algorithm on the normalized log-fitness vectors of the genes. This clustering is equivalent to clustering the gene insertions by angular distance or correlation of their fitness measurement vectors.[11] For computational considerations in our simulation scenarios, we currently set the number of clusters such that there are on average 20 genes per partition.

2.4. *Simulation*

To evaluate the performance of our classifier, we simulate sets of insertions and fitness measurements under a fixed number of experiments. We simulate different scenarios where we vary the proportion of insertions that affect fitness under any experimental conditions. In this study we simulate cases where 0%, 25%, 50%, 75%, and 100% of insertions affect fitness. Simulating these distinct scenarios is important because the hierarchical Bayesian model estimates parameters of individual insertions from a parameter distribution estimated over the entire data set. For each scenario, we simulate 100 separate sets of 100 gene insertions to test the performance of the three methods. We note that the Bayesian model is fit separately on each of these sets of 100.

We adopt the following algorithm for simulating bacterial counts and fitness measurements. First, across all gene insertions in a set we define a probability δ that a gene insertion affects fitness under any experimental conditions. We then proceed through the following steps to draw the mutant counts.

For each gene insertion i:

- Draw parameter τ from gamma distribution $gamma(\hat{\alpha}, \hat{\beta})$, in which $\hat{\alpha}$ and $\hat{\beta}$ are the gamma parameter maximum likelihood estimates from the experimental control counts of E.coli mutants provided by Price 2018.[1] This distribution is not significantly different from the empirical control count distribution by the Kolmogorov-Smirnov test ($p > .3$).
- Draw the simulated control count from $poisson(\tau)$. Denote $poisson(\tau)$ as the neutral distribution.
- Choose a fitness factor, F from $uniform(.15, .95)$. We denote $poisson(\tau * F)$ as the affected distribution.
- With probability δ, draw θ from $uniform(.3, .95)$. Else set θ to be 1. θ is the probability that an experiment does not affect mutant fitness.
- For every experiment, draw a count from the control distribution with probability θ. Otherwise draw a count from the deleterious distribution.

Pre-fixed simulation distribution parameters were chosen to account for all reasonable biological possibilities. Uniform distributions were chosen by the maximum entropy principle to reflect our uncertainty surrounding the true distribution of real data sets.[12] The fitness measurements and t-

statistics for each experiment can be calculated for each gene insertion using the control count and Eq. (5) and (6).

2.5. *Real data*

We apply our methods to Escherichia coli BW25113 TnSeq data provided by Price 2018.[1] They examine the fitness of E.coli mutants produced by 3789 distinct gene insertions. They subjected mutants to 162 experimental conditions. We apply the EM and Bayesian classifiers to the provided 3789 x 162 matrix of fitness measurements. We use the t-statistic classification results provided by Price 2018.

3. Results

We evaluate the following performance metrics for each of the classification methods. We use the mean of the posterior distribution draws of the Gaussian mixture parameters to define the Bayesian model.[6] We use the following metrics to evaluate the performance of the classifiers.

3.1. *Metrics*

Define the true label for fitness measurement $E_{i,j}$ as $l_{i,j}$, taking value 0 if $E_{i,j}$ is the result of a neutral experiment and value 1 if $E_{i,j}$ is the result of a deleterious experiment. Let the predicted label for $E_{i,j}$ be $\widehat{l_{i,j}}$. Similarly $\widehat{l_{i,j}}$ is 0 if the classifier labels the fitness measurement as a neutral result and 1 if the classifier labels the measurements as a deleterious result.

3.1.1. *Classification rate*

The classification rate is the raw percentage of experiments that the model classifies correctly. Therefore the Classification Rate for the ith insertion would be:

$$CR_i = \frac{\sum_{j=1}^{m} I_{\{l_{i,j} = \widehat{l_{i,j}}\}}}{m} \tag{7}$$

where $I_{\{l_{i,j} = \widehat{l_{i,j}}\}}$ is an indicator function that takes value one if $l_{i,j} = \widehat{l_{i,j}}$ and zero otherwise.

3.1.2. *False positive rate*

The false positive rate is the Type I error. It is the percentage of neutral experiments that the model incorrectly classifies as deleterious. In ideal scenarios, this value should be low. The False Positive Rate for the ith mutant is therefore:

$$FP_i = \frac{\sum_{j=1}^{m} I_{\{\widehat{l_{i,j}} = 1 \ \wedge \ l_{i,j} = 0\}}}{\sum_{j=1}^{m} I_{\{l_{i,j} = 0\}}} \tag{8}$$

where $I_{\{\widehat{l_{i,j}} = 1 \wedge l_{i,j} = 0\}}$ is an indicator function that takes value one if $\widehat{l_{i,j}} = 1$ and $l_{i,j} = 0$.

3.1.3. *Positive classification rate*

The positive classification rate is the percentage of deleterious experiments that the model correctly classifies as deleterious. In ideal scenarios, this value should be high. The positive classification rate for the ith insertion is therefore:

$$PC_i = \frac{\sum_{j=1}^{m} I_{\{\widehat{l_{i,j}}=1 \ \wedge \ l_{i,j}=1\}}}{\sum_{j=1}^{m} I_{\{l_{i,j}=1\}}} \tag{9}$$

where $I_{\{\widehat{l_{i,j}}=1 \ \wedge \ l_{i,j}=1\}}$ is an indicator function that takes value one if $\widehat{l_{i,j}} = 1$ and $l_{i,j} = 1$. Otherwise the function takes value 0.

3.1.4. *Cross entropy*

We measure the accuracy of our probabilistic estimates using cross-entropy. The cross entropy for the classification of the ith insertion is defined as:

$$CE_i = -\sum_{i=1}^{m} \widehat{l_{i,j}} \log(a_{i,j}) + (1 - \widehat{l_{i,j}}) \log(1 - a_{i,j}) \tag{10}$$

Cross entropy is a common loss function for evaluating classifiers that produce probability estimates ranging from 0 to 1.[4] The greater the difference between the true and model classifications, the higher the cross entropy will be. For example, if the true label is 1 and $a_{i,j}$ is 0, then the classifier performs badly and the cross entropy will be high. However, a better probability estimate of .49 will correspond to a lower cross entropy value.

3.2. *Simulation Results*

We simulate the scenarios in which 0%, 25%, 50%, 75%, and 100% of gene insertions are affected by experimental conditions. For each scenario, we simulate one hundred sets of one hundred insertions. On each set, we separately fit the Bayesian model on a single Markov chain with 1000 warm-up iterations and 1000 sampling iterations. We take the posterior means of the Gaussian mixture parameters to define our Bayesian classification model.

The simulation results demonstrate that the three methods provide identical classifications for 64% of the 50,000 simulated genes. These classifications produced models with over a 98% classification rate. This is expected as the simulated fitness values for many gene insertions are either obviously unimodal or clearly clustered into two groups. In an additional 10% of cases, all the classifiers achieved at least a 90% classification rate. Thus, the entire simulation population does not tell us much about the relative performance of the classifiers on difficult classification problems.

We proceed to examine only the 26% of the cases where the t-statistic, EM algorithm, and Bayesian classifier do not provide identical classifications and at least one of the classifiers fails to achieve an 90% classification rate. We call this the difficult subset.

We see in Figure 1 and Table 1, Column 3 that the t-statistic performs relatively well when the proportion of affected mutants is small (0%, 25%). For higher proportions, we see that the t-

statistic's performance deteriorates in the second and third classification quantiles relative to the other methods. On the other hand, the EM algorithm performs well when the proportion of affected mutants is large (75%, 100%). The EM algorithm suffers in performance for the first and second and third quantiles, especially for lower proportion (0%, 25%, 50%). Only the Bayesian model demonstrates consistent behavior across proportions and quantiles, outperforming both the other methods except when the proportion of affected mutants is 0%.

We see from the positive classification rate in Figure 1 and Column 4 in Table 1 that the t-statistic is by far the least sensitive to changes in fitness and therefore has the lowest positive classification rate. The Bayesian algorithm provides a vast improvement on the positive classification rate. But the EM algorithm overall provides the most sensitive classification results, especially true at lower proportions. The EM algorithm achieves this sensitivity by incurring higher false positive rates. The Bayesian algorithm does not suffer from as high false positive rates. The t-statistic expectedly maintains the lowest false positive rate. Therefore, we see that the Bayesian algorithm achieves higher and consistent classification by compromising between sensitivity of the EM algorithm and the conservatism of the t-statistic.

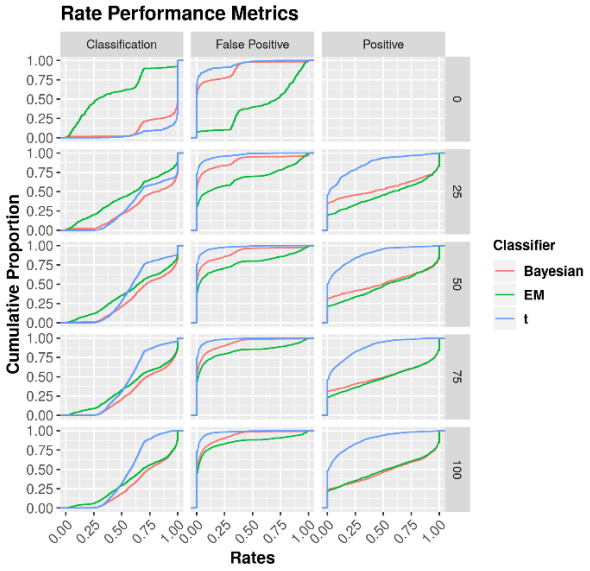


Fig. 1. Cumulative distributions of classification, false positive and positive classification rates on the difficult subset of simulated gene insertions. Columns indicate the metric displayed, and rows indicate the proportion of mutants affected in each mutant set.

Table 1. Mean Classification Rate, Positive Classification Rate, False Positive Rate and Cross Entropy for Classifiers

	2. % Affected	3. Mean CR	4. Mean PCR	5. Mean FPR	6. Mean CE
Bayesian	0	.90	NA	.08	65.13
	25	.75	.57	.07	242.27
	50	.72	.60	.06	279.49
	75	.73	.61	.05	301.30
	100	.73	.64	.05	288.97
EM	0	.40	NA	.33	305.95
	25	.58	.65	.20	313.95
	50	.63	.64	.14	320.05

	75	.66	.63	.10	334.89
	100	.68	.63	.09	334.79
T	0	.95	NA	.05	171.68
	25	.72	.22	.03	267.99
	50	.62	.21	.02	308.60
	75	.59	.22	.02	339.06
	100	.57	.21	.02	343.20

From Figure 2 and Column 6 in Table 1, we see that the Bayesian and EM method produce smaller cross entropy losses for most classifications compared to the t-statistic. However, we also see that the Bayesian and EM methods have fatter tails, indicating a significant subset of cases where the two methods provide poor probability estimates. From Table 1 Column 6, we see that from an entropy standpoint, the Bayesian algorithm outperforms the EM algorithm and t-statistic on average in every scenario. Therefore, we can see that the Bayesian algorithm provides accurate probabilistic estimates more consistently.

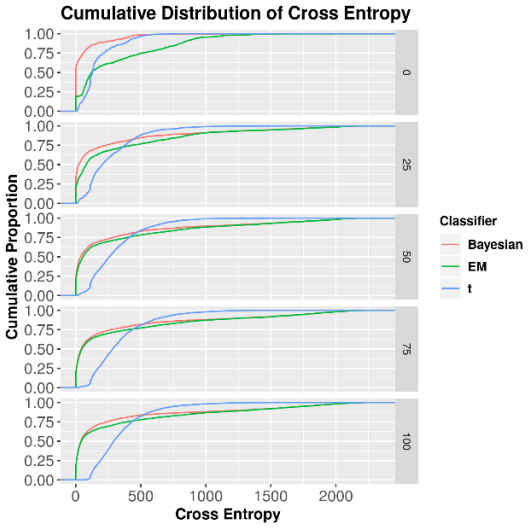


Fig. 2. Cumulative distribution of Cross Entropy Distributions. Cross entropy values near zero indicate accurate probability estimates of classification confidence.

3.3. *Comparisons on real data*

We apply the EM and Bayesian methods to the fitness measurements from the real E.coli data (see section 2.5 for details). For the t-statistic, we use the classifications produced by the work of Price 2018[1]. The t-statistic is by far the most conservative, identifying 496 genes as important to some examined bacterial function. The EM algorithm identifies 1322 genes and the Bayesian method identifies 1786 genes. Of the 496 genes identified by the t-statistic, the EM algorithm shares 137 identifications. The Bayesian algorithm shares 455 gene identifications with the t-statistic. In Figure 3 we present three examples where each of the three classifiers fails to identify a gene's function where the other two are successful.

The mutant from the insertion into gene b0002 is an instance where the t-statistic does not identify a gene where the Bayesian model and EM algorithm do. The EM algorithm and Bayesian

model provide the same classifications for b0002, while we see that the t-statistic fails to identify any changes in fitness. This failure of the t-statistic behavior can be attributed to the clear existence of two separate mixture components with separate variances. The t-statistic calculates the variance from both mixtures and therefore underestimate significance.

We next give an example where the EM algorithm does not identify a gene (b0008) that the t-statistic and Bayesian model identify. In this case in Figure 3, we see that the BIC does not detect the presence of two mixtures and our implementation of the EM algorithm and therefore assumes no changes in fitness. We have considered changing the BIC threshold for two-mixture selection, but any changes resulted in much worse simulation results.

Now we examine the insertion on b1198. This insertion belongs to the 16 cases where the Bayesian algorithm does not identify a gene that the EM algorithm and t-statistic both identify as important to some function. In each of these cases the EM algorithm and t-statistic identify a positive fitness change from a gene insertion. This is improbable, as a gene deletion should not increase fitness. The Bayesian model's priors explicitly prevent this classification result.

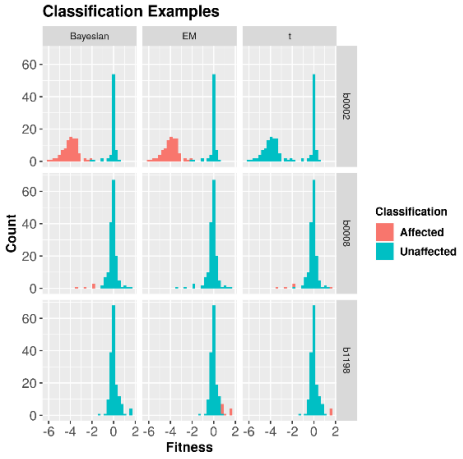


Fig. 3. Classifications for mutants produced by insertions into genes b0002, b0008, and b1198. Bars represent counts of fitness measures under various experimental conditions.

3.4. *Software*

R scripts for the implementation of the classification methods can be found at: http://www.nathantintle.com/supplemental/TnSeqRFunctions.R

4. Discussion

We have presented a two-component Gaussian mixture framework for classifying experimental effects on mutant fitness. This framework provides an alternative to the current frequentist framework. We have shown how the frequentist approach produces conservative estimates due to its estimation of a large variance encompassing all of mutant's fitness values despite the existence of two smaller distributions. The mixture framework addresses this problem by estimating the smaller variances of two smaller components.

Furthermore, simulations demonstrate that the Bayesian classifier generally outperforms the EM algorithm. By incorporating reasonable priors and exploiting a hierarchical structure, the Bayesian

model leverages inter-gene information to provide a compromise between the sensitivity of the EM algorithm and the conservatism of the t-statistic. The Bayesian model's performance is also nearly invariant under the proportion of mutants affected. Given high uncertainty about the genes studied, the Bayesian model should be the model of choice for classification.

On the real E.coli data, we see that the Bayesian classifier is able to identify all the genes with negative fitness changes that the t-statistic identifies. The Bayesian classifier demonstrates significantly more sensitivity to fitness changes while maintaining consistency with the t-statistic. This behavior is distinct from the EM algorithm, which has significantly different identifications and seems to be insensitive to lower mixing probabilities. Still, both mixture classifiers are able to identify multi-functional genes at a much higher rate than the t-statistic.

Despite the promise of the methods proposed, further work is necessary to validate our approach on additional datasets for which true fitness changes are known. We note that while the performance of the Bayesian classifier is generally better than the EM algorithm, the computational time of the Bayesian classifier may be prohibitive in some cases (e.g., it takes 30.8 hours with 5 cores to fit the E.coli 3789 x 162 fitness measurement matrix). Further work will seek to enhance the computational time of the Bayesian classifier, though we acknowledge that it may never be as 'instantaneous' as the EM algorithm or t-statistic approaches.

The success of the Bayesian classifier encourages further expansion of the hierarchical model structure. Hyper-prior distributions can be defined to account for multiple strains per mutant or even genes across bacteria. Covariance priors can be added to leverage co-fitness information[1] to make more robust classifications. Further development of the hierarchical structure will allow rich probabilistic models of gene function and fitness. In the meantime, we suggest use of the proposed Bayesian classifier to improve classification accuracy of changes in mutant fitness.

Acknowledgments The authors of this project were partially supported by NSF-MCB-1715211.

References

1. M. N. Price *et al.*, *Nat.* **557**, 503—509 (2018).
2. T. van Opijnen, K. L. Bodi and A. Camilli, *Nat. Methods* **6**, 767—772 (2009).
3. K. M. Wetmore *et al.*, *mBio* **6**, e00306-15 (2015).
4. T. Hastie, R. Tibshirani and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction* (2nd ed.). New York, NY: Springer (2009).
5. L. Scrucca, M. Fop, T. B. Murphy and A. E. Raftery, *The R Journal* **8**, 205—223 (2016).
6. A. Gelman, J. B. Carlin, H. S. Stern and D. B. Rubin, *Bayesian data analysis* (3rd ed.). Taylor & Francis (2013).
7. Stan Development Team, *RStan: the R interface to Stan* **2.16.2** (2017).
8. B. Carpenter *et al.*, *Journal of Statistical Software* **76**, (2017).
9. A. Gelman, *Bayesian Anal.* **1**, 515—534 (2006).
10. C. Disselkoen *et al.*, *Front. Microbiol.* **7**, 1191 (2016).
11. S. Zhong, *International Joint Conference on Neural Networks* **5**, 3180—3185 (2005).
12. W. Boomsma, J. Ferkinghoff-Borg and K. Lindorff-Larsen, *PLoS Comput. Biol.* **10**, e1003406 (2014).

SNPs2ChIP: Latent Factors of ChIP-seq to infer functions of non-coding SNPs

Shankara Anand*, Laurynas Kalesinskas*, Craig Smail*, and Yosuke Tanigawa*†
Department of Biomedical Data Science, Stanford University, Stanford, CA 94305, U.S.A.
**These authors contributed equally to this work.*
†*E-mail: ytanigaw@stanford.edu*

Genetic variations of the human genome are linked to many disease phenotypes. While whole-genome sequencing and genome-wide association studies (GWAS) have uncovered a number of genotype-phenotype associations, their functional interpretation remains challenging given most single nucleotide polymorphisms (SNPs) fall into the non-coding region of the genome. Advances in chromatin immunoprecipitation sequencing (ChIP-seq) have made large-scale repositories of epigenetic data available, allowing investigation of coordinated mechanisms of epigenetic markers and transcriptional regulation and their influence on biological functions. To address this, we propose SNPs2ChIP, a method to infer biological functions of non-coding variants through unsupervised statistical learning methods applied to publicly-available epigenetic datasets. We systematically characterized latent factors by applying singular value decomposition to 652 ChIP-seq tracks of lymphoblastoid cell lines, and annotated the biological function of each latent factor using the genomic region enrichment analysis tool. Using these annotated latent factors as reference, we developed SNPs2ChIP, a pipeline that takes genomic region(s) as an input, identifies the relevant latent factors with quantitative scores, and returns them along with their inferred functions. As a case study, we focused on systemic lupus erythematosus and demonstrated our method's ability to infer relevant biological functions. We systematically applied SNPs2ChIP on publicly available datasets, including known GWAS associations from the GWAS catalogue and ChIP-seq peaks from a previously published study. Our approach to leverage latent patterns across genome-wide epigenetic datasets to infer the biological functions will advance understanding of the genetics of human diseases by accelerating the interpretation of non-coding genomes.

Keywords: non-coding genome; functional interpretation; epigenome; latent factor discovery; biomedical ontology; enrichment analysis; large-scale inference; data integration

1. Introduction

Genome-wide association studies (GWAS) have successfully identified many associations between genetic variants and human diseases.[1,2] However, functional interpretation of these associations remains challenging as most GWAS hits fall into non-coding regions of the genome.[3] Advancements in high-throughput genome-wide molecular profiling methods, such as ChIP-seq, enable molecular characterization of gene regulatory landscapes, such as histone modification and transcription factor (TF) binding profiles.[4] Leveraging growing biomedical ontologies, such as the gene ontology (GO), human phenotype ontology (HPO), and Mouse Genome Infor-

matics (MGI) phenotype ontology, tools based on statistical enrichment analysis on genomic regions, such as the genomic region enrichment analysis tool (GREAT), have been used to investigate the function of the non-coding genome.[5–9] Further, collaborative research efforts, such as ENCODE, the Roadmap Epigenomics project, and Genotype-Tissue Expression program (GTEx), have also systematically generated data-rich molecular catalogues.[10–12] These large-scale epigenomic profiles, as well as other publicly available datasets on the NCBI sequence read archive, are integrated into epigenetic data resources, such as ChIP-Atlas and ReMap, which provides an emerging opportunity for data mining and meta-analysis.[13,14]

Advancements in epigenetic analysis suggest that latent patterns in epigenomic regulatory profiles can be discovered and characterized for downstream analyses. For example, one TF can bind to numerous genomic loci with specific sequence features and multiple TFs can work together by forming dimers, executing coordinated transcriptional regulatory programs.[15] Moreover, it is known that many TFs have multiple functions through precise coordination in different contexts, that there are known interactions between histone modifications and TF occupancy, and that histone modifications and TF occupancy influence gene expression.[10–12,15] With these phenomena in mind, there has been works in harnessing these patterns for functional interpretation of non-coding genomes. ChromHMM and Segway, unsupervised statistical learning methods, successfully summarizes patterns of epigenetic profiles as interpretable annotations,[16,17] while eQTL studies examines non-coding variants in light of molecular phenotype, such as expression levels of neighboring genes.[12] While these approaches show some success in utilizing neighboring epigenomic signals to explore molecular interpretation of non-coding genomes, they are limited in leveraging genome-wide patterns of both histone modification and TF occupancy across different functional contexts. In principle, one can extend these analyses by leveraging all experimentally collected epigenomic profiles and characterizing latent patterns for functional interpretation of non-coding genomic regions on a genome-wide scale.

Here we present SNPs2ChIP, a novel method to infer function of non-coding variants by (1) characterizing latent patterns in epigenomic regulatory profiles using an unsupervised latent factor discovery algorithm applied to 652 ChIP-seq tracks in the ChIP-Atlas dataset, (2) inferring the biological functions of the identified latent factors using GREAT enrichment analysis, and (3) development of a pipeline that takes genomic loci as input and infers functionality of the loci by identifying relevant latent factors using a quantitative score. Our computational approach contributes to dissecting the genetic architecture of human diseases by accelerating functional interpretation of non-coding variants.

2. Results

2.1. SNPs2ChIP *analysis framework overview*

We developed a method, SNPs2ChIP, to infer functions of non-coding loci that consists of two computational steps: (A) construction of reference ChIP-maps and (B) using the reference ChIP-maps to infer biological functions for user queries. To briefly summarize the first part of our method, we collected chromatin-profiling data from ChIP-Atlas, one of the largest publicly-available databases of ChIP-seq signals with manually curated metadata,[13] and featurized the

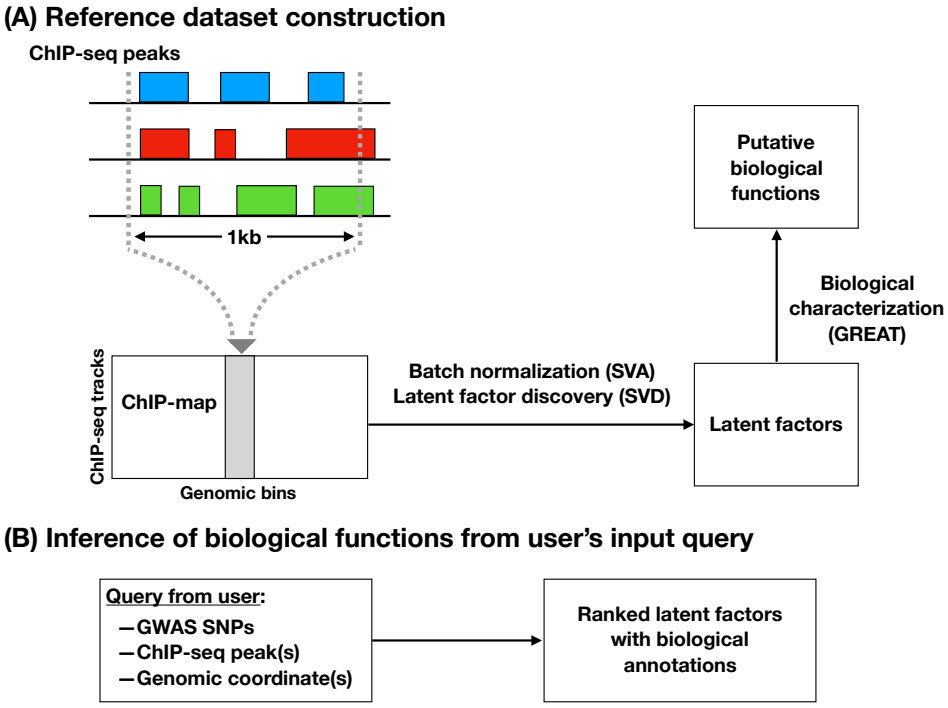


Fig. 1. SNPs2ChIP method overview. (A) Construction of SNPs2ChIP reference dataset. ChIP-seq peaks of 652 assays are aggregated into a feature matrix, ChIP-map, followed by batch normalization with surrogate variable analysis (SVA). Latent factors are characterized with singular value decomposition (SVD) and their biological functions are inferred with the genomic region enrichment analysis tool (GREAT). (B) SNPs2ChIP pipeline. Using the pre-computed reference, SNPs2ChIP identifies the most relevant latent factors and returns them with their annotated biological functions.

ChIP-seq peaks across TFs and histone marks into a matrix, called a "ChIP-map." To balance the trade-off in specificity of the functional prediction and the genomic coverage of the ChIP-map, we prepared two matrices for high-specificity and high-coverage analysis, by varying the stringency of the featurization methods. After featurization, we applied batch normalization with surrogate variable analysis (SVA) and singular value decomposition (SVD) in each map, resulting latent factors preserving a linear structure optimal for interpretation..[18] This was followed by applying GREAT to find the biological functions enriched in each latent factor (Fig. 1A).[9] With latent factors and enriched functions as pre-computed reference, we developed a pipeline that takes a loci as input and returns a list of relevant latent factors as well as their enriched function. A query can be one or multiple genomic loci: GWAS SNPs, ChIP-seq peaks, or genomic coordinates of interest (Fig. 1B).

2.2. *Batch normalization of heterogeneous epigenetic features*

We focused on 652 lymphoblastoid cell line experiments, the most numerous cell line in the ChIP-Atlas database, and downloaded all non-empty ChIP-seq peak files. We divided the entire genome into genomic bins of 1 kbp in size and placed ChIP-seq peaks, represented by the strength of the peak, into the bins. This was done across 652 tracks, which created a

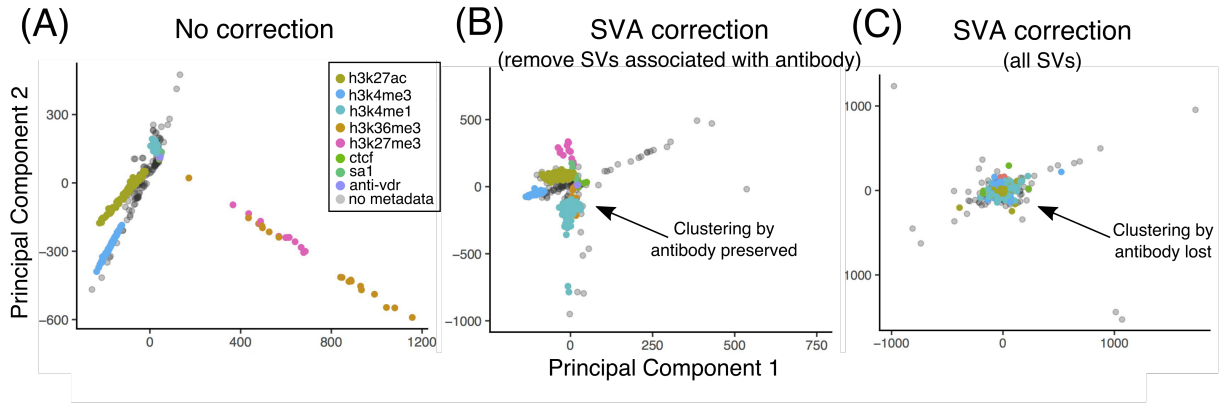


Fig. 2. Batch normalization of ChIP-map with SVA.

ChIP-map matrix. After removing genomic bins that did not contain any peaks, we found 379,541 (covering 12.1 % of genome) genomic bins and 662,024 (21.1 %) genomic bins for the high-specificity dataset and high-coverage dataset, respectively (Methods).

To normalize batch effects in each ChIP-map, we applied the SVA algorithm, a normalization method useful when technical covariates are not known or have missing entries.[18] Out of 39 significant surrogate variables (SVs) identified from SVA, we found that three SVs were significantly associated (p-value $< 1.0 \times 10^{-30}$, linear regression) with antibody - a biological effect necessary to protect. The first SV captured variation attributed mainly to H3K4me1 and H3K4me3; the second SV captured variation for H3K27ac and H3K4me3; and the third SV captured variation for CTCF, H3K4me3 and SA1. Note that the variation from one sub-group of a given covariate can be split across multiple SVs, as is the case with H3K4me3.

We assessed the effect of the removing these SVs when regressing out SVs from ChIP-map and compared with results of keeping all SVs in the regression. We implemented the regression using a QR decomposition, enabling an efficient, high-dimensional multivariate multiple regression. When removing SVs significantly associated with antibody, clear clusters were preserved in the corrected data reflective of antibody, but not for technical effects such as ancestry (Fig. 2A-B). Conversely, when we including all SVs in the regression, no clusters were observed for antibody, indicating an over-correction of data, i.e. removal of biological signal of interest (Fig. 2C). Therefore, using a combination of SVA, linear regression and clustering, we were able to preserve biologically important variation while removing unwanted technical variation.

2.3. *Latent factor discovery and their biological characterization*

To find interpretable latent factors in an unbiased manner, we applied an unsupervised statistical learning algorithm, SVD, to the batch normalized ChIP-map. Using the high-specificity dataset, we found that the first three latent factors explain 8.2 %, 6.0 %, and 4.6 % of the variance, respectively, and that the top 50 and 100 factors comprehensively explain 59 % and 72.5 % of the variance, respectively. For the high-coverage dataset, we found the first three latent factors explain 14.0 %, 10.7 %, and 5.7 % of the variance, respectively, and that the top

50 and 100 factors comprehensively explain 72.6 % and 82.6 % of the variance, respectively.

To characterize the biological functions of each latent factor, we identified the top 5,000 genomic bins ranked using the genomic bin contribution score derived from decomposed matrices by SVD (Methods - Eq. (1)). We applied GREAT enrichment analysis for the top genomic bins in each latent factor and identified enriched functional terms using three ontologies: GO, HPO, and MGI phenotype ontology.[5–9]

2.4. SNPs2ChIP *identifies relevant functions of the non-coding genome*

To illustrate the utility of SNPs2ChIP to infer the function of non-coding genome, we applied the pipeline to known GWAS SNPs and ChIP-seq peaks from previously published datasets.

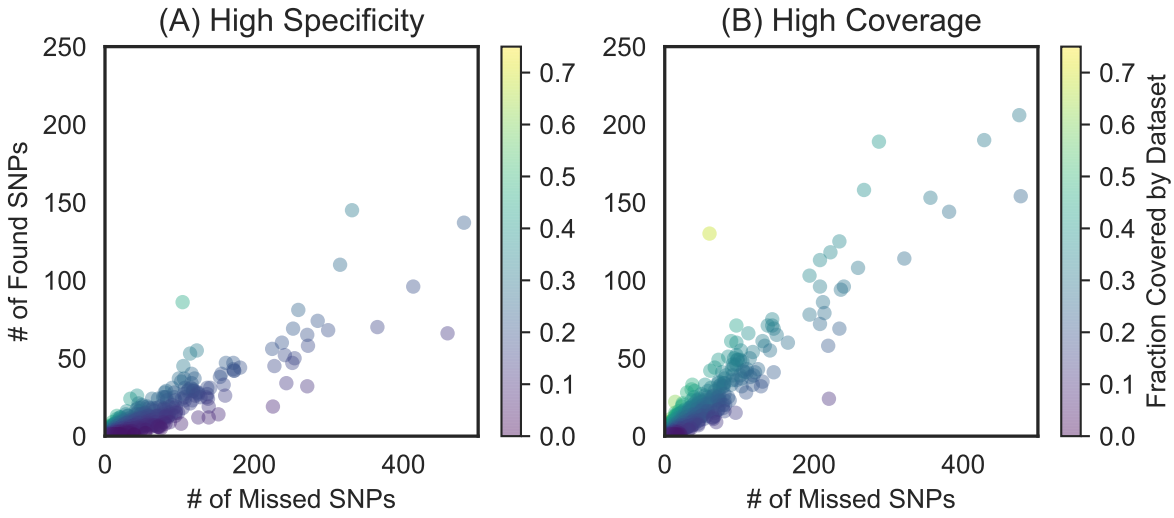


Fig. 3. Genome-wide coverage of the two reference datasets of SNPs2ChIP. For each phenotype in the GWAS catalog, we queried SNPs2ChIP and summarized what percentage of the SNPs can be mapped to the latent factors for the (A) high specificity dataset and (B) high coverage dataset.

2.4.1. *Genome-wide SNPs coverage of the reference datasets*

Given that our reference datasets do not contain empty genomic bins, thus excluding parts of the genome, we first evaluated the coverage of our reference dataset by applying the SNPs2ChIP pipeline to all previously reported SNPs from the GWAS catalogue.[1] We applied the pipeline for each disease/trait and summarized the number and percentage of SNPs covered by our reference datasets. Out of the 51,892 known non-intergenic GWAS SNPs we tested, we found our high-specificity and high-coverage datasets covers 9,241 (17.8 %) and 14,636 (28.2 %) of SNPs (Fig. 3).

2.4.2. *Non-coding GWAS SNPs of systemic lupus erythematosus*

To illustrate the utility of our approach to infer biological functions associated with non-coding GWAS SNPs of diseases, we performed a case-study on systemic lupus erythematosus

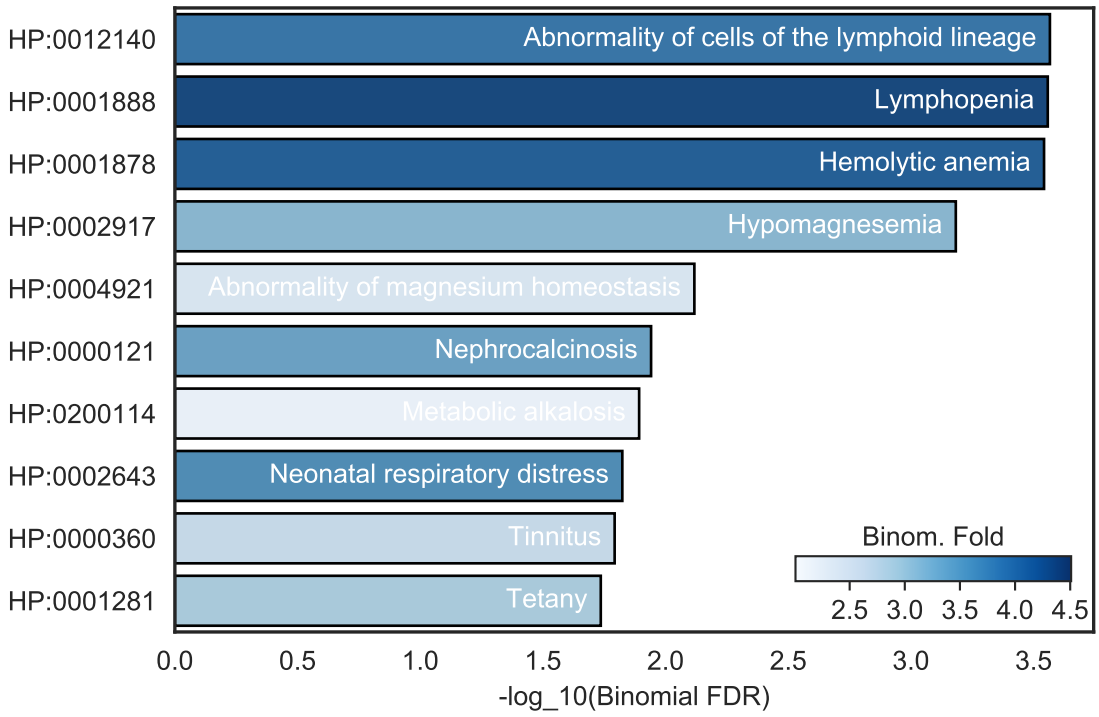


Fig. 4. SNPs2ChIP identifies the relevant biological functions given GWAS hits for systemic lupus erythematosus. GREAT binomial FDR and binomial fold for HPO ontology are shown.

(SLE). SLE is an autoimmune disorder with a prevalence of 0.1 % and a poorly characterized genetic and epigenetic basis.[19] Out of 425 GWAS SNPs associated with SLE, 110 and 158 SNPs are covered in the high-specificity and high-coverage reference dataset, respectively. Applying the pipeline to the SNPs covered by high-specificity dataset, the top latent factor identified explained 10.7 % of the variance in the epigenetic landscape and was enriched for multiple biological concepts associated with SLE. Using HPO as the reference ontology, we found human phenotypes, such as "Abnormality of cells of the lymphoid lineage" (HP:0012140, binomial FDR $= 2.7\times10^{-4}$), "Lymphopenia" (HP:0001888, FDR $= 2.8\times10^{-4}$), and "Hemolytic anemia" (HP:0001878, FDR $= 2.9\times10^{-4}$), which are all known phenotypes for SLE (Fig. 4).[20,21]

2.4.3. *ChIP-seq peaks for vitamin D receptors*

To further test the applicability of SNPs2ChIP, we applied the pipeline to ChIP-seq peaks assosciated with vitamin D receptors (VDR) as an example. Vitamin D is known to participate in transcriptional regulation through VDRs and regulates calcium homeostatic functions.[22] Its deficiency has been implied in multiple phenotypes, including increased risk of fracture, muscle weakness, and skeletal mineralization defect.[23]

Using the ChIP-seq peaks highlighted in a previously published study,[24] we applied SNPs2ChIP and identified relevant phenotypes, such as "Parietal foramina" (HP:0002697, FDR $= 1.3\times10^{-4}$) and "Flat forehead" (HP:0004425, FDR $= 2.3\times10^{-3}$).

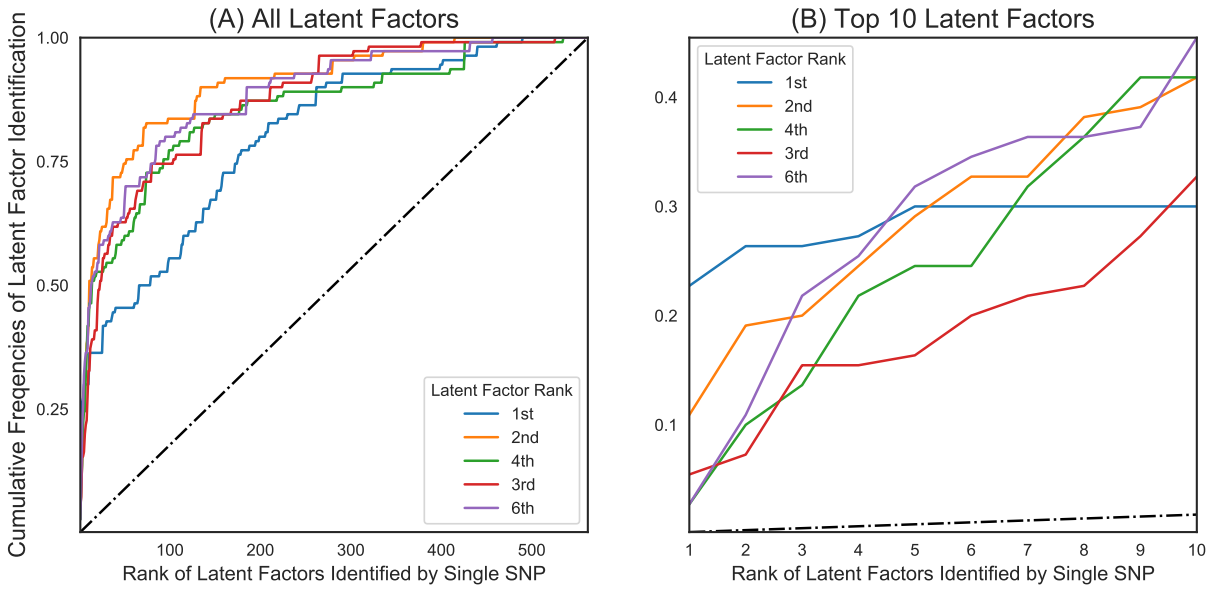


Fig. 5. Robustness analysis in the latent factor identification. By using all SNPs associated with SLE, we found the top 5 relevant latent factors (Methods, Eq. (4)). Iterating through each SNP, we plot the cumulative frequency of identifying each of the top 5 latent factors within the rank specified for (A) all latent factors and (B) top 10 ranks. The dashed black line indicates the cumulative frequency under the random null model.

2.5. *Robustness Analysis in the latent factor identification*

In the SNPs2ChIP pipeline, the identification of the relevant latent factor given a user query is a critical step. To assess the robustness, we applied the pipeline on all of the SLE associated SNPs with the high-specificity dataset and found the top 5 latent factors enriched across the group (Methods, Eq. (4)). We then applied our pipeline on each SNP independently and identified the relevant latent factors for each single SNP (Methods, Eq. (2)). We recorded the number of SNPs that successfully mapped to each of the top 5 latent factors within the top n ranks and reported the results as a cumulative distribution (Fig. 5).

3. Discussion

In this study, we propose a new method, SNPs2ChIP, to infer the function of genomic loci in the non-coding genome by leveraging latent patterns in publicly available ChIP-seq data tracks. Using latent factors characterized by SVD and annotating them with biomedical ontologies, we developed a pipeline that allows us to take genomic regions as input and return relevant latent factors with their enriched biological functions. We applied our method to GWAS SNPs and found that SNPs2ChIP can identify relevant biological functions associated with disease, demonstrating the utility of the genome-wide epigenomic latent factors in interpretation of non-coding SNPs. In addition, we demonstrated the applicability of our method for vitamin

D receptor ChIP-seq peaks, illustrating the utility of our approach for a diverse set of queries.

Further, as shown in our robustness analysis, SNPs2ChIP has an ability to identify relevant latent factors and functions even from a single SNP. This is a major advantage of SNPs2ChIP: it requires a minimal amount of input, one genomic coordinate, to infer biological function as it leverages latent patterns in the epigenome from across the whole genome.

As we rely on existing ChIP-seq data and we focused on lymphoblastoid cell lines, our reference dataset has limited coverage of the genome, which is 12.1 % and 21.1 % for our high-specificity and high-coverage datasets, respectively. While they still provide a GWAS set coverage of 17.8 % and 28.2 %, a further expansion of the reference dataset may expand the applicability of the methods.

The resources made available with this study, including the SNPs2ChIP pipeline as well the processed datasets, can provide a starting point to infer the biological functions of non-coding genomes. Combined with the expansion of large-scale epigenomic datasets,[13,14] our results highlight the utility of latent factor analysis in interpreting the non-coding genome.

4. Methods

4.1. *Featurization of the heterogeneous epigenetic assays*

From the ChIP-Atlas database, we downloaded all available ChIP-seq peak files with FDR corrected q-value threshold of 1.0×10^{-5} for lymphoblastoid cell lines.[13] Out of the 682 BED files we obtained from the database, we found that 652 were non-empty and used these for our analysis. To featurize the data, we defined genomic bins of size 1kbp across all autosomes and saved them as a custom, genomic bin BED file. For the high-specificity dataset, we kept the top 25,000 statistically significant peaks for each of the 652 BED files, to minimize the confounders due to experimental design, and intersected each of them with the genomic bin BED file using BEDTools.[25] For the high-coverage dataset, we used all of the peaks in the BED files and intersected these with the genomic bin BED file. For each pair of genomic bin and ChIP-seq assay from the BED intersection, we aggregated the negative log q-values into a matrix and removed the genomic bins with no peaks. We generated two ChIP-maps, our feature matrices, for both the high-specificity and high-coverage datasets.

4.2. *Batch normalization by surrogate variable analysis*

We applied the SVA algorithm to the centered, scaled, and log-transformed input ChIP-map to eliminate technical effects which may obscure biological variation.[18] SVA identifies, in an unsupervised manner, batches of variation across rows and columns of the input data matrix that appear at a frequency greater than expected by chance; each of these batches is represented as a single surrogate variable. We observed that the metadata for the samples had a high rate of missingness; therefore, we devised a novel two-step approach for the removal of technical effects and the protection of biological effects of interest. In the first step, we found statistically significant associations between SVs and known covariates for the set of samples with non-missing metadata using linear regression, where highly significant p-values indicate strong correlations between SV and covariates. As a result, we assigned labels to SVs based on the likely biological or technical variation captured by each SV. In the second step, we removed

the SVs associated with biological effects of interest, and regressed out the remainder from the input data matrix. We investigated the quality of SVs and the preservation of biological signal through manual inspection of principal component analysis plots.

4.3. *Latent factor discovery with singular value decomposition (SVD)*

We applied SVD for our SVA normalized matrix. The normalized matrix, which we denote as W, is of size $N \times M$, where N and M denote the number of ChIP-seq tracks and genomic bins, respectively. We obtained the matrix decomposition, $W = UDV^T$, where $U = (u_{i,k})_{i,k}$ is an orthonormal matrix of size $N \times K$ whose columns are left (ChIP-seq track) singular vectors, D is a diagonal matrix of size $K \times K$ whose elements are singular values, and $V = (v_{j,k})_{j,k}$ is an orthonormal matrix of size $M \times K$ whose columns are right (genomic bin) singular vectors. While singular values in D represent the magnitude of the latent factors, singular vectors in U and V summarize the strength of association between latent factors and ChIP-seq tracks, and latent factors and genomic bins, respectively.

4.3.1. *Quantification of strength of associations between latent factor and genomic bins*

To quantify the strength of associations between latent factor and genomic bins, we define several quantitative scores built on the linear structures of latent factors.[26,27] We first define the **factor score matrix for genomic bins** as $G = VD$. Mathematically, the factor score matrix is equivalent to the matrix consisting of principal component vectors.[26] Each element of this matrix, which we call the **genomic bin factor score** and denote as $g_{j,k}$, is the projection of the j-th column vector in the input matrix W of length N, which represents the epigenetic landscape of j-th genomic bin across samples, to the k-th latent factor (principal component).[26]

To quantify the relative importance of a genomic bin for a given latent factor, we define the **genomic bin contribution score** for k-th latent factor by squaring the genomic bin factor scores for k-th factor and normalizing it across latent factors, i.e.

$$\mathrm{cntr}_k^{\mathrm{bin}}(j) = (v_{j,k})^2 \tag{1}$$

The sum of genomic bin contribution scores across genomic bins is guaranteed to be one, i.e. $\sum_j \mathrm{cntr}_k^{\mathrm{bin}}(j) = 1$, because V is an orthonormal matrix. One can interpret the score as the percent-importance of a genomic bin for the factor.[26,27]

Similarly, to quantify the relative importance of a latent factor for a given genomic bin, we define the **genomic bin squared cosine score** for j-th genomic bin as follows:

$$\cos^{2\,\mathrm{bin}}_j(k) = \frac{(g_{j,k})^2}{\sum_{k'}(g_{j,k'})^2} \tag{2}$$

The sum of genomic bin squared cosine scores across latent factors is guaranteed to be one, i.e. $\sum_k \cos_j^{2\,\mathrm{bin}}(k) = 1$, because of the demoninator in Eq. (2). One can interpret the score as the relative importance of latent factors for a particular genomic bin.

4.3.2. *Quantification of strength of associations between latent factor and samples*

We also define the same set of scores to quantify the strength of associations between latent factors and samples. We first define the **factor score matrix for samples** as $S = UD =$

$(s_{i,k})_{i,k}$. To quantify the relative importance of samples to latent factors and latent factors to samples, we define the **sample contribution score** and the **sample squared cosine scores** as follows:

$$\mathrm{cntr}_k^{\mathrm{sample}}(i) = (u_{i,k})^2\,; \quad \cos^2{}_i^{\mathrm{sample}}(k) = \frac{(s_{i,k})^2}{\sum_{k'}(s_{i,k'})^2} \tag{3}$$

With these scoring systems we can effectively quantify the associations among latent factors, genomic bins, and samples.

4.4. GREAT *analysis for biological characterization of latent factor*

To characterize the functions of latent factors, we applied GREAT version 3.0.0 to each latent factor.[9] Using ontology-based gene annotations as a reference, GREAT takes a set of genomic regions as an input and reports enriched ontology terms. In our analysis, we focused on gene ontology (GO), human phenotype ontology (HPO), and Mouse Genome Informatics (MGI) phenotype ontology.[5–8] For each latent factor, we created the query files for GREAT by selecting the top 5,000 genomic bins ranked by genomic bin contribution score (Eq. (1)) and applied GREAT for these queries using default parameters.[9,27] Given our interest to characterize the putative functions of non-coding genomes, we focused on the GREAT binomial test and collected summary statistics, such as binomial p-value, binomial FDR, and binomial fold change. We sorted the functional terms outputted by GREAT using binomial FDR and identified the ontology terms that most characterize the function of each latent factor.

4.5. *Application of the* SNPs2ChIP *pipeline for GWAS hits and ChIP-seq peaks*

The SNPs2ChIP pipeline consists of three steps: (1) identification of the genomic bins given a user query, (2) identification of the relevant latent factors for the genomic bins, and (3) reporting the results of GREAT enrichment for the relevant latent factors.

4.5.1. *Identification of the genomic bin for a given user's query*

SNPs2ChIP takes genomic coordinates as an input. For GWAS SNPs and ChIP-seq peaks, one first needs to obtain their genomic coordinates. These coordinates are then mapped to the corresponding genomic bins, if they contain a ChIP-seq peak.

4.5.2. *Identification of the relevant latent factor for the genomic bins*

We identify the relevant latent factors for a given genomic bin by genomic bin squared cosine score (Eq. (2)). We can identify the relevant latent factors for multiple genomic bins, which typically corresponds to multiple inputs, by taking a weighted average of genomic bin squared cosine scores. Let's denote $J = \{j_1, \ldots, j_m\}$ be the set of genomic bins of interest and $\{w_1, \ldots, w_m\}$ be the corresponding weights. We defined the weighted average of genomic bin squared cosine score as follows:

$$\cos^2{}_J^{\mathrm{bin}}(k) = \frac{\sum_{j\in J} w_j \cdot \cos^2{}_j^{\mathrm{bin}}(k)}{\sum_{j\in J} w_j} \tag{4}$$

We set the default value of weights to be uniform, i.e. $\{w_1, \ldots, w_m\} = \{1/m, \ldots, 1/m\}$ but the user can specify a set of weights based on external knowledge, such as statistical significance and effect size estimates from GWAS. Once we identify the relevant latent factors, we report the results of GREAT enrichment analysis to the users.

4.5.3. *Systematic application of* SNPs2ChIP *for known GWAS hits*

We downloaded the GWAS Catalog v1.0 from the European Bioinformatics Institute, containing 82,735 curated SNPs.[1] The catalog was subsequently filtered to exclude SNPs that were classified as intergenic to focus on SNPs associated with transcriptional cis-regulation, resulting 51,892 SNPs. Individual SNPs were processed by the SNPs2ChIP pipeline to determine their enriched phenotype. To validate the robustness of the method, SNPs were grouped by disease and run to determine their combined, enriched phenotype. As the pipeline is designed for high-throughput data analysis, querying thousands of SNPs was done in mere seconds.

Acknowledgments

This study was originally conceived as a class project for BIOMEDIN212: "Introduction to Biomedical Informatics Research Methodology" at Stanford University. We thank the teaching team: Hunter Boyce, Steven Bagley, and Russ B. Altman, as well as our classmates for constructive comments. C.S. is supported by the Stanford University BD2K Training Grant (T32 LM012409). L.K. is supported by the Stanford University Biomedical Informatics Training Grant (T15 LM007033). Y.T. is supported by the Funai Overseas Scholarship from Funai Foundation for Information Technology and the Stanford University School of Medicine.

Author contributions

Y.T. conceived and designed the study. S.A., L.K., C.S., and Y.T. designed and carried out the computational analyses. The manuscript was written by S.A., L.K., C.S., and Y.T.

Availability

All the source code used in this project as well as pre-processed reference dataset are available in our GitHub repository: `https://github.com/lkalesinskas/SNPs2ChIP`.

References

1. J. MacArthur, E. Bowler, M. Cerezo *et al.*, The new NHGRI-EBI Catalog of published genome-wide association studies (GWAS Catalog), *Nucleic Acids Research* **45**, D896 (2017).
2. P. M. Visscher, N. R. Wray, Q. Zhang *et al.*, 10 Years of GWAS Discovery: Biology, Function, and Translation., *American journal of human genetics* **101**, 5 (2017).
3. F. Zhang and J. R. Lupski, Non-coding genetic variants in human disease, *Human Molecular Genetics* **24**, R102 (2015).
4. T. S. Furey, ChIP–seq and beyond: New and improved methodologies to detect and characterize protein–DNA interactions, *Nature Reviews Genetics* **13**, 840 (2012).
5. M. Ashburner, C. A. Ball, J. A. Blake *et al.*, Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium, *Nature Genetics* **25**, 25 (2000).

6. The Gene Ontology Consortium, Expansion of the Gene Ontology knowledgebase and resources, *Nucleic Acids Research* **45**, D331 (2017).
7. S. Köhler, N. A. Vasilevsky, M. Engelstad *et al.*, The Human Phenotype Ontology in 2017, *Nucleic Acids Research* **45**, D865 (2017).
8. J. A. Blake, J. T. Eppig, J. A. Kadin *et al.*, Mouse Genome Database (MGD)-2017: Community knowledge resource for the laboratory mouse, *Nucleic Acids Research* **45**, D723 (2017).
9. C. Y. McLean, D. Bristor, M. Hiller *et al.*, GREAT improves functional interpretation of cis-regulatory regions., *Nature biotechnology* **28**, 495 (2010).
10. ENCODE Project Consortium, An integrated encyclopedia of DNA elements in the human genome., *Nature* **489**, 57 (2012).
11. Roadmap Epigenomics Consortium, A. Kundaje, W. Meuleman *et al.*, Integrative analysis of 111 reference human epigenomes, *Nature* **518**, 317 (2015).
12. G. Consortium, Genetic effects on gene expression across human tissues, *Nature* **550**, 204 (2017).
13. S. Oki, T. Ohta, G. Shioi *et al.*, Integrative analysis of transcription factor occupancy at enhancers and disease risk loci in noncoding genomic regions, *bioRxiv* (2018).
14. J. Chèneby, M. Gheorghe, M. Artufel, A. Mathelier and B. Ballester, ReMap 2018: An updated atlas of regulatory regions from an integrative analysis of DNA-binding ChIP-seq experiments, *Nucleic Acids Research* **46**, D267 (2018).
15. S. A. Lambert, A. Jolma, L. F. Campitelli *et al.*, The Human Transcription Factors, *Cell* **172**, 650 (2018).
16. J. Ernst and M. Kellis, Discovery and characterization of chromatin states for systematic annotation of the human genome, *Nature Biotechnology* **28**, 817 (2010).
17. M. M. Hoffman, O. J. Buske, J. Wang *et al.*, Unsupervised pattern discovery in human chromatin structure through genomic segmentation, *Nature Methods* **9**, 473 (2012).
18. J. T. Leek, W. E. Johnson, H. S. Parker, A. E. Jaffe and J. D. Storey, The SVA package for removing batch effects and other unwanted variation in high-throughput experiments, *Bioinformatics* **28**, 882 (2012).
19. G. C. Tsokos, Systemic Lupus Erythematosus, *New England Journal of Medicine* **365**, 2110 (2011).
20. S. J. Rivero, E. Díaz-Jouanen and D. Alarcón-Segovia, Lymphopenia In Systemic Lupus Erythematosus, *Arthritis & Rheumatism* **21**, 295 (1978).
21. S. I. G. Kokori, J. P. A. Ioannidis, M. Voulgarelis, A. G. Tzioufas and H. M. Moutsopoulos, Autoimmune hemolytic anemia in patients with systemic lupus erythematosus, *The American Journal of Medicine* **108**, 198 (2000).
22. L. L. Issa, G. M. Leong and J. A. Eisman, Molecular mechanism of vitamin D receptor action, *Inflammation Research* **47**, 451 (1998).
23. M. F. Holick and T. C. Chen, Vitamin D deficiency: A worldwide problem with health consequences, *The American Journal of Clinical Nutrition* **87**, 1080S (2008).
24. S. V. Ramagopalan, A. Heger, A. J. Berlanga *et al.*, A ChIP-seq defined genome-wide map of vitamin D receptor binding: Associations with disease and evolution., *Genome research* **20**, 1352 (2010).
25. A. R. Quinlan and I. M. Hall, BEDTools: A flexible suite of utilities for comparing genomic features, *Bioinformatics* **26**, 841 (2010).
26. H. Abdi and L. J. Williams, Principal component analysis, *Wiley Interdisciplinary Reviews: Computational Statistics* **2**, 433 (2010).
27. Y. Tanigawa, J. Li, J. M. Justesen *et al.*, Components of genetic associations across 2,138 phenotypes in the UK Biobank highlight novel adipocyte biology, *bioRxiv* (2018).

Extracting allelic read counts from 250,000 human sequencing runs in Sequence Read Archive

Brian Tsui, Michelle Dow, Dylan Skola, Hannah Carter†

Department of Medicine, University of California San Diego, 9500 Gilman

San Diego, California 92093, USA

email: hkcarter@ucsd.edu

The Sequence Read Archive (SRA) contains over one million publicly available sequencing runs from various studies using a variety of sequencing library strategies. These data inherently contain information about underlying genomic sequence variants which we exploit to extract allelic read counts on an unprecedented scale. We reprocessed over 250,000 human sequencing runs (>1000 TB data worth of raw sequence data) into a single unified dataset of allelic read counts for nearly 300,000 variants of biomedical relevance curated by NCBI dbSNP, where germline variants were detected in a median of 912 sequencing runs, and somatic variants were detected in a median of 4,876 sequencing runs, suggesting that this dataset facilitates identification of sequencing runs that harbor variants of interest. Allelic read counts obtained using a targeted alignment were very similar to read counts obtained from whole-genome alignment. Analyzing allelic read count data for matched DNA and RNA samples from tumors, we find that RNA-seq can also recover variants identified by Whole Exome Sequencing (WXS), suggesting that reprocessed allelic read counts can support variant detection across different library strategies in SRA. This study provides a rich database of known human variants across SRA samples that can support future meta-analyses of human sequence variation.

Keywords: Big data, omic analysis, FAIR, variant, single cell

1. Introduction

The reduction of sequencing cost in recent years[1] has allowed researchers to progress from sequencing and analyzing a single reference human genome to studying the individual genomes of thousands of subjects[2]. The large number of sequencing studies being conducted, together with journal publication requirements for authors to deposit raw sequencing runs in a centralized and open access sequencing archive like Sequence Read Archive (SRA)[3] have made it possible to perform large scale data analysis on the millions of publically-available sequencing runs.

The SRA contains raw sequencing runs from a variety of projects from large scale consortium studies including Epigenome Roadmap[4], ENCODE[5], The 1000 Genomes Project[2], to small studies being conducted by various independent laboratories. However, the publicly available raw sequencing data are large in size which translates into high storage and computational requirements that hinder access for the broader research community. These requirements can be somewhat mitigated by using preprocessed data such as gene expression matrices, ChIP-seq peak files, or summarized variant information, as such files are much smaller in size. For example, the 1000 Genomes project, The Cancer Genome Atlas (TCGA)[6] and Genotype-Tissue Expression project (GTEx)[7] all offer summarized variant information extracted from the raw sequences in Variant Call Format (VCF) files, containing allelic read counts for both reference and alternative alleles and base quality information which could be used for variant calling.

There have been many efforts to reprocess raw sequencing reads to a more tractable form. However, many of the SRA data reprocessing efforts[8,9] have focused on quantifying gene expression using public RNA-seq data deposited in the SRA. Sequencing data also capture information about sequence variants, raising the possibility of studying patterns of genetic variation using the SRA.

†: corresponding author

The possibility of extracting variants from RNA-seq was demonstrated on a small scale in a 2015 study[10] where the authors extracted variants using the GATK RNA-seq variant calling pipeline on 5,499 RNA-seq runs in the SRA.

Variant calling typically requires multiple user-specified parameters such as a minimum cut-off for total or read-specific coverage, and usually attempts to model sequencing error explicitly. The primary information used in variant detection is the allelic fraction, the proportion of sequencing reads that support the variant position. Read mapping is highly concordant between alignment tools like bowtie[11], bwa[12], novoalign[13], supporting the idea, at least for DNA and RNA sequencing experiments, estimates of allelic fraction should be fairly consistent regardless of the specific alignment tool. Using a conservative set of known genetic variants that are unlikely to be the result of sequencing errors, simple filters on coverage or allelic fraction should be sufficient to control error rates at acceptable levels. This would make it possible to collect and analyze known variants across the SRA without applying more complex variant callers.

To explore this possibility, we constructed an allelic read count extraction pipeline to systematically reprocess all available sequencing runs from the SRA. We first applied standard quality filtering to the unaligned reads (see Methods) and then aligned the reads to a subset of the human reference genome that covers 390,000 selected somatic and germline variants curated by the NCBI dbSNP[14] using bowtie2[11]. To show that this targeted reference does not introduce unwanted biases into the alignment step, we validated our pipeline performance against alignments performed using whole reference genomes. We next used the TCGA sample-matched Whole Exome Sequencing (WXS) and RNA-seq cohort to confirm that allelic read counts derived from RNA-seq accurately recover variants detected by WXS. We then applied this pipeline to systematically extract variants from over 250,000 sequencing runs in the SRA. Finally, we demonstrated that this allelic read count resource can be used to investigate variants in RNA sequencing studies, even at the single cell level.

2. Results

2.1. *Building a fast allelic fraction extraction pipeline for the SRA*

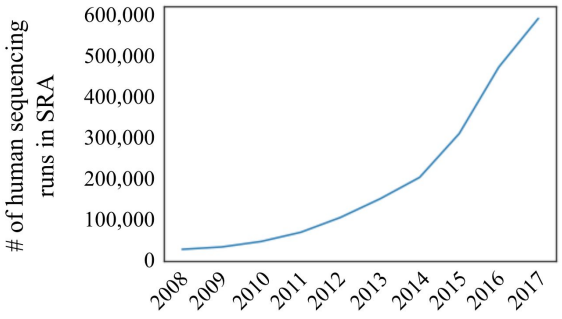


Fig. 1. Number of human sequencing runs are increasing exponentially in the SRA

As of the end of 2017 the SRA included data from 10,642 human sequencing studies consisting of 697,366 publicly available sequencing runs, encompassing various library strategies such as RNA-seq, WXS, whole genome sequencing (WGS), and ChIP-seq (Methods) and this number continues to increase at a rapid pace **(Fig. 1)**. All of the human sequences deposited in the database were derived from samples carrying germline and somatic variants from the corresponding biospecimen regardless of the original study designs. This presents the opportunity to perform meta-analysis of human genetic variation across studies in the SRA.

However, the complete SRA spans over 1,835 trillion bases, introducing both computational and storage resource requirements that would hinder most researchers from conducting a meta-analysis across many sequencing studies. Therefore, to enable efficient secondary analysis for researchers with limited access to high performance computing (HPC) infrastructure, we sought to

Table 1. Key characteristics of variants in targeted reference

Property	Variant type	Number of variants	% of variants
	All	393,242.00	
	Has 3D structure. SNP3D table	20,800.00	5.29
Resource link property	**Cited by PMC article**	170,292.00	43.30
	Cited in PubMed or referenced in a clinical database	201,900.00	51.34
Substitution type	**Non-synonymous missense**	91,827.00	23.35
	synonymous	32,778.00	8.34
	Non-synonymous frameshift	17,824.00	4.53
	Nonsense mutation	9,286.00	2.36
Genotype properties	**Genotypes available, also on high density Genotyping kit and have phenotype associations present in dbGaP**	148,114.00	37.66
Phenotype properties	**Submitted from a locus-specific database**	141,029.00	35.86
	Has OMIM/OMIA	59,617.00	15.16
	Somatic (not germline) variant	37,704.00	9.59

referenced in selected variant databases (OMIM, LSDB, TPA, or in NCBI curated as diagnostic related). The variants consist mostly of missense mutations with synonymous and truncating mutations accounting for about 15% of the database. Most are germline variants, although the dataset includes a small set of curated somatic mutations[15]. The characteristics of the variants are summarized in **Table 1**.

We created the reference alignment index by masking the reference to exclude DNA sequences outside of a region spanning the 1000 base pairs upstream and 1000 base pairs downstream of each variant. This filtering method had been first adopted by Deng *et al*. to optimize sequencing data processing turnaround times[16].

process this vast amount of data into a form that can fit on a 1 TB hard disk. To accomplish this, we developed an efficient data processing pipeline (**Fig. 2**).

We first created a targeted alignment reference that focuses on regions that harbor known variants (n=393,242) curated by NCBI dbSNP[14]. These consist predominantly of variants with PubMed references or that have been

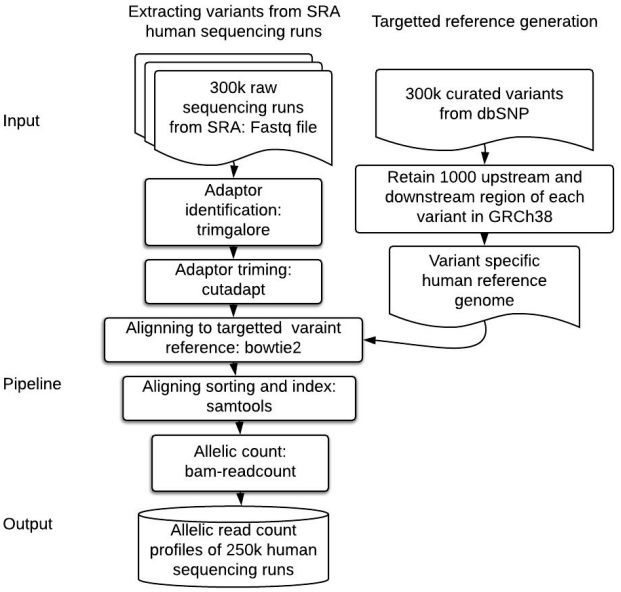


Fig. 2. Simple pipeline for extracting >300,000 human sequencing runs from SRA. For each sequencing run, first adaptors are identified and trimmed from raw sequencing reads. Then we align the reads to the targeted reference and extract the allelic read counts.

2.2. *Large scale allelic read count extraction of human sequence data*

We retained only sequencing runs from the top five library strategies (RNA-seq, WGS, WXS, AMPLICON, ChIP-seq), and sequencing runs with more than 150 million bases sequenced (equivalent to at least three million reads if the samples have 50 bp per read), corresponding to a total of 304,939 sequencing runs. Of these, 253,005 were successfully processed (**Fig. 3**) without error with 300 cpu-cores in 30 days. Library strategies were divided between paired-end (64.8%) and single-end (35.2%) sequencing. The difference between the number of pair-end sequencing and single-end sequencing reflects the differing needs of various experimental designs (Supplementary

Table 1). For example, paired-end sequencing greatly improves the identification of splice isoforms in RNA-seq and structural variants in exome-seq, whereas it provides fewer benefits for other library types that would justify the increased cost relative to single-end sequencing.

One utility that emerges from reprocessing the sequencing data is for imputing experimental annotations. For example, the SRA metadata is not standardized to contain important experimental variables like read length or adaptor sequences, however this information can be easily determined from the raw sequences. A median read length of 95 bp was observed. Most runs (206,360 = 81.56%) had adaptors automatically detected and removed. Sequence and mapping statistics are detailed in the Supplementary Table 1. Over these sequencing runs, a median of 2.98% of base pairs were identified as adaptors and were removed. A median base quality Phred score of 36 was observed, suggesting a high overall quality of the sequenced bases in the SRA.

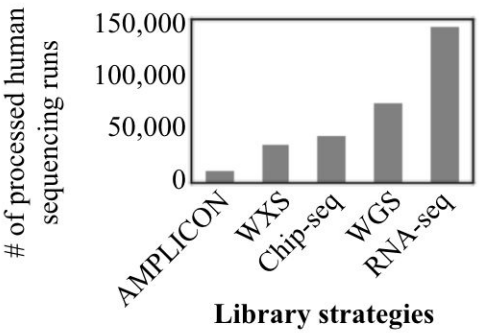


Fig. 3. Distribution of processed SRA data

Overall, a median of 296.3 million bases and 10,044,529 read fragments per sample were observed. A median of 5.83% of the reads were aligned to the targeted variant regions (**Methods**). Adding read length, adaptor contents, number of reads and percentage aligned to the metadata allows the user to better understand the quality of the sequencing runs and filter them accordingly.

2.3. *Pipeline performance for targeted variant detection*

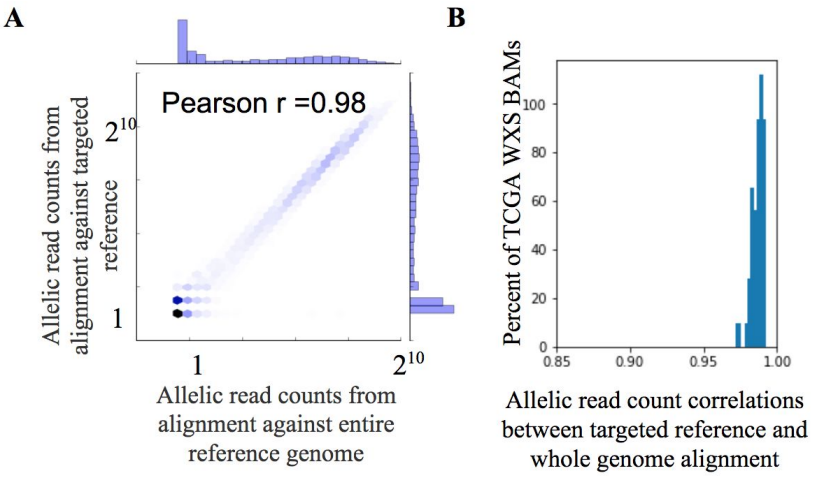


Fig. 4. Targeted reference remains accurate for sequence alignment. **A** Hex density plot showing the high allelic read count correlation between the whole genome alignment (x-axis) and targeted reference alignment (y-axis). Histogram of allelic read counts on whole genome alignment (x-axis, top) and on targeted genome alignment (y-axis, right). **B** Distribution of allelic read count correlations (x-axis) over TCGA WXS BAMs (y-axis).

To assess the accuracy of allelic read counts extracted from this targeted reference we compared counts obtained through our pipeline to those extracted from samples pre-aligned to the complete hg38 genome index and downloaded directly from the TCGA. We also took advantage of matched DNA/RNA sequencing in TCGA to evaluate the extent to which allelic read counts extracted from RNA-seq reflect the variants detected from WXS (See section 2.5). We used 524 whole exome tumor sequences from the TCGA Low Grade Glioma (LGG) dataset to assess the performance of our pipeline, as this dataset included the well-known variant (IDH1 R132H) which could serve as a positive control.

The reads from each tumor were aligned to the targeted SNP index and the allelic read counts were compared to the pre-generated alignments available from the TCGA. The resulting variant-locus-by-nucleotide read count matrix contains the read count for each of the four

nucleotides across the 393,242 targeted variants at 387,950 genomic sites. We then flattened the nucleic base read count matrix into a single allelic read count vector. For each sample, we compared allelic read counts for all variants obtained using alignment to a targeted reference against allelic read counts obtained from the existing TCGA alignments to a complete reference. Read counts were highly correlated. **Figure 4A** shows an example from a single TCGA tumor (UUID: 2b0048e0-a062-40d2-a1e1-4bb763ea0ead), in which a median of 98.2% variants differed less than one $\log_2$ fold change in allelic read count from the existing alignment (95% confidence interval: 0.0088 - 0.0554). We found similar correlation across all 524 samples, with a median Pearson correlation (R) of 0.98 for the allelic read counts (95% CI: 0.928 - 0.992; **Fig. 4B**).

2.4. *Effects of PCR duplicates on estimating allelic fraction*

We next evaluated the necessity of removing putative PCR duplicate reads after alignment based on the extent to which such duplicates bias the estimate of allelic fraction in TCGA. Although most sequence alignment pipelines include a step for removing duplicate reads that result from PCR amplification, recent studies have cast doubt on the benefit of doing so for variant analysis[17,18]. Also, naively removing the duplicated reads could result in overcorrection in high coverage sequencing[19]. We therefore investigated the effect of sequence duplicate removal for all 300k targeted variants across the 524 samples. We compared the allelic read counts extracted with and without duplicate removal for each tumor WXS alignment, and observed a median correlation of 0.983 (95% CI: 0.983-0.990), suggesting duplicate removal had limited impact on allelic read counts. However, we did observe a substantial bias in allelic read count estimates when duplicates are included among sites with very high sequence read coverage. **Figure 5A** shows an example using UUID: 0e2c395e-ddda-4833-b1ee-31a9bd08a845. In this sample, deduplicated allelic read counts recover 88.9% of the original allelic read counts among all the variants with ≤100 reads support, while the deduplicated allelic only recover 33.7% of the original allelic read count among all the variants with >100 reads, a 2.63 fold reduction in read count extracted from in the high coverage region (**Fig. 5A**, slope of grey bar and red bar respectively). Nonetheless, across all 524 samples we observed a difference in allelic fraction < 0.05 for over 90% of the variants when duplicates were excluded, except in extreme cases with over 10,000 mapped reads (median 0.4% of the variants) (**Fig. 5B**).

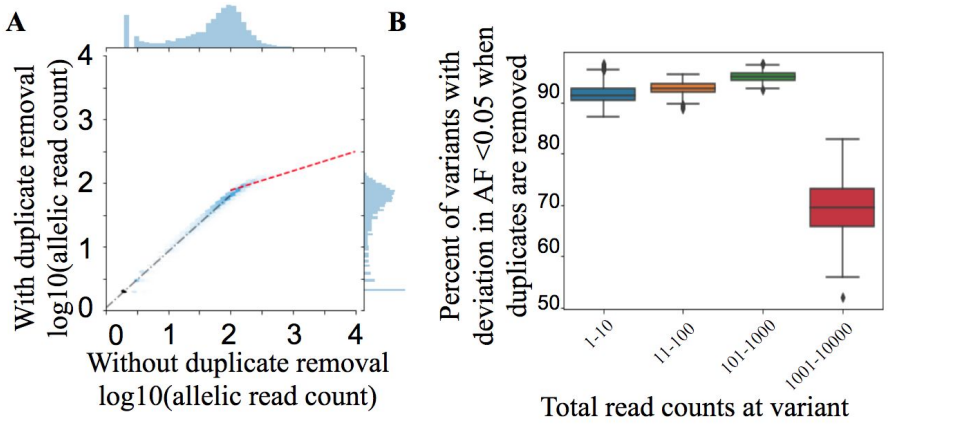


Fig. 5. A Among regions with <100 reads (grey dashed line), allelic read counts correlate linearly between alignments with duplicate removal (y-axis) and without duplicate removal (x-axis). However, duplicate removal may potentially underestimate read counts in regions with ≥100 reads (red dashed line). **B** Allelic fraction are comparable regardless of duplicate removal except in sites with extremely high read count.

Thus with high quality sequencing data, filtering duplicates should result in only minor improvement to the data.

2.5. *Evaluating variant extraction from RNA-seq using matched DNA/RNA samples*

The SRA includes over 100k RNA-seq runs and these data contain information about the variant status of the transcribed DNA. To determine the extent to which variants can be extracted from RNA-seq by our pipeline, we first compared allelic fractions between matched exome sequencing on the one hand and RNA sequencing data in TCGA on the other. TCGA contains samples which have been subjected to both WXS and RNA-seq, which makes it a natural resource for comparing the performance of variant calls derived from RNA-seq data using the WXS-derived variants. We evaluated the possibility of using allelic read counts from RNA-seq to detect both germline and somatic variants.

To evaluate the reliability of allelic read counts for identifying germline variants in RNA sequence reads, we first compared read fractions for germline variants that were homozygous in the corresponding TCGA WXS sample. After collecting all sites that had at least 10 reads and were homozygous for the variant allele in the WXS read data, we evaluated the read counts at those same sites in the RNA-seq data. A median of 5827 sites had at least 10 reads to support the variant in both WXS and RNA-seq for each sample. Across all samples, a median of 97% (95% CI: 95.5% - 97.9%) of sites that were homozygous in the DNA were also found to be homozygous in the matched RNA-seq data.

Next, we explored the utility of allelic read counts for identifying somatic mutations from RNA sequencing data. First, as a positive control, we evaluated the hotspot IDH1 somatic mutation on chromosome 2:208248388 with 395G>A in the template strand, which is most prevalent somatic variant in TCGA LGG on WXS as called by Varscan [20] (n=371, 70.80% of patients). This variant had been previously identified as enriched in LGG tumors and its status is a major molecular prognostic factor in glioma as noted by the World Health Organization (WHO)[21]. Using the 524 LGG tumors, we estimated allelic composition using read counts in the matched RNA-seq and WXS independently with our pipeline. The IDH1 mutation status in WXS exhibits a bimodal distribution (**Fig. 6A**). We selected 10 reads as the cutoff for defining a positive WXS variant. The reference allele was detected in the WXS in all tumors, and 351 patients also had the alternative allele. Over these patients the RNA-seq achieved an area under the precision recall curve (AUPRC) of 0.98 in detecting IDH1 variants observed in the WXS data (**Fig. 6B**).

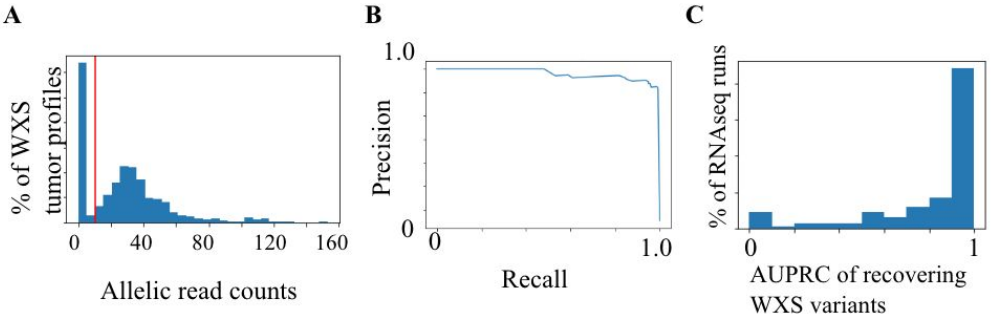


Fig. 6 RNA-seq can recover variants extracted from WXS. A Minor allelic read counts of IDH1 hotspot mutation. Vertical red line is the binomial distribution cutoff (10 read counts). **B** distribution of minor allele of IDH1 (395C>T in template strand). **C** RNAseq has high area under the precision recall curve (AUPRC) of recovering WXS variants

We next evaluated the top 100 most frequently observed somatic variants reported by TCGA in the LGG samples that also coincided with the targeted variants, since recurrent mutations are more likely to be

drivers and present the most attractive therapeutic targets[22]. We used the Precision Recall Curve (PRC) framework to determine the extent to which allelic read counts supported expression of the mutant allele. RNA-seq generally recapitulated WXS variants (**Fig. 6C**), with 70% of the variants having an AUPRC > 0.8, suggesting that majority of the variants called by exome sequencing are expressed in the tumor. However, we do observe 6% of the variants with an AUPRC less than 0.1 when their presence was predicted from RNA-seq allelic fraction. Importantly, these later variants were found in fewer than 10 WXS samples, such that the most recurrent somatic mutations are also more frequently consistently expressed. Thus while absence of a somatic variant cannot be definitively determined from RNA-seq (mutations can be present but not expressed), the most recurrent variants appear to be frequently expressed, suggesting that many somatic mutations of interest will be detectable in RNA-seq data from cancer studies deposited in the SRA.

2.6. Variant landscape of the SRA

After validating the general reliability of our allelic fraction estimates, we analyzed 300K variants across the SRA. Properties of the variants are listed in Table 1. Of 300K variants, 170,292 were referenced by PubMed and 138,559 were curated by NCBI as clinically-relevant variants. Out of 156,757 variants with annotated functional effects, the majority were missense mutations (n=91,827). Also, 37,704 variants were annotated as somatic mutations, derived from cancer studies. Overall, the data included a median of three variants per gene across 21,889 genes. We collected read counts for reference and alternative alleles at these 300K positions for 253,005 human sequencing samples in the SRA. We used default minimum threshold of two reads[23] as the cut-off for Varscan[20]. The distribution of the number of variants are shown in Figure 7. Known germline variants were detected in a median of 912 sequencing runs, known somatic variants were detected in a median of 4,876 sequencing runs, and known reference alleles were detected in a median of 33,232 sequencing runs. 337 somatic variants, 3,068 germline variants and 23,044 reference alleles were covered by at least two reads in more than half of the sequencing runs, suggesting that SRA data can be repurposed for studying many variants. To facilitate the analysis of variants, we collected allelic read count in each SRA sample into a table (see Data Availability). This read count file allows researchers to rapidly identify which sequencing runs in the SRA have read support for a particular variant.

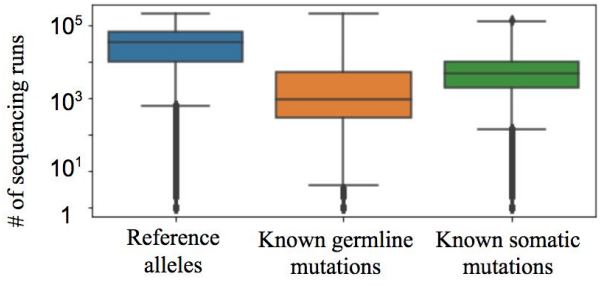


Fig. 7 Distribution of variants detected associated with each variant type

2.7. Extracting unannotated single cell variants in cancer in SRA

Genotype annotations are often missing or incomplete in the SRA, and this limits the reusability of the SRA data. Here, we show that, using the reprocessed data, we were able to recover an important oncogenic mutation BRAF V600E in a single cell RNA-seq study of a patient with myeloid leukemia at diagnosis and as well as at three and six months after diagnosis [24].

Traditional variant calling relies on high sequencing depths to provide the statistical power to make confident calls. However, since each cell carries only two copies of each chromosome, the low recovery of single cell sequencing makes variant calling from DNA resequencing difficult.

Since RNA also contains information about underlying variants and may exist at hundreds of copies per cell[25], calling variants from single-cell RNA-seq data may circumvent the limitations of DNA resequencing for variants in transcribed regions.

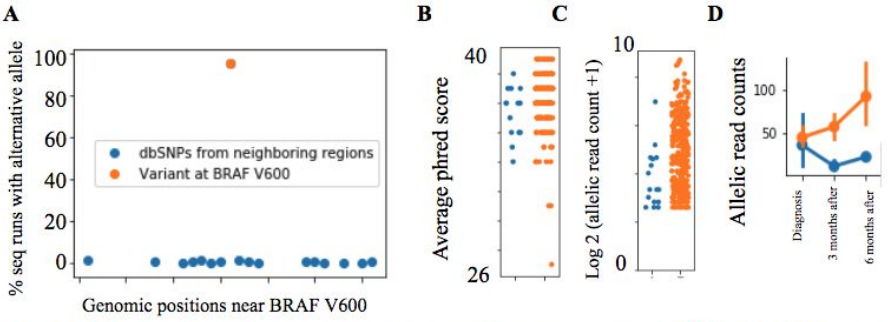


Fig. 8 A allelic read counts can recover obvious variants (example: chr7-140753336). **B** Base quality, and **C** read count of reads at chr7-140753336 for reference allele (blue) and alternative allele (orange). **D** Allelic read count of alternative allele can track cancer progression.

We were able to detect an important oncogenic mutation, BRAF V600E, in single cells using our unified allelic read counts. The overall read depth for the region was 45.9 reads and 17 sites within the 20 bp windows around BRAF V600E had read support for the reference allele. Alternative alleles at the BRAF V600 hotspot were detected in more than 95% cells (**Fig. 8A**). Also, the alternative allele (T) had a median base quality Phred score of 38 (**Fig. 8B**) and a median of 22.0 reads to support it (**Fig. 8C**). Interestingly, we observed a reduction in the reference allele read count over the course of treatment (**Fig. 8D**) with a corresponding higher fraction of reads supporting the alternate allele, suggesting that the clone with BRAF mutations became more prevalent among the surviving cancer cells, concording with the observation in the study that relapse occurred after treatment.

3. DISCUSSION

Most published studies on non-protected raw sequencing data are expected to be deposited in the NCBI SRA as a result of journal requirements, and this vast amount of raw sequencing data represents a an opportunity to power large-scale meta-analyses for the interaction of sequence variants with experimental conditions. However, these petabytes worth of sequencing data introduce a computational challenge for analyzing such variants. One solution is to develop a map of relevant sequence variants in the SRA using allelic count profiles.

To create allelic read count profiles from the SRA, we constructed a bioinformatics pipeline with short processing turnaround time by mapping the raw sequencing reads to a targeted reference specific to key somatic and germline variant(s) curated by the NCBI dbSNP. We validated the accuracy of the pipeline by comparing read counts obtained with targeted alignment to counts obtained using complete alignment pipelines, and evaluated genotype consistency across multiple sequencing datasets derived from the same sample. These results confirm that the targeted alignment pipeline generates allelic read counts that are highly correlated to those from whole genome alignments.

Variant calling has traditionally been performed from DNA sequences, but WXS and WGS library strategies comprise only 40% of the total human SRA data. Thus we also sought to infer the presence of variants from RNA-seq allelic read counts. While RNA may be less reliable for inferring the presence or absence of variants due to gene and allele-specific expression, 61.8% of the RNA-seq samples have more than a million reads mapped onto the targeted variant regions. We also found that highly recurrent somatic mutations detected in WXS of low grade gliomas were also frequently expressed in matched RNA-seq data. Thus, it would also be interesting to utilize the

germline allelic read counts extracted from the SRA RNAseq dataset to conduct a large-scale systematic EQTL study. We may also use the somatic allelic read counts in single cell cancer studies to help decipher the interactions between clonal mutations and clonal expressions in tumor heterogeneity.

To the best of our knowledge, this is the first attempt to massively reprocess the human samples in the SRA for the purpose of extracting allelic read counts. The computational infrastructure required to generate variant data at scale presents a barrier to many researchers. Consortia that generate a large volume of sequencing data, such as GTEx, TCGA or the 1000 Genome Project, all offer preprocessed files that enable researchers from the broader community to identify novel findings. Although variant calls are available for some of the datasets included in SRA, significant effort would be required to aggregate these disparate datasets, and most of the non-consortia SRA samples do not have such data available. Simply providing allelic read counts derived through a common bioinformatic pipeline also avoids technical variation that can result from different choice of computational tools and their associated parameter choices. Therefore, we contend that our unfiltered allelic read counts will have broad utility for *post hoc* analysis.

Many applications require estimates of the magnitude of allelic fraction for inference. This would be particularly useful for questions related to imprinting or reconstruction of tumor subclonal architecture. We found that presence of duplicate reads did not significantly bias estimates of allelic fraction when the quality of the sequencing data is high. However for lower quality datasets or different library strategies, it may still be necessary to remove duplicate reads to obtain high quality estimates. Further analysis is merited to determine which datasets or variants are most confounded if duplicates are not removed. Future releases of the database will include estimates of allelic fraction both before and after removing PCR duplicates.

In conclusion, by reprocessing the raw sequencing runs from the SRA, we improve the findability, accessibility, interoperability, reusability (FAIR)[26] of of 250,000 sequencing runs. As the SRA continues to grow, it will be necessary to continuously update the map of variants present in SRA samples. To support variant meta-analyses using the SRA, the next requirement will be unification of the SRA data, including biospecimen and experimental annotations. We anticipate that further refinement of the SRA through efforts such as this will promote reanalysis of existing datasets and lead to new biological discoveries.

4. METHODS

4.1. *SRA Metadata download*

SRA metadata (files: NCBI_SRA_Metadata_Full_.tar.gz and SRA_Run_Members.tab) were downloaded from ftp.ncbi.nlm.nih.gov/sra/reports/Metadata/ on Jan 4 2018. These files contain the raw freetext biospecimen and experimental annotations. SRA_Run_Members.tab details the relationships between SRA project ID (SRP), sample ID (SRS), experiment ID (SRX) and sequencing run IDs (SRR). We processed only sequencing runs with accession visibility status “public”, with availability status “live”, and sequencing runs that contains more than 150 million nucleotides bases. We also only included sequencing runs generated from the following library strategies: RNA-Seq, WGS, WXS, ChIP-Seq, AMPLICON. Only samples with layout defined as either SINGLE or PAIRED were considered. We removed SRA study ERP013950 as we noticed it has annotation indicating a total of 85,608 WGS sequencing runs which seem to stem from erroneous submission, as it was only associated with nine biological samples (BioSample) IDs and the experimental annotation was unclear on the nature of the study.

4.2. *NCBI dbSNP structure*

NCBI dbSNP[14] curated a set of SNPs and uses each bit in the bitfield encoding schema to indicate a specific evidence support (ftp://ftp.ncbi.nlm.nih.gov/snp/specs/dbSNP_BitField_latest.pdf). Some evidence supports are derived from databases, for example, NCBI ClinVar (https://www.ncbi.nlm.nih.gov/clinvar/), Online Mendelian Inheritance in Man (OMIM, url: https://www.omim.org/), Locus-Specific DataBases (LSDB, url: http://www.hgvs.org/locus-specific-mutation-databases), and Third Party Annotation (TPA, url: https://www.ddbj.nig.ac.jp/ddbj/tpa-e.html). ClinVar contains a curated set of published human variant-phenotype associations. OMIM contains the genotypes and phenotypes of all known mendelian disorders for over 15,000 human genes. LSDB provides gene-centric links to various databases that collect information about variant phenotypes. TPA is a nucleotide sequence data collection assembled from experimentally determined variants from DDBJ, EMBL-Bank (https://www.ebi.ac.uk/), GenBank, International Nucleotide Sequence Database Collaboration (INSDC) (http://www.insdc.org/), and/ or Trace Archive (https://trace.ncbi.nlm.nih.gov/Traces/home/) with additional feature annotations supported by peer-reviewed experimental or inferential methods.

4.3. *Targeted reference building*

Variants were obtained from dbSNP (downloaded on 4, January on 2017 from ftp://ftp.ncbi.nlm.nih.gov/snp/organisms/human_9606_b150_GRCh38p7/VCF/00-All.vcf.gz), which contained 325,174,853 sites in total, effectively one tenth of our selected human reference genome length (3,099,734,149 bp, version: hg38). We retained only variants with a resource link to any of the existing databases or with support from NCBI curation, indicated by a non zero value for byte 2 of Flag 1 in the NCBI bit field encoding schema, resulting in 393,242 variants. To generate a targeted reference for these variants, we defined 1000 bp downstream and 1000 bp upstream of each SNP as the mapping window. All the regions outside of the windows were masked with base "N" using bedtools v2.26.0 in the reference FASTA file. The reference index was built using bowtie2 v2.2.6[11] with the merged FASTA file, using default parameters.

4.4. *Extracting variants from raw sequencing read FASTQ file*

We used SRA[3] prefetch v2.8.0 to download SRR files. Next, fastq-dump v2.4.2 from SRA tool kit was used to extract FASTQ files from SRR into the standard output stream. Trim Galore! version 0.4.0 (url: https://github.com/FelixKrueger/TrimGalore) was then applied to identify adapter sequences using the first 10,000 reads, and the identified adaptor sequence was trimmed in the FASTQ file using cutadapt version 1.16[27], the trimmed reads were then aligned onto the targeted reference (we did not use Trim Galore! to trim the adaptor as it cannot be easily UNIX piped). Bowtie2 was run with the "--no-unal" parameter to retain only the reads mappable to the target regions in order to minimize the amount of aligned reads for sorting. The alignment file was than sorted using samtools v1.2. and samtools idxstats was used for calculating the number of reads that mapped onto each FASTA reference record. bam-readcount v0.8.0 was used for extracting the per-base allelic read count and per-base quality in the sorted alignment file for each of the targeted genomic coordinates. The paired-end reads were processed the same way as the single-end reads with the exception that paired-end and interleave reads options in fastq-dump, cutadapt, and bowtie2, were specified to ensure proper treatment of paired-end reads. The allelic read counts consist of both the reference allele and alternative allele, and they are retained in the output regardless of the zygosity.

4.5. *TCGA download*

A gdc_manifest was downloaded from the gdc portal on 2017-12-27. We downloaded the TCGA data using gdc-client v1.3.0. We downloaded the associated metadata using the TCGA REST API interface https://api.gdc.cancer.gov/files/. All the alignment files preprocessed from TCGA using GATK pipeline were downloaded. The alignment files were mapped onto GRCh38 with all the raw reads, including read sequence duplicates.

5. Supplementary code and data availability

The python scripts for the pipeline and the jupyter-notebooks for generating the figures are deposited on github (https://github.com/brianyiktaktsui/Skymap) and the data is publicly available on synapse (https://www.synapse.org/#!Synapse:syn11415602). Supplementary table 1 is available on http://hannahcarterlab.org/skymapvariantpsbsupplementarytable1/.

6. Acknowledgments

We thank all members of the Carter, Mesirov and Ideker lab for scientific feedback and comments. The results here are partly based upon the data generated by the TCGA Research Network: http://cancergenome.nih.gov/. This work was funded by NIH grants DP5-OD017937, RO1 CA220009 and a CIFAR fellowship to H.C.; National Library of Medicine Training Grant T15LM011271 to M.D. Preprint of this article is submitted for consideration in Pacific Symposium on Biocomputing © 2018 copyright World Scientific Publishing Company.

References

1. Wetterstrand, K. A. DNA sequencing costs: data from the NHGRI Genome Sequencing Program (GSP). (2013).
2. The 1000 Genomes Project Consortium. A global reference for human genetic variation. *Nature* **526,** 68 (2015).
3. Leinonen, R., Sugawara, H., Shumway, M. & International Nucleotide Sequence Database Collaboration. The sequence read archive. *Nucleic Acids Res.* **39,** D19–21 (2011).
4. Roadmap Epigenomics Consortium *et al.* Integrative analysis of 111 reference human epigenomes. *Nature* **518,** 317–330 (2015).
5. ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature* **489,** 57–74 (2012).
6. Cancer Genome Atlas Research Network *et al.* The Cancer Genome Atlas Pan-Cancer analysis project. *Nat. Genet.* **45,** 1113–1120 (2013).
7. Carithers, L. J. & Moore, H. M. The Genotype-Tissue Expression (GTEx) Project. *Biopreserv. Biobank.* **13,** 307–308 (2015).
8. Collado-Torres, L. *et al.* Reproducible RNA-seq analysis using recount2. *Nat. Biotechnol.* **35,** 319–321 (2017).
9. Lachmann, A. *et al.* Massive Mining of Publicly Available RNA-seq Data from Human and Mouse. *bioRxiv* 189092 (2017). doi:10.1101/189092
10. Deelen, P. *et al.* Calling genotypes from public RNA-sequencing data enables identification of genetic variants that affect gene-expression levels. *Genome Med.* **7,** 30 (2015).
11. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nat. Methods* **9,** 357–359 (2012).
12. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows-Wheeler transform.

Bioinformatics **25,** 1754–1760 (2009).
13. Ruffalo, M., LaFramboise, T. & Koyutürk, M. Comparative analysis of algorithms for next-generation sequencing read alignment. *Bioinformatics* **27,** 2790–2796 (2011).
14. Kitts, A. & Sherry, S. *The Single Nucleotide Polymorphism Database (dbSNP) of Nucleotide Sequence Variation*. (National Center for Biotechnology Information (US), 2011).
15. *Classes of Genetic Variation Included in dbSNP*. (National Center for Biotechnology Information (US), 2005).
16. Deng, J. *et al.* Targeted bisulfite sequencing reveals changes in DNA methylation associated with nuclear reprogramming. *Nat. Biotechnol.* **27,** 353–360 (2009).
17. Ebbert, M. T. W. *et al.* Evaluating the necessity of PCR duplicate removal from next-generation sequencing data and a comparison of approaches. *BMC Bioinformatics* **17 Suppl 7,** 239 (2016).
18. Stratford, J. *et al.* Abstract 5276: Impact of duplicate removal on low frequency NGS somatic variant calling. *Cancer Res.* **76,** 5276–5276 (2016).
19. Zhou, W. *et al.* Bias from removing read duplication in ultra-deep sequencing experiments. *Bioinformatics* **30,** 1073–1080 (2014).
20. Koboldt, D. C. *et al.* VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Res.* **22,** 568–576 (2012).
21. Louis, D. N. *et al.* The 2016 World Health Organization Classification of Tumors of the Central Nervous System: a summary. *Acta Neuropathol.* **131,** 803–820 (2016).
22. Chang, M. T. *et al.* Identifying recurrent mutations in cancer reveals widespread lineage diversity and mutational specificity. *Nat. Biotechnol.* **34,** 155–163 (2016).
23. Xu, C. A review of somatic single nucleotide variant calling algorithms for next-generation sequencing data. *Comput. Struct. Biotechnol. J.* **16,** 15–24 (2018).
24. Giustacchini, A. *et al.* Single-cell transcriptomics uncovers distinct molecular signatures of stem cells in chronic myeloid leukemia. *Nat. Med.* **23,** 692–702 (2017).
25. Albayrak, C. *et al.* Digital Quantification of Proteins and mRNA in Single Mammalian Cells. *Mol. Cell* **61,** 914–924 (2016).
26. Wilkinson, M. D. *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* **3,** 160018 (2016).
27. Martin, M. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal* **17,** 10–12 (2011).

Semantic workflows for benchmark challenges: Enhancing comparability, reusability and reproducibility*

Arunima Srivastava
Computer Science and Engineering, The Ohio State University, 2015 Neil Ave Columbus, OH 43210
Email: srivatava.1@osu.edu

Ravali Adusumilli, Hunter Boyce
Canary Center for Cancer Early Detection, Stanford University, 3155 Porter Dr., Palo Alto, CA, 94305
Email: ravali@stanford.edu, hboyce@stanford.edu

Daniel Garijo, Varun Ratnakar, Rajiv Mayani
Information Sciences Institute, University of Southern California, Marina del Rey, Los Angeles ,CA 90292
Email: dgarijo@isi.edu, varunr@isi.edu, mayani@isi.edu

Thomas Yu
Sage Bionetworks, 2901 Third Ave., Suite 330, Seattle WA 98121
Email: thomas.yu@sagebionetworks.org

Raghu Machiraju
Computer Science and Engineering, The Ohio State University, 2015 Neil Ave Columbus, OH 43210
Email: machiraju.1@osu.edu

Yolanda Gil
Information Sciences Institute, University of Southern California, Marina del Rey, Los Angeles ,CA 90292
Email: gil@isi.edu

Parag Mallick#
Canary Center for Cancer Early Detection, Stanford University, 3155 Porter Dr., Palo Alto, CA, 94305
Email: paragm@stanford.edu

Benchmark challenges, such as the Critical Assessment of Structure Prediction (CASP) and Dialogue for Reverse Engineering Assessments and Methods (DREAM) have been instrumental in driving the development of bioinformatics methods. Typically, challenges are posted, and then competitors perform a prediction based upon blinded test data. Challengers then submit their answers to a central server where they are scored. Recent efforts to automate these challenges have been enabled by systems in which challengers submit Docker containers, a unit of software that packages up code and all of its dependencies, to be run on the cloud. Despite their incredible value for providing an unbiased test-bed for the bioinformatics community, there remain opportunities to

* The work is partially supported by DARPA Deep Purple Program through a DOI contract #D17AC00006, by DARPA SIMPLEX program award W911NF-15-1-0555, and by NIH award 1R01GM11709701.

Corresponding Author

further enhance the potential impact of benchmark challenges. Specifically, current approaches only evaluate end-to-end performance; it is nearly impossible to directly compare methodologies or parameters. Furthermore, the scientific community cannot easily reuse challengers' approaches, due to lack of specifics, ambiguity in tools and parameters as well as problems in sharing and maintenance. Lastly, the intuition behind why particular steps are used is not captured, as the proposed workflows are not explicitly defined, making it cumbersome to understand the flow and utilization of data. Here we introduce an approach to overcome these limitations based upon the WINGS semantic workflow system. Specifically, WINGS enables researchers to submit complete semantic workflows as challenge submissions. By submitting entries as workflows, it then becomes possible to compare not just the results and performance of a challenger, but also the methodology employed. This is particularly important when dozens of challenge entries may use nearly identical tools, but with only subtle changes in parameters (and radical differences in results). WINGS uses a component driven workflow design and offers intelligent parameter and data selection by reasoning about data characteristics. This proves to be especially critical in bioinformatics workflows where using default or incorrect parameter values is prone to drastically altering results. Different challenge entries may be readily compared through the use of abstract workflows, which also facilitate reuse. WINGS is housed on a cloud based setup, which stores data, dependencies and workflows for easy sharing and utility. It also has the ability to scale workflow executions using distributed computing through the Pegasus workflow execution system. We demonstrate the application of this architecture to the DREAM proteogenomic challenge.

Keywords: Workflows; Semantic Workflows; DREAM Challenges; Proteogenomics; Benchmarking; Big Data

1. Introduction

The volume of experimental data being generated in the field of experimental biology is growing at a rapid pace in both size and variety[1,2]. With the advent of increasingly diverse data types, many of which are high throughput, the bioinformatics community is introducing sophisticated computational approaches for data analysis[3,4].

To compare different approaches, community-wide competitive benchmark challenges have gained popularity as an unbiased method to better understand the variety of pipelines proposed by different groups. Popular challenges include the Dialogue for Reverse Engineering Assessments and Methods (DREAM)[5], Critical Assessment of Structure Prediction (CASP) protein structure prediction[6] and The Association of Biomolecular Resource Facilities' (ABRF) Proteome Informatics Research Group's (iPRG) detection and prediction challenges[7]. These challenges give competitors the opportunity to test (in a blind and unbiased manner) their approach against others in the field, and have been instrumental in advancing diverse areas from protein structure prediction[8] to variant calling[9] to analysis of pathology data[10].

Unfortunately, evaluations in these competitions have traditionally been limited to metrics that evaluate solely based on scores. Comparisons of the methods that gave rise to those results are often left to manual interpretation. When the difference between a winner and an extremely poor performer may come down to a handful of parameters in otherwise identical workflows, the lack of transparency in methods is a huge missed opportunity for the bioinformatics community. In addition, winning methods are rarely shared with the broader community, as it is cumbersome to make winning methods accessible beyond the competition framework. Thus, while these

challenges provide a forum for bioinformatics researchers to independently evaluate the performance of their approaches against others, the current execution environment for challenges does not facilitate deep comparison and sharing of approaches.

Consequently, there is a critical need to reconsider the infrastructure used for executing benchmark challenges. Here we examine the potential benefits of conducting benchmark challenges within a semantic workflow environment. Workflow environments, such as Galaxy[11] and GenePattern[12], would enable a challenge to examine not just the final results, but also all the steps of a method. This could include all dependencies, relevant data, and workflow components. By having challengers enter their submissions as workflows, which are executed on challenge data in the cloud, it becomes possible to more deeply perform a meta-analysis of the entries. In addition, submissions could be easily reused and shared by members of the broader scientific community.

This work describes our effort to date using the WINGS[13] semantic workflow system to submit entries to the DREAM proteogenomic challenge. While WINGS is an established (ready-to-download for server) workflow system[14], employing it as a submission and storing protocol for data analysis challenges is a novel use of this framework. In addition to the advantages typical of workflow systems, WINGS has additional features due to its use of semantic representations and reasoning about workflow steps and data. WINGS uses semantic annotations of data characteristics and step requirements in order to facilitate the selection of appropriate input parameter values based on metadata. WINGS additionally supports the creation of an abstract workflow component for a class of tools that perform a similar task, which greatly facilitates the comparison of different challenge entries. Finally, WINGS uses the W3C PROV standard[15] to record the complete provenance of the workflow execution details that led to a final result, including what tools and versions were used, how algorithm parameters were set, and the overall method. Key features of the execution environment of WINGS include: (a) a framework for recording all runtime dependencies of multi-step workflows, where each step is a self–contained component facilitated by employing Docker[16] images. Docker offers a virtual platform for building, sharing and running application within self-sufficient "containers" which allow encapsulation and storage of WINGS workflows. This includes the tools and data underlying each step (facilitating benchmarking), (b) a dynamic cloud based environment to house these workflows, complete with all runtime dependencies and data (facilitating reproducibility), and (c) a scalable execution environment (combination of WINGS and the

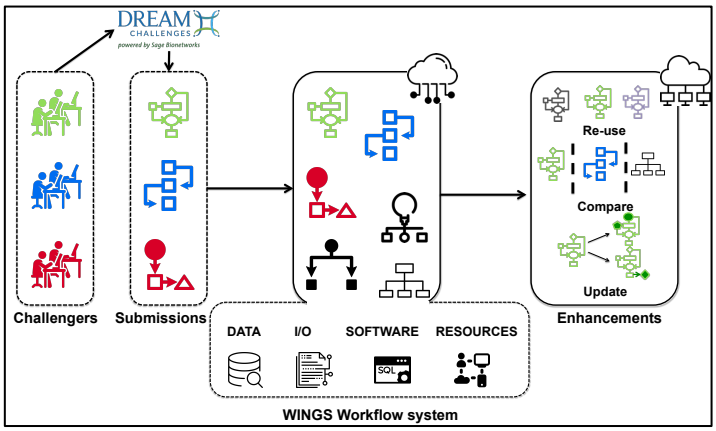


Fig. 1. Schematic for WINGS workflows in the context of data modeling and analysis competitions e.g. DREAM challenges. Building semantic workflows on the WINGS architecture enables widespread use of algorithms and methods, and enables storage and maintenance of data and workflows for use with high-throughput experiments.

Pegasus workflow management system[17] for distributed computing to reduce computational cost) to run workflows multiple times with new parameters or data (facilitating reusability).

Figure 1 shows a schematic of the use of WINGS for DREAM challenges. Integrating WINGS in current bioinformatics benchmarking challenges will support the reuse of the best performing solutions. Furthermore, it will expedite comparison between multiple different solutions, which potentially use similar constructs and tools, but differ in parameterizations that lead to significant result changes. This concludes to a better understanding of the underlying reasons that lead to a successful solution. Lastly, the extensive provenance records of all submitted solutions will greatly facilitate widespread use and adoption.

We discuss the WINGS design and the specifics of the workflow and environment construction in the sections below. Further, as proof of concept, we employ WINGS workflows to construct a full-scale pipeline for the NCI-CPTAC DREAM proteogenomic (protein prediction) challenge[18] that exhibits the main features of WINGS for reusability of workflows, reproducibility of results, and benchmarking of how results are impacted by subtle workflow variations. Lastly, we build multiple variations of the protein prediction workflow, altering different steps to illustrate how WINGS facilitates comparisons of different implementations of the workflow.

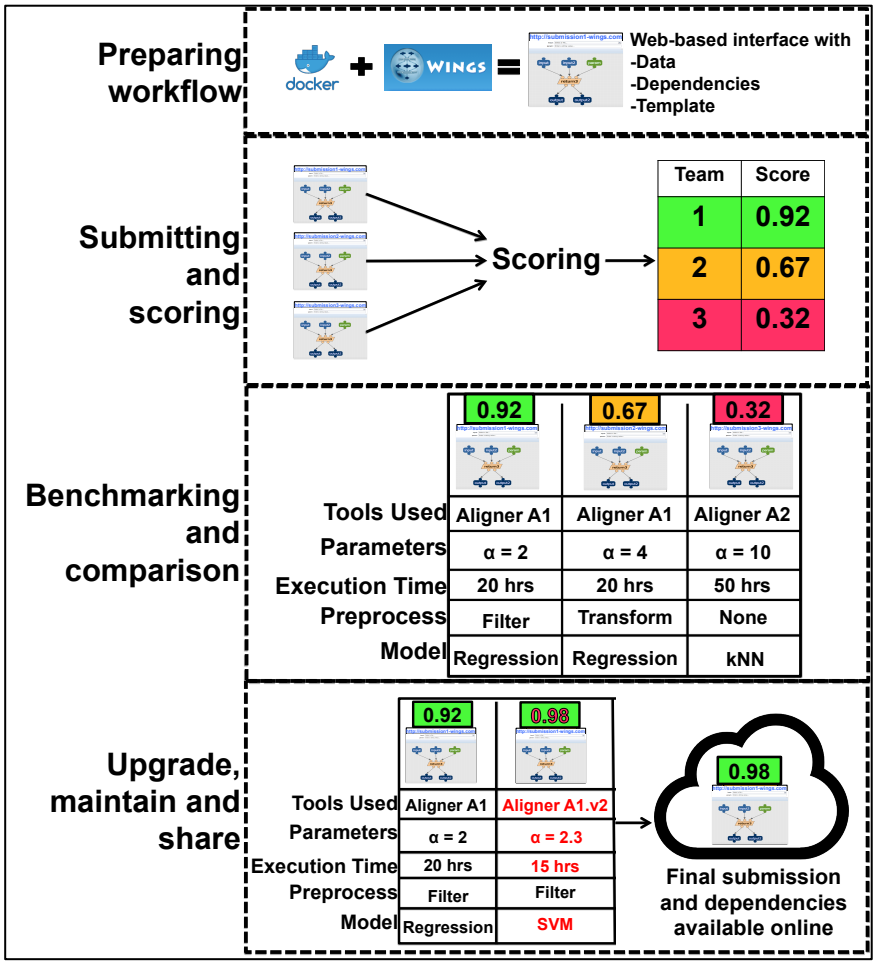


Fig. 2. Using WINGS in each phase of benchmarking challenges to facilitate benchmarking, reproducibility, and reusability.

2. Methods and Materials

The WINGS workflow system can be readily integrated with the existing work cycle of a benchmark challenge such as the DREAM challenges. **Figure 2** describes the typical phases of a benchmark challenge and how a system like WINGS could fit the process. Each section below defines these phases and how the integration of WINGS can

facilitate benchmarking, reproducibility, and reusability.

2.1. *Preparing and submitting workflows in WINGS for benchmark challenges*

The architecture and setup of WINGS (described in detail in the **supplementary materials**) facilitate easy usability and efficient sharing. A WINGS image, encapsulated by a Docker[16] container embedded with possible dependencies and software tools that may be needed by challengers to implement workflow steps, is built and made available at the onset of the challenge (**Figure 2**). New tools and software, as required by the codebase of each submission, can then be additionally included by the user within the WINGS framework where the submission pipeline is built.

WINGS facilitates the effective combination of utilities, scripts and tools based on different languages together under the umbrella of one single workflow, while allowing the user to see the high level view of the workflow steps in terms of the functions included within the workflow. **Figure 3** showcases the different components of a WINGS workflow. The main constructs involved are (1) *Components,* which encapsulate executable code described in terms of input data, parameters and outputs, each with unique datatypes and other semantic constraints (2) *Abstract components*, which can execute one of several codes with the same general functionality (e.g. an abstract component for normalization could be implemented by different normalization techniques, all employed on the same input, but resulting in different normalized data), (3) *Input parameters*, which may be string, integer, float, boolean or date values, (4) *Input files,* with metadata describing their type and contents, and (5) *Intermediate and final data*, which is output obtained from a component's execution that can be used as input to another component for further analysis.

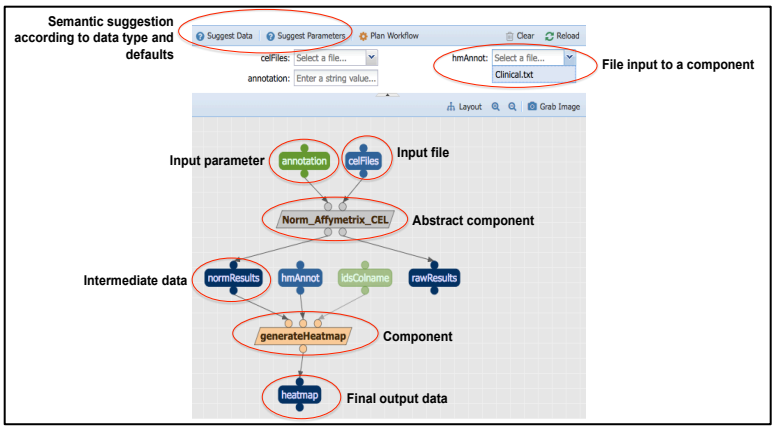


Fig. 3. Multiple components are connected in WINGS to design a workflow, as is typical of workflow systems. WINGS has unique features supported by semantic representations and reasoning: (a) automated suggestions of datasets and parameter values that are compatible with the current design of the workflow, (b) the possibility of defining abstract components that can be implemented by different tools.

Construction of a workflow in WINGS involves: (1) Creating data types and uploading raw input data, (2) Creating individual components for each distinct step in the workflow and supplying the code and scripts to generate outputs from inputs, (3) Connecting the components to reflect the flow of data from one to another. Additionally, the user can specify semantic metadata and validation rules to datasets, components, and workflows, which are used by WINGS to reason about the workflow and suggest data or parameters as well as to validate those provided by the

user. The details of building a workflow in WINGS, using standard RNA-Seq processing as an example, are included in the **supplementary materials**.

We used WINGS for the NCI-CPTAC DREAM proteogenomic challenge. We created a workflow for predicting protein levels from transcriptomics data, which includes the processing of transcriptomics data from raw sequencing reads to a normalized gene-expression matrix used for protein level prediction.

2.2. *Benchmarking, comparison, upgrade and sharing of workflows*

Benchmarking challenges, such as the DREAM challenges, have historically evaluated the performance of each challenger's submission and reported on the top performing approaches. With the integration of WINGS, all submitted entries would be described as WINGS workflows. Each step of the workflows would be encapsulated in self-contained modules. Thus, each submitted workflow and their steps, can be benchmarked and compared amongst one another. WINGS abstract components would prove especially useful for comparisons as a challenger's workflow component will house the execution machinery for their specific approach while maintaining the same input and output as the components designed by their peers. Additionally, benchmarking and comparison facilitates iteratively fine-tuning a bioinformatics workflow, as it allows for easy comparisons of different input parameters, files and software modules. A record of executed workflows, with the associated meta-data as maintained in WINGS, helps identify and correct errors as well as optimize a workflow.

We use the protein prediction pipeline template provided to DREAM proteogenomic challenge participants and construct 6 variations on the same workflow (using abstract components), enabling benchmarking and comparative analysis.

Different variations of the workflow are initially compared on the basis of the same performance metric used to evaluate the results of the DREAM proteogenomics challenge. This is a correctness score, which is the aggregated Pearson's correlation of predicted protein levels to actual protein levels across samples. To further our understanding of the comparison between workflow variations, we compare three scales of data amongst each workflow execution: aligned reads, quantified transcriptomics expression, and final protein level prediction. This allows us to understand the factors culminating in the resulting correctness score. Aligned reads are compared by read coverage areas of the resulting BAM files (comparison employs deeptools module "multibamsummary"[19]), quantified expression and predicted protein levels are compared by assessing sample and gene-wise Spearman correlation of transcript/protein levels. WINGS facilitates this step-by-step comparison by allowing intermediate outputs to act as input to components performing individualized comparison. Executing non-WINGS challenge entries to store and compare intermediate output is potentially cumbersome and prone to errors as we would need: (a) access to the complete pipeline of each participant, (b) detailed annotations within the subsequent code explaining each step of the pipeline, and (c) computational power and storage to execute multiple workflows and store each intermediate and final output.

Upon completion of a challenge, the best performing solutions can easily be maintained and upgraded within the confines of the WINGS system. Any tools and data utilized can be swapped

for latest versions. Additionally, utilizing the capabilities of containers ensures that the latest workflow and its ecosystem (dependencies and tools) can be encapsulated and shared with the community. The reusability of a workflow is not hampered by missing configurations, by lack of expertise to setup the computational environment, or by the absence of comprehensive descriptions of the pipeline itself.

3. Results

3.1. *WINGS workflow construction for the DREAM proteogenomic challenge*

As proof of concept for incorporating WINGS into a benchmark challenge, we built a workflow that performed protein level prediction from processed and normalized transcriptomics (RNA-Seq) data, mimicking the requirements of sub-challenge 2 of the NCI-CPTAC DREAM proteogenomic challenge 2018. Our workflow included the generation of a canonical transcriptomic expression matrix from raw reads allowing us to examine how sensitive the predictions were to changes at many phases of the workflow. Below we describe (**Figure 4**), (1) The entire workflow for protein level prediction from transcriptomics data and (2) The data and data types required to be uploaded and constructed in WINGS to facilitate workflow execution.

3.1.1. *The protein prediction workflow*

As our workflow aims to gauge protein levels for a set of samples from raw and unprocessed transcriptomics (RNA-Seq) data, it is divided to three distinct sections. (1) Alignment of raw read output from the sequencer, (2) Quantification and normalization per sample of aligned reads and lastly (3) Prediction of protein levels from processed and normalized transcriptomics data (**Figure 4**).

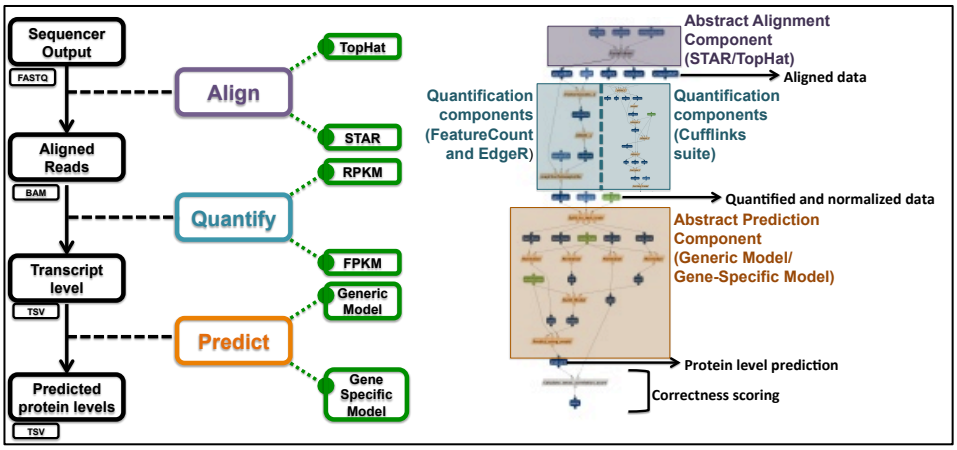


Fig. 4. The protein prediction workflow as implemented in WINGS. The black boxes show the workflow schematic in terms of input, intermediate and output files. Alignment (purple), quantification (blue) and prediction (orange) are the three main sections of the workflow. The green boxes represent the changes to tools and parameters that result in variation of this predictive pipeline, and subsequently different outputs. On the left is the WINGS wire diagram of the complete workflow, with annotations marking the three main steps.

3.1.2. *The data and data type categorization for a workflow*

Input, output and intermediate files that are produced by the workflow dictate data types within WINGS (**Figure 4**). For the protein prediction workflow, the input files – RNA-sequencer output (FASTQ format), the output files – protein level matrix (TSV format) and the intermediate files - aligned reads (amongst others) (BAM format) guide the different data types to be constructed by the user apropos to the workflow.

The data utilized for protein prediction is The Cancer Genome Atlas/Clinical Proteomic Tumor Analysis Consortium (TCGA/CPTAC)-Colorectal Cancer datasets[20,21], which is one of the foundational proteogenomics datasets published by the National Cancer Institute (NCI). The data consists of transcriptomics and proteomics for 89 patient samples that are processed, analyzed and well characterized by multiple published experiments[22]. The raw data is available from both TCGA and CPTAC, and the processed data was extracted from supplementary material of associated publications. The data is housed within the WINGS image, hosted on an Amazon Web Server (**supplementary material**), contained within the workflow ecosystem, along with all the tools and scripts needed by the pipeline.

3.2. *Workflow variations for predicting protein levels*

We select 3 specific changes to the protein prediction workflow, spanning the three levels of input data processing and compared the final result. We aimed to make changes at each level of data dimensionality to assess the impact on the final protein prediction. The changes are made to (1) Alignment tools, (2) Transcript level quantification method and (3) Protein level prediction method as is summarized in **Figure 4**.

Alignment Tools (STAR[23] versus TopHat[24]) – We utilize the two widely adopted alignment tools for comparison. STAR is a fast, reliable reads aligner which requires a large amount of computing power but claims to address most shortcomings of other RNA-Seq aligners. TopHat is a traditional splice read mapper for RNA-Seq, which uses the ultra high-throughput short read aligner Bowtie to perform read alignment followed by identification of splice junctions.

Transcript level quantification method (FPKM versus RPKM) –The two most popular methods to quantify transcripts level expression are Fragments Per Kilobase of transcript per Million mapped reads (FPKM) and Reads Per Kilobase of transcript, per Million mapped reads (RPKM). Both normalize according to gene length, RPKM utilizes reads whereas FPKM estimates abundance based on fragments observed in a paired end experiment. We utilize the cufflinks suite[3] (cufflinks, cuffmerge, cuffquant and cuffnorm) to assess the FPKM quantification and featureCounts[25] with the EdgeR[26] R package to obtain the RPKM quantification.

Prediction method (Generic-Linear versus Gene-Specific) – The winners of the DREAM proteogenomic challenge employed multiple different models and one of the superior results was obtained by employing a Gene-Specific modeling technique for prediction[27]. Within our workflow, we aim to emulate their technique by building a unique linear model for each of the proteins to be predicted (Gene-Specific) and compare it against a one-fits-all linear model (Generic-Linear) that uses the entirety of the training data irrespective of gene and site specificity.

3.3. *Benchmarking and correctness of protein prediction across workflow variations*

As detailed above, a total of 6 different variations of the protein prediction workflow were executed using WINGS. Workflow variations included changes to the 3 distinct sections of the protein prediction workflow, namely alignment, quantification and prediction. **Table 1** summarizes the correctness (of prediction) score of the final result obtained from each variant of the workflow. We also note the approximate time (automatically recorded for each WINGS workflow execution) taken for each workflow completion. We observe the differences in quality of results based on the changes in different steps and dimensions of the prediction workflow. Namely, the largest change in resulting quality emanated from the different models used for prediction. The gene-specific model outperformed the generic linear model in all configurations. The alignment and quantification presented some minute changes in the final result quality but large differences in computational resource utilization, as the execution time was vastly different between STAR and TopHat usage, as well as evaluation of RPKM and FPKM.

Table 1. Pearson correlation based correctness score, and time taken for execution of each workflow configuration for protein level prediction of 89 samples and ~3000 proteins

Alignment	Quantification	Predictive Model	Correctness Score	Time Taken
STAR	FPKM	Linear	0.2161	~29 hrs
STAR	RPKM	Linear	0.2155	~20 hrs
STAR	FPKM	Gene-Specific	0.9064	~29 hrs
STAR	RPKM	Gene-Specific	**0.9124**	~20 hrs
TopHat	RPKM	Linear	0.2053	~103 hrs
TopHat	RPKM	Gene-Specific	0.9080	~103 hrs

3.4. *Comparison of workflow variations for predicting protein level*

Since intermediate output at each level is readily available in the WINGS provenance records, we explore each of the workflow variations at 3 different scales. Namely, we compare the aligned reads, the transcript quantitation and finally the predicted protein levels. **Figure 5** shows the WINGS workflow and the corresponding output for comparing aligned reads (BAM files). The component uses the utilities described in the section above to calculate the

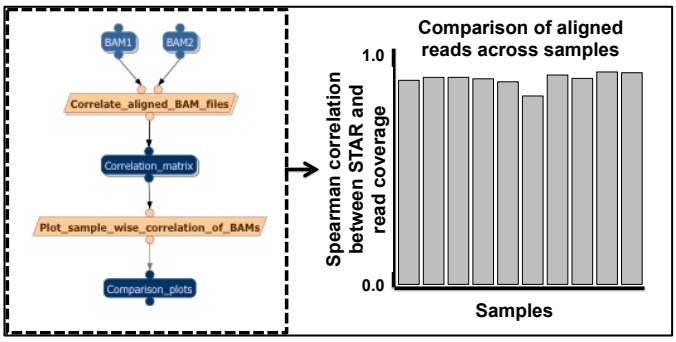


Fig. 5. Correlation between TopHat and STAR aligned reads across 10 samples (right) from the protein prediction workflow in WINGS (right).

correlation between read coverage for aligned reads obtained from both TopHat and STAR. **Figure 6** presents the component performing comparison of transcript quantification utilizing both FPKM and RPKM methodologies. The output visualizes a comprehensive comparison of both quantifications, by assessing the number of genes identified, gene and sample wise correlation and dynamic ranges of the gene-level expression.

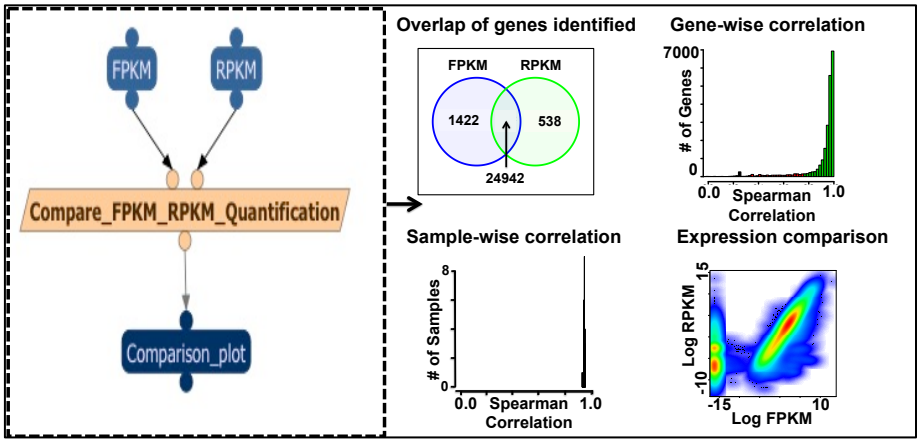


Fig. 6. Comparison between FPKM and RPKM transcript quantification obtained from the protein prediction workflow and the corresponding WINGS component utilized. Includes (Top Left) Overlap of genes identified using both the quantification methods, (Top Right) Gene-Wise expression correlation, (Bottom Left) Sample-wise expression correlation and (Bottom Right) Scatterplot of the entire quantification from both methods.

Lastly, **Figure 7** compares the final protein level prediction for two different models (Gene-Specific and Linear), as described in the section above. We show the component performing as well as visualizing the comparative analysis. Results include distribution comparison of predictions from both models and present correlation and dynamic ranges for both sets of predicted protein abundance. Changes to each step of a sequential workflow propagate downstream to alter the culminating output. The detailed analysis possible within the confines of WINGS allows us to fully understand the impact of each step's process on the final result of the protein prediction workflow. Further, since all intermediate data is accessible for each execution, data analysis and exploration can be performed in parallel at each step, including quality metrics, sanity checks and identifying critical data attributes characterizing inner workings of the pipeline. WINGS components performing analysis and exploration could be appended to the main workflow where they access intermediate data and provide immediate context to the workflow execution.

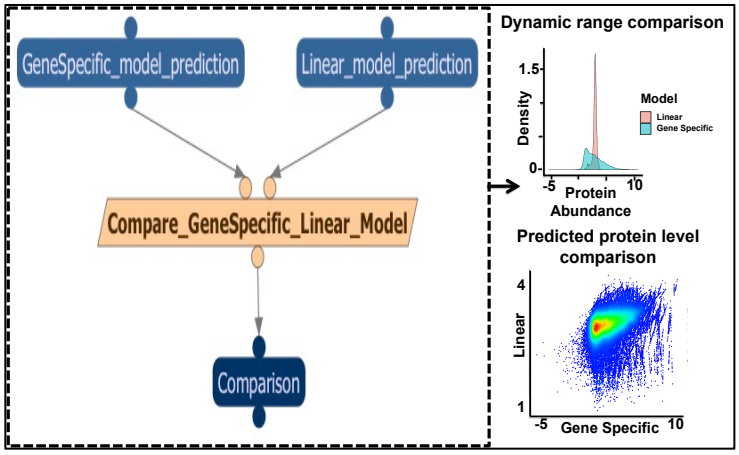


Fig. 7. Comparison between Gene-Specific and Linear modeling results obtained from the protein prediction workflow and the corresponding WINGS component utilized. Includes (Top) distribution comparison between the predicted protein levels from using each model for the 27 test samples (Bottom) Scatterplot comparison of each predicted protein level by both models for the 27 test samples.

4. Discussion and Conclusion

Our work presents the WINGS workflow infrastructure as an easy to use, effective and efficient platform for storing, maintaining and executing solutions submitted to analytical and modeling challenges. WINGS not only allows for standardization of submissions and effective reuse of workflows, it also allows for intuitive comparison between workflows as well as potential for changes and upgrades to ensure widespread adoption and rigorous reproducibility. As a proof of concept, we developed a protein prediction workflow using WINGS, akin to the DREAM proteogenomic challenge, which uses raw RNA-sequencing data as input, processing and modeling it to generate prediction for protein levels. WINGS houses the input data, performs benchmarking with different tools, techniques and models to identify the most effective configuration for protein prediction. In addition, for each variation of the workflow, we are able to identify and isolate critical changes in data across different steps as well as explore the nuances of the predictive model. Our experiments show the vast capability of WINGS and its usefulness to future bioinformatics analysis and modeling challenges. Additionally, incorporation of the WINGS paradigm in the context of data modeling and analytical challenges sheds light on a broader question of why a solution performs better than another. Constructing workflows with WINGS allows for researchers to use the most innovative methods by easily reusing the best performing approaches available for any given research question.

Supplementary material available at: https://github.com/arunima2/Supplementary_PSB_2019

References

1. Marx V. Biology: The big challenges of big data. *Nature*. 2013. doi:10.1038/498255a.
2. Stephens ZD, Lee SY, Faghri F, et al. Big data: Astronomical or genomical? *PLoS Biol*. 2015. doi:10.1371/journal.pbio.1002195.
3. Trapnell C, Roberts A, Goff L, et al. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nat Protoc*. 2012. doi:10.1038/nprot.2012.016.
4. Causey JL, Ashby C, Walker K, et al. DNAp: A Pipeline for DNA-seq Data Analysis. *Sci Rep*. 2018;8(1):6793. doi:10.1038/s41598-018-25022-6.
5. DREAM Challenges. http://dreamchallenges.org/.
6. Protein Structure Prediction Center. http://predictioncenter.org/.
7. Proteome Informatics Research Group (iPRG). https://abrf.org/research-group/proteome-informatics-research-group-iprg.
8. Moult J, Fidelis K, Kryshtafovych A, Schwede T, Tramontano A. Critical assessment of methods of protein structure prediction (CASP)-Round XII. *Proteins Struct Funct Bioinforma*. 2018;86:7-15. doi:10.1002/prot.25415.
9. Ewing AD, Houlahan KE, Hu Y, et al. Combining tumor genome simulation with crowdsourcing to benchmark somatic single-nucleotide-variant detection. *Nat Methods*. 2015. doi:10.1038/nmeth.3407.
10. Araújo T, Aresta G, Castro E, et al. Classification of breast cancer histology images using Convolutional Neural Networks. Sapino A, ed. *PLoS One*. 2017;12(6):e0177544. doi:10.1371/journal.pone.0177544.

11. Afgan E, Baker D, Batut B, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic Acids Res*. 2018;46(W1):W537-W544. doi:10.1093/nar/gky379.
12. Reich M, Liefeld T, Gould J, Lerner J, Tamayo P, Mesirov JP. GenePattern 2.0. *Nat Genet*. 2006;38(5):500-501. doi:10.1038/ng0506-500.
13. Gil Y, Ratnakar V, Kim J, et al. Wings: Intelligent Workflow-Based Design of Computational Experiments. *IEEE Intell Syst*. 2011;26(1).
14. Zheng CL, Ratnakar V, Gil Y, McWeeney SK. Use of semantic workflows to enhance transparency and reproducibility in clinical omics. *Genome Med*. 2015. doi:10.1186/s13073-015-0202-y.
15. Missier P, Belhajjame K, Cheney J. The W3C PROV family of specifications for modelling provenance metadata. *Proc 16th Int Conf Extending Database Technol - EDBT '13*. 2013. doi:10.1145/2452376.2452478.
16. Merkel D. Docker: lightweight Linux containers for consistent development and deployment. *Linux J*. 2014. doi:10.1097/01.NND.0000320699.47006.a3.
17. Gil Y, Ratnakar V, Deelman E, Mehta G, Kim J. Wings for Pegasus: Creating Large-Scale Scientific Applications Using Semantic Representations of Computational Workflows. *Proc Twenty-Second Natl Conf Artif Intell*. 2007:1767-1774.
18. NCI-CPTAC DREAM Proteogenomics Challenge. https://www.synapse.org/#!Synapse:syn8228304/wiki/413428.
19. Ramírez F, Dündar F, Diehl S, Grüning BA, Manke T. DeepTools: A flexible platform for exploring deep-sequencing data. *Nucleic Acids Res*. 2014. doi:10.1093/nar/gku365.
20. Tomczak K, Czerwińska P, Wiznerowicz M. The Cancer Genome Atlas (TCGA): An immeasurable source of knowledge. *Wspolczesna Onkol*. 2015;1A:A68-A77. doi:10.5114/wo.2014.47136.
21. Whiteaker JR, Halusa GN, Hoofnagle AN, et al. CPTAC Assay Portal: a repository of targeted proteomic assays. *Nat Methods*. 2014;11(7):703-704. doi:10.1038/nmeth.3002.
22. Zhang B, Wang J, Wang X, et al. Proteogenomic characterization of human colon and rectal cancer. *Nature*. 2014. doi:10.1038/nature13438.
23. Dobin A, Davis CA, Schlesinger F, et al. STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics*. 2013. doi:10.1093/bioinformatics/bts635.
24. Kim D, Pertea G, Trapnell C, Pimentel H, Kelley R, Salzberg SL. TopHat2: Accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biol*. 2013. doi:10.1186/gb-2013-14-4-r36.
25. Liao Y, Smyth GK, Shi W. FeatureCounts: An efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics*. 2014. doi:10.1093/bioinformatics/btt656.
26. Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*. 2010. doi:10.1093/bioinformatics/btp616.
27. Li H, Guan Y. Guanlab's solution to the 2018 NCI-CPTAC DREAM Proteogenomics Challenge. https://www.synapse.org/#!Synapse:syn11522015/wiki/496744.

Precision Medicine: Improving health through high-resolution analysis of personal data

Steven E. Brenner†
University of California, Berkeley

Jill P. Mesirov
University of California, San Diego

Martha Bulyk
Brigham & Women's Hospital and Harvard Medical School

Alexander A. Morgan
Khosla Ventures

Dana C. Crawford
Case Western Reserve University

Predrag Radivojac
Indiana University

For the 2019 Pacific Symposium on Biocomputing's session on precision medicine, we present new research on computational techniques in range of areas including data curation, whole genome analysis, transcriptomics, microbiome profiling, EHR data-mining, and histological image processing.

Keywords: genomics, trancriptomics, personalized medicine, precision medicine

1. Introduction

For this session we sought submissions that drive research forward in the development of techniques in high resolution data science to advance personalization in clinical care based on quantified models. The roots of using data to improve healthcare and to personalize medicine are ancient and run deep in medicine. Hippocrates recommended that physicians learn to read so they could keep records and learn how to treat new patients by studying the case histories their colleagues compiled. James Lind, a naval surgeon, performed the first controlled clinical trial of a therapeutic intervention in 1747, with a multi-arm study of six different possible interventions for scurvy. William Osler (1849-1919), originator of the modern system of training physicians, said "The good physician treats the disease, the great physician treats the patient who has the disease." However, it has only been within the last few decades that we have had the tools to change the approach to understanding a patient from a somewhat subjective art to a deeply quantified science. We have advanced rapidly in molecular profiling from expensive single genomes to increasingly low cost genomic, transcriptomic, and proteomic profiling of single human cells. The massive switch to electronic health records, including the rise of large volumes of electronic imaging data in such forms as CT and MRI, has created huge volumes of computationally tractable data within the healthcare system. With an ever increasingly connected world, biosensors and mobile health tracking devices are providing new streams of phenotypic data. Inspired by the very earliest efforts in pushing medicine toward being a system of constant improvement and innovation based on data and experimentation (planned and naturally occurring), data is being collected in ever larger

† Supported by U41 HG007346 and U19 HD077627

volumes. However, we increasingly need innovation in bioinformatic techniques that help organize this data, discern the multi-omic characterization of disease, elucidate pathophysiology at the level of cells and tissues, and create actionable insights for points of intervention. The papers in this session span this gamut, and we hope will help drive the field from being not only precise, but also accurate in promoting the health and wellbeing improvements that can have widespread impact.

2. Session Papers

2.1. *Data Curation Tools and Techniques*

Peyton Greenside and colleagues [1] have developed a tool, CrowdVariant, based on Google's crowdsourcing platform to allow non-experts to annotate genomic data. They demonstrate with data from the Genome In A Bottle Consortium that the general public can be quickly trained to annotate deletions as a proof of concept. As the authors note, the images derived from genomic data, such as NGS read alignment create visual patterns that non-experts can be quickly trained to identify and interpret, opening up plenty of opportunity for future efforts to leverage the "wisdom of the crowd" in the expensive task of genome annotation, and potentially other forms of biomedical data.

Moving from the human crowd to the internal crowd of microbial flora, Wontack Han and Yuzhen Ye [2] have developed a repository of microbial marker genes and a set of tools to link microbial markers with human host phenotype, with an initial focus on diabetes, liver cirrhosis, and cancer. Their computational pipeline, Mi2P (Microbiome to Phenotype) is a publicly available project in Sourceforge.[a]

Another project helping to manage data related to precision medicine is the work of Zhiyue Tom Hu and colleagues [3], where they describe a framework for addressing inconsistency in large pharmacogenomic data sets, where individual potential therapeutics are screened against cancer cell lines. The method, Alternating Imputation and Correction Method (AICM), uses shared overlap of a handful of tested medications to bring divergent datasets into alignment for comparison across the full span of data. They show the validity of this approach with three large pharmacogenomic datasets.

2.2. *Techniques in Probing Complex Genome-Phenome Interactions*

Autism is a complex phenotype, with a strongly heritable component little explained by known genetic variants. Maya Varma and colleagues [4] have made creative use of a creative control group (progressive supranuclear palsy) to probe the genomic dark matter of non-coding regions to identify a set of genetic markers associated with autism. Despite significant work to remove

[a] https://sourceforge.net/projects/mi2p/

potential batch effects, they are able to achieve very strong classification accuracy (0.96 AUC) based on genetic features for identifying autism cases, suggesting the features they have identified in non-coding regions may be causal in ways that we have yet to identify.

Xinyuan Zhang and colleagues [5] tackle a different kind of complexity, as they look for pleiotropy in cardiovascular and neurological diseases in a dataset of 530,000 SNPs coupled with phenotypes extracted from EHR data for 43,870 individuals from the eMERGE network. Genes certainly play different roles in different contexts, such as different tissue types, different environmental stimuli, and different life histories; however, pleiotropy has been hard to detect in prior studies, due to a mix of factors including small datasets barely powered to find even simple single variant-phenotype interactions and poor phenotypic characterization. Here, leveraging a large dataset and the rich clinical annotations, they present a framework mixing a range of approaches to detect pleiotropy.

2.3. *Molecular Biology of the Tissues*

The natural extension for precision medicine discovery from the genome is moving into functional data and specifically gene expression. However, gene expression is very context specific, as noted in the work in this session. Derek Reiman and colleagues [6] look at the relationship between histopathology and gene expression in cancer, with a special focus on immune infiltration in the tumor micro-environment, of potential relevance to immune therapies in oncology. Applying a neural net based approach, they show that integrating features derived from digital surgical pathology imaging and RNA-Seq can automatically predict infiltration of the tumor by NK cells, macrophages, and CD8+ T-cells.

Binglan Li and colleagues [7] also focus on gene expression, and did tissue specific transcriptome wide association studies on clinical phenotypes in set of 4,360 individuals in an AIDS clinical trial, leveraging data on the context specificity of gene expression and eQTL's from the GTEx (Genotype Tissue Expression Project). This work has a poster at the conference and a paper in the proceedings.

2.4. *Creating Actionable Insights*

Precision medicine is about moving beyond just discovery to changing clinical practice with precise, personalized data. This session includes two pieces of work in this direction. The first is similar in direction with the previously mentioned work in that it focuses on eQTLs and gene expression regulatory relationships, but its focus is on therapeutic discovery and drug repositioning. Francesca Vitali and colleagues [8] use a network biology and semantic similarity approach to look for putative shared functional relationships between diseases to propose opportunities for drug repurposing.

Rounding out our session is work positioned to directly make recommendations around care decisions, particularly around the problem of when to order lab tests for critically ill patients. Patients in the ICU can have rapidly worsening clinical status, and blood-based diagnostic testing can help detect early signs of dangerous conditions such as sepsis or kidney failure. However, testing is not free, both in actual expense, but also patients do not have an infinite blood volume. Although patients in the ICU can have continuous venous access, in the general case, a blood draw is a form of invasive procedure, with discomfort and some risk involved. Li-Fang Cheng and colleagues [9] have developed a reinforcement learning framework to train a system for an optimal testing policy. This type of approach can both reduce unnecessary lab testing, but also suggests testing earlier than is currently done, in advance of critical events, ideally enabling early intervention to prevent poor outcomes.

References

1. Greenside, Peyton, et al. "CrowdVariant: a crowdsourcing approach to classify copy number variants," *Pac Symp Biocomput, 2019.*
2. Ye, Yuzhen and Han, Wontack, "A repository of microbial marker genes related to human health and diseases for host phenotype prediction using microbiome data," *Pac Symp Biocomput, 2019.*
3. Hu, Zhiyue Tom, et al. "AICM: A Genuine Framework for Correcting Inconsistency Between Large Pharmacogenomics Datasets," *Pac Symp Biocomput, 2019.*
4. Varma, Maya, et al. "Machine Learning Approach Identifies Single Nucleotide Variants in Noncoding DNA Associated with Autism Spectrum Disorder," *Pac Symp Biocomput, 2019.*
5. Zhang, Xinyuan, et al. "Detecting pleiotropy across cardiovascular and neurological diseases using univariate, bivariate and multivariate methods on 43,870 individuals from the eMERGE network," *Pac Symp Biocomput, 2019.*
6. Reiman, Derek, et al. "Integrating RNA expression and visual features for immune infiltrate prediction," *Pac Symp Biocomput, 2019.*
7. Li, Binglan, et al. "Influence of tissue context on gene prioritization for predicted transcriptome-wide association studies," *Pac Symp Biocomput, 2019.*
8. Vitali, Francesco, et al. "Precision drug repurposing via convergent eQTL-based molecules and pathway targeting independent disease-associated polymorphisms," *Pac Symp Biocomput, 2019.*
9. Cheng, Li-Fang, et al. "Learning an Optimal Policy for Ordering Patient Laboratory Tests in Intensive Care Units," *Pac Symp Biocomput, 2019.*

CrowdVariant: a crowdsourcing approach to classify copy number variants

Peyton Greenside

Biomedical Informatics, Stanford University,
Stanford, CA 94305, USA
pgreens@stanford.edu

Justin Zook and Marc Salit

National Institute of Standards and Technologies (NIST),
Material Measurement Laboratory, 100 Bureau Dr., Gaithersburg, MD 20899 USA
justin.zook@nist.gov, salit@nist.gov

Madeleine Cule

Verily Life Sciences, Calico,
269 E Grand Ave, South San Francisco, CA 94080 USA
cule@calicolabs.com

Ryan Poplin, Mark DePristo*

Google Brain, Verily Life Sciences,
1600 Amphitheatre Parkway, Mountain View, CA USA
rpoplin@google.com, mdepristo@google.com
**Corresponding Author*

Copy number variants (CNVs) are an important type of genetic variation that play a causal role in many diseases. The ability to identify high quality CNVs is of substantial clinical relevance. However, CNVs are notoriously difficult to identify accurately from array-based methods and next-generation sequencing (NGS) data, particularly for small ($<$ 10kbp) CNVs. Manual curation by experts widely remains the gold standard but cannot scale with the pace of sequencing, particularly in fast-growing clinical applications. We present the first proof-of-principle study demonstrating high throughput manual curation of putative CNVs by non-experts. We developed a crowdsourcing framework, called CrowdVariant, that leverages Google's high-throughput crowdsourcing platform to create a high confidence set of deletions for NA24385 (NIST HG002/RM 8391), an Ashkenazim reference sample developed in partnership with the Genome In A Bottle (GIAB) Consortium. We show that non-experts tend to agree both with each other and with experts on putative CNVs. We show that crowdsourced non-expert classifications can be used to accurately assign copy number status to putative CNV calls and identify 1,781 high confidence deletions in a reference sample. Multiple lines of evidence suggest these calls are a substantial improvement over existing CNV callsets and can also be useful in benchmarking and improving CNV calling algorithms. Our crowdsourcing methodology takes the first step toward showing the clinical potential for manual curation of CNVs at scale and can further guide other crowdsourcing genomics applications.

Keywords: copy number variation, precision medicine, crowdsourcing

1. Introduction

Copy number variation is a type of structural variation that involves large-scale duplications or deletions of parts of a chromosome. Copy number variants can have substantial effects on cell and organism phenotype and are associated with many kinds of human disease (Redon et al., 2006) (Feuk, Carson, & Scherer, 2006) (Sudmant et al., 2015). Identifying CNVs is an important component of clinical pipelines for assessing genetic mutations that contribute to disease progression. Numerous algorithms have been developed to characterize these variants from genotyping arrays and next-generation sequencing data (English et al., 2015) (Tattini, D'Aurizio, & Magi, 2015) (Mills et al., 2011) (Kidd et al., 2008). However, these algorithms often have poor concordance on both the location and the type of copy number variant, particularly for small-scale (< 10kbp) CNVs (Scherer et al., 2007) (Pinto et al., 2011), leading experts to rely heavily on manual curation. One key challenge in further developing and assessing these algorithms is the lack of a large set of "gold standard" or reference copy number variants.

Crowdsourcing has been used successfully to obtain gold standard labels in projects such as Galaxy Zoo (Raddick et al., 2010), ClickWorkers (Ishikawa, ST and Gulick, 2012), FoldIt (Cooper et al., 2010), and Zooniverse (Prather et al., 2013), but little investigation has been done to understand how crowdsourcing can be best utilized to analyze genomic variation (Haghighi et al., 2018). Basic questions include whether or not any domain expertise is truly needed, how large the crowd should be, and how to best train and display genetic variation to workers. We investigated the use of crowdsourcing platforms to classify copy number variants, focusing on deletions, and to address these basic questions. Google has developed the Crowd Compute platform to facilitate large-scale crowdsourcing problems, and we developed our framework with this platform to enable high throughput classifications. In this work we show proof of principle in a well characterized reference genome, an essential first step before deploying the method on more variable genomes such as from clinical samples. In a similar vein, we focus on deletions as the most frequent and also likely easiest to classify type of structural variation before focusing on more complex applications. CrowdVariant can be used to develop high confidence CNV sets, to benchmark new CNV detection algorithms, and to enable high throughput manual curation of CNVs using both experts and non-experts.

2. Results

2.1. *The CrowdVariant Framework*

The CrowdVariant framework uses a crowdsourcing platform to display putative copy number variant sites to workers and aggregates classifications from a pool of workers to determine the copy number state. Using this framework, we first ran an experiment to compare non-expert and expert classifications on a pilot set of putative CNV sites and then expanded our classi-

fications to curate a genome-wide set of high confidence CNVs [Figure 1].

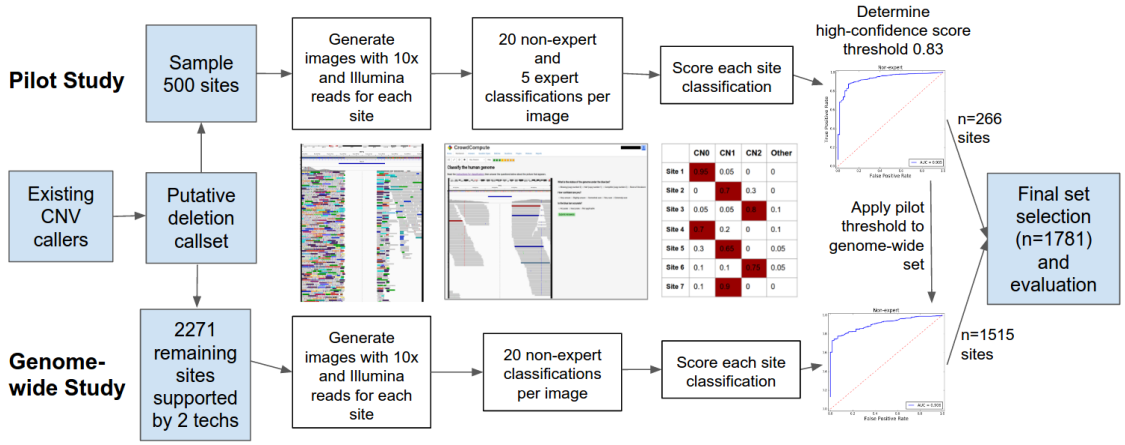


Fig. 1. The experimental design was constructed to first evaluate a pilot set of sites with both experts and non-experts before applying the same framework to a genome-wide set of sites using non-experts only.

CrowdVariant displays pileup images of putative copy number variant sites using the Integrative Genomics Viewer (IGV), showing all reads aligned to the site and the flanking regions [Supplementary Figure 1] (Thorvaldsdóttir, Robinson, & Mesirov, 2013). Workers classify the site, assess break point accuracy and report their confidence based on seeing one image at a time.

We selected a set of 500 putative deletion sites for the pilot phase of our study. We first called putative sites using an ensemble approach from multiple sequencing technologies (Illumina, PacBio, Complete Genomics and BioNano) and corresponding algorithms (see Supplementary Methods for details) (Abyzov, Urban, Snyder, & Gerstein, 2011) (Garrison & Marth, 2012) (Mohiyuddin et al., 2015) (Hormozdiari, Hajirasouliha, McPherson, Eichler, & Sahinalp, 2011) (Iqbal, Caccamo, Turner, Flicek, & McVean, 2012) (Mak et al., 2016) (Chaisson et al., 2014) (Nattestad & Schatz, 2016) (Drmanac et al., 2010). We then randomly selected from all putative sites 500 pilot sites ranging from 100bp to 3000bp with varying levels of support from existing algorithms [Supplementary Table 1].

We used aligned 10X Genomics (10X) and Illumina paired-end (Illumina) reads from the reference Ashkenazim trio made available by the Genome In A Bottle (GIAB) Consortium (Zook et al., 2016). For each putative copy number variant site, we generated an image for each member of the trio (son/mother/father) using Illumina reads, one image for the son's diploid reads and one image for each haplotype of the son's reads using 10X reads. Although workers potentially saw multiple images of the same site, we did not disclose to workers the experimental design, the sequencing technology, the individual or the site being shown in an

effort to most fairly compare experts and non-experts.

In our pilot study, 20 non-experts each classified all 6 images for the 500 pilot sites. We launched a global recruitment for self-reported experts curators with over 110 individuals from several dozen institutions signing up to classify variants. The participation rate was highly variable with an average of 76 questions per expert [Supplementary Figure 2]. We ensured that all 6 images for at least 100 sites were classified by 5 experts each.

2.2. *Non-experts can curate high quality copy number variants*

Both experts and non-experts agreed on a consensus classification for the majority of sites [Supplementary Figure 3]. We visualized the responses for non-experts [Figure 2] and experts [Figure 3] by weighting each copy number classification and clustering workers and sites to reveal performance differences across sequencing platforms and individuals. We kept the identity of each non-expert worker separate, but we merged the expert answers into artificial workers 1 through 5 as experts did not answer enough questions individually to be meaningfully compared. For 86% of images, at least 70% of non-expert workers agreed on the classification, showing that non-experts can be trained to interpret copy number variants in a consistent manner [Supplementary Table 2]. Non-experts primarily had difficulty classifying haplotype images and systematically confused CN2s as CN1s for haplotype images only (see Fig. 8 haplotype heatmaps). Beyond these systematic errors, there were several non-experts that deviated from the majority either from lack of effort or understanding. Improving the documentation by showing more than 2 examples of each copy number type could further improve non-expert performance.

Agreement among workers was used to assign a final classification and confidence score to each putative site. We defined the CrowdVariant score as the proportion of workers that voted in favor of the most popular classification (CN0/CN1/CN2/None of the Above), with higher scores reflecting more confident classifications. We incorporated worker classifications for all images of the same site, but classified each site for each individual in the trio independently. We counted all diploid classifications but only those haploid classifications where the pair of haplotype images was consistent with a diploid classification [Supplementary Methods]. We assign the most likely copy number state to each site by selecting the classification with the largest proportion of votes.

Non-experts performed similarly to experts when comparing the rate of Mendelian violations among the trio (classifications that would not plausible from Mendelian inheritance) for each site [Supplementary Methods] [Table 1]. We found that 89% and 90% of all sites were classified without a Mendelian violation for experts and non-experts, respectively. The sites with Mendelian violations had lower scores and could largely be filtered out of the high quality set. The CrowdVariant scores discriminated Mendelian violations from genetically plausible trio classifications with an AUC of 0.89 for non-experts and an AUC of 0.87 for experts [Supplementary Methods] [Supplementary Figure 4]. For comparison, we randomized all answers

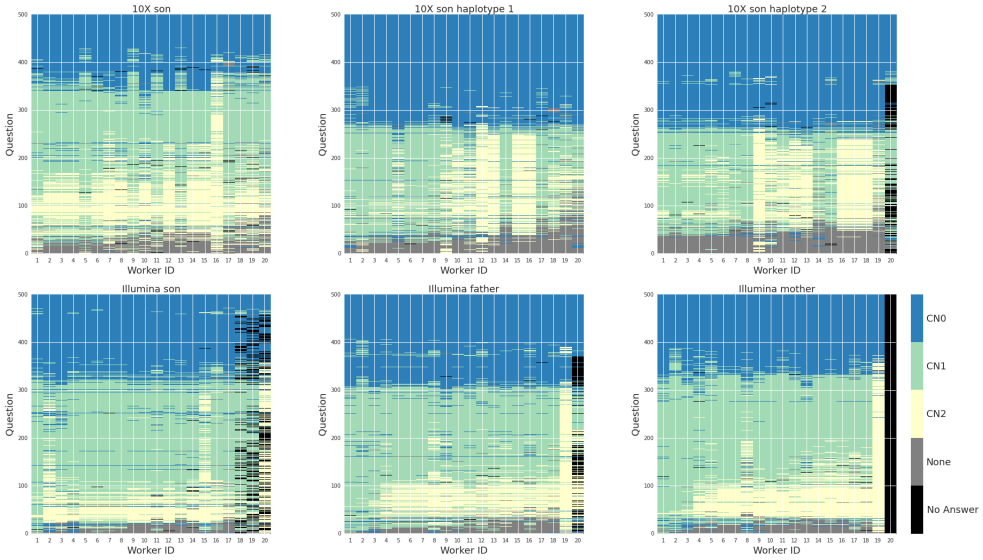


Fig. 2. Non-expert classifications for 500 sites were color coded, weighted and clustered (see Supp. Methods for details). Rows represent a question (i.e. an image of a putative site using a particular sequencing technology) and columns represent workers. Clockwise from top left: 10X son, 10X son haplotype 1 only, 10X son haplotype 2 only, Illumina mother, Illumina father, Illumina son.

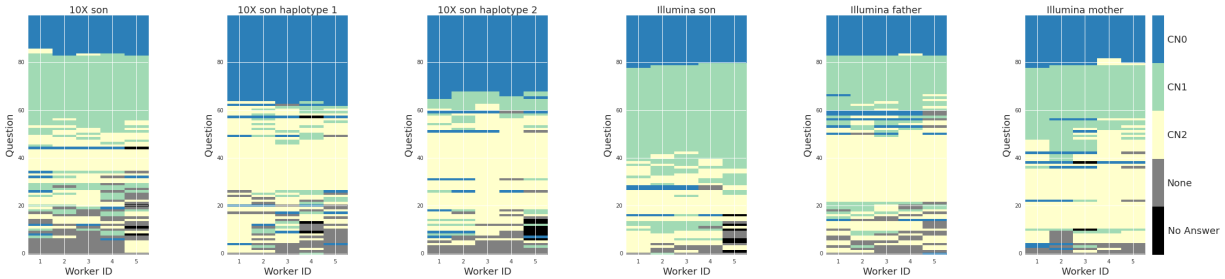


Fig. 3. Expert classifications for 100 sites were color coded, weighted and clustered (see Supp. Methods for details). Rows represent a question (i.e. an image of a site using a particular sequencing technology) and columns represent workers. Left to right: 10X son, 10X son haplotype 1 only, 10X son haplotype 2 only, Illumina son, Illumina father, Illumina mother.

by re-sampling the entire worker by classification matrices for experts and non-experts and re-computed the rate of Mendelian violations [Supplementary Table 3]. The AUCs for expert and non-expert randomized answers were 0.47 and 0.50, respectively, and both 95% confidence intervals overlapped a random AUC of 0.5.

We curated a high confidence set of CNVs for the son (NA24385) with high probability of correctness and no Mendelian violations [Supplementary Materials]. We initially intended to use self-reported confidence to filter lower quality classifications, but most non-experts consistently reported medium to high confidence despite minimal training [Supplementary Figure 5]. To avoid relying on self-reported confidence, we ranked all 500 sites by their CrowdVariant score and selected all sites with a higher score than the site with the first Mendelian violation.

Metric \ Data Set	Expert	Non-expert
Percent of sites without violation	89/100 (89%)	448/500 (90%)
ROC AUC	0.87	0.89
ROC AUC 95% confidence interval	[0.79, 0.95]	[0.86, 0.92]
Average violation probability	0.15	0.14

This violation occurred at score 0.83 and resulted in discarding approximately half of the sites for a total of 266 high confidence sites. The high confidence set of sites contains 122 CN0, 138 CN1, 5 CN2 and 1 "None of the above" classification. 252 out of 266 are supported by at least two other technologies. Importantly, for all sites in the high quality set that were classified by both experts and non-experts, there was 100% agreement (n=56 sites) between experts and non-experts.

2.3. *CrowdVariant can classify CNVs with variable support or unclear breakpoints*

CrowdVariant agrees with consensus classifications from existing algorithms, while also classifying variants that are challenging for existing algorithms. CrowdVariant scores assigned to each site are correlated with the number of technologies underlying the original calls [Figure 4]. CrowdVariant classifications also show strong agreement with svviz (Spies, Zook, Salit, & Sidow, 2015), a semi-automated visualization tool that determines whether each read supports the reference allele, alternate allele, or is ambiguous. We used a preliminary heuristic method to classify copy number variants based on the read counts supporting the reference and alternate alleles as determined by svviz for each dataset, and required agreement across all datasets that had clear support for a genotype [Supplementary Methods]. When comparing all high confidence classifications, agreement with svviz was 82%. CrowdVariant was able to resolve 26 sites that were uncertain for svviz, explaining part of the discrepancy. When we removed sites that were classified as "None of the Above" in CrowdVariant or uncertain in svviz, agreement was 91% between the two methods. Agreement with svviz also increased with the number of supporting technologies [Figure 5].

The true power of incorporating many data types is clear when all 6 images of the same site are viewed together [Figure 6]. We find in multiple cases the crowd is able to resolve copy number state where other methods cannot, particularly when the boundary points are incorrect or ambiguous [Figure 7, Supplementary Figure 7]. While non-experts make some mistakes, we find that they do so in a consistent manner, such as mistaking a difficult-to-sequence region for a deletion, and they could likely be trained to recognize other features in the image that would clarify these mistakes. Phased data is particularly powerful for classifying heterozygous CNVs that are otherwise ambiguous and provides visual confirmation of the CrowdVariant

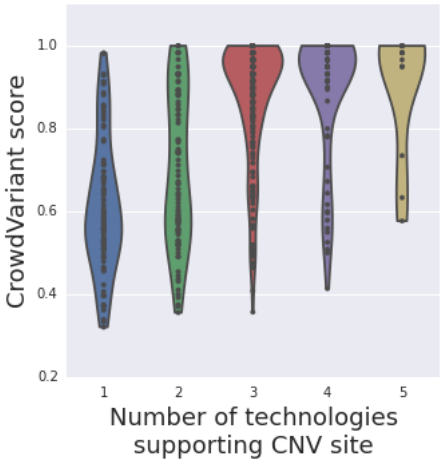


Fig. 4. CrowdVariant scores determined by non-expert workers stratified by the number of supporting technologies from existing CNV callers.

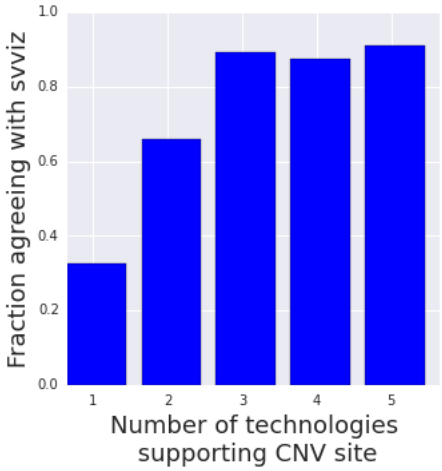


Fig. 5. Agreement (within each bin) with svviz classifications for sites with varying support from orthogonal technologies. We only compare sites with CN0, CN1 or CN2 classifications from both methods.

results in conjunction with all other images for the site.

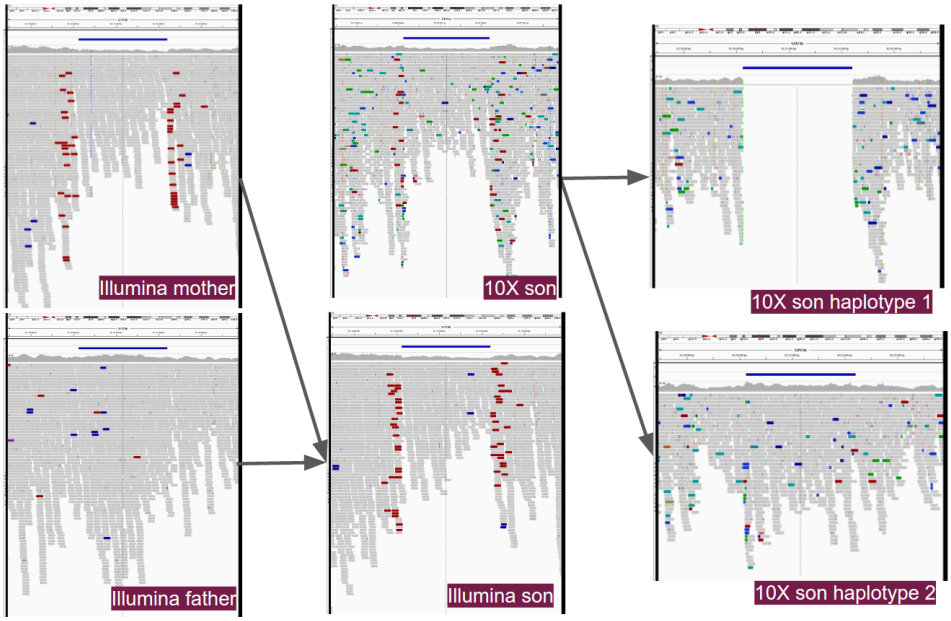


Fig. 6. Viewing all image types together shows the power of combining familial and phasing information in different sequencing platforms. This variant (chr15:36160125-36162210) was classified as copy number 1 in the son with CrowdVariant score 1.0 and is part of the high quality set. The variant is visible in the mother, both diploid son images and one of the haplotype images. Clockwise from top left: Illumina mother, 10X son, 10X son haplotype 1, 10X son haplotype 2, Illumina son, Illumina father.

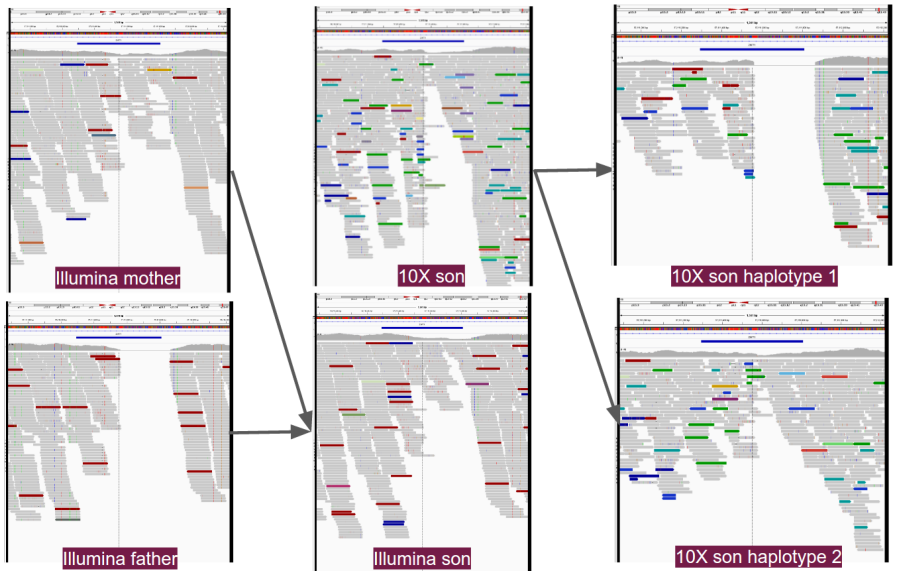


Fig. 7. Viewing all image types together shows the power of combining familial and phasing information in different sequencing platforms. This variant (chr19:57111292-57111809) was classified as CN1 in the son with score 0.89 and is part of the high quality set. Svviz classified this example as CN2 due to the imprecise breakpoints. Clockwise from top left: Illumina mother, 10X son, 10X son haplotype 1, 10X son haplotype 2, Illumina son, Illumina father. Mother appears to share CNV with the son, while the father is wildtype. Visualizations produced by default IGV settings.

2.4. *CrowdVariant can be used to curate a genome-wide high quality set of copy number variants*

Having demonstrated that we can use non-expert workers to curate a high quality set of copy number variants, we expanded our classifications genome-wide. We took all putative CNV sites that were supported by GIAB callsets from at least 2 technologies and had not been classified in the pilot set (n=2271) and recruited 20 non-expert classifications for each site for all 6 image types. Due to the larger volume of images, not every worker classified all images in the genome-wide set. Consistent with the pilot study, we observed strong agreement among non-expert workers in the genome-wide set. Again, the primary inconsistencies were classifications for the haplotype images [Figure 8].

We scored each site by the proportion of workers voting for each classification and applied the threshold determined by the first 500 sites to curate high quality genome-wide classifications. This resulted in 1,515 new high confidence sites for the son (NA24385). The CrowdVariant scores for these sites correlate with the number of supporting technologies [Figure 9]. Likely due to requiring 2 supporting technologies, these sites were in even stronger agreement with svviz with 97.2% agreement among sites given CN0/CN1/CN2 classifications with both methods [Figure 10]. The high quality genome-wide set includes calls for 93 sites that svviz found uncertain. The additional genome-wide set includes 959 CN1, 552 CN0, 3 CN2 and 1 None of the Above. The CrowdVariant scores for the genome-wide set of CNVs also demonstrate similar concordance with orthogonal technologies [Figure 9] and classify Mendelian violations in

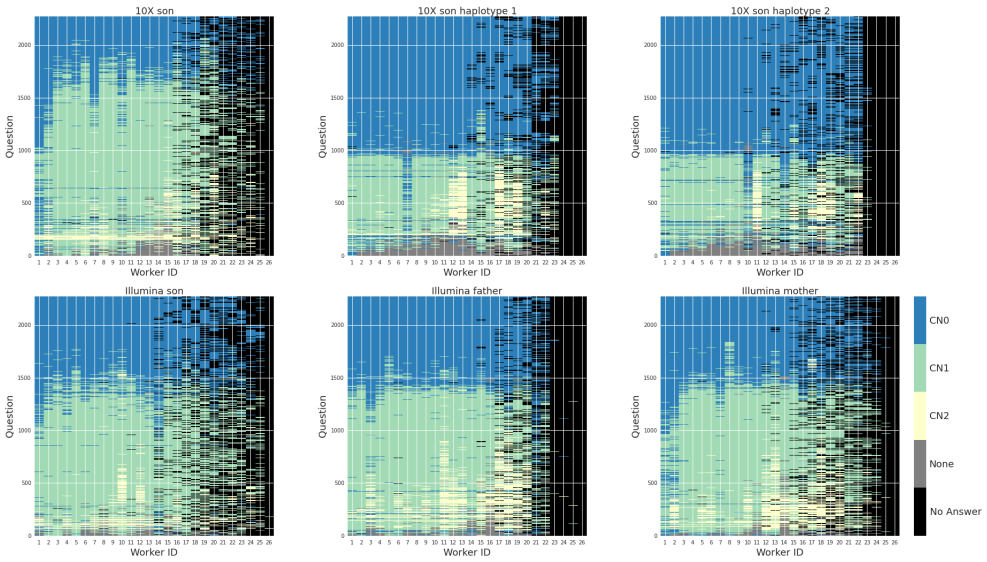


Fig. 8. Non-expert classifications for genome-wide sites in Phase 3 were color coded, weighted and clustered. Rows represent a question (i.e. an image of a particular site using a particular sequencing technology) and columns represent workers. Clockwise from top left: 10X son, 10X son haplotype 1 only, 10X son haplotype 2 only, Illumina mother, Illumina father, Illumina son.

the trio with auROC 0.94 [Supplementary Figure 8]. Above the threshold for high confidence determined from the pilot study, there was only one Mendelian violation in the genome-wide set occurring at a score of 0.94 [Supplementary Figure 9]. Combining with the 266 high quality sites from the pilot set, we finalized a set of 1,781 high confidence CNVs.

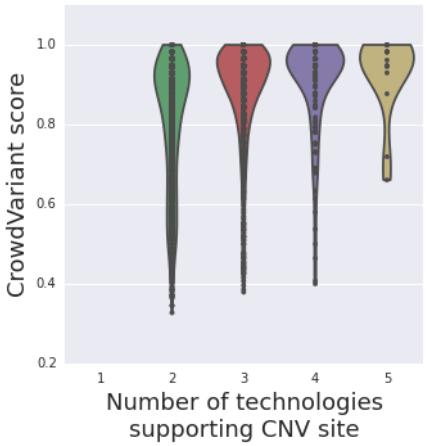


Fig. 9. CrowdVariant scores for all genome-wide sites determined by non-expert workers stratify by the number of supporting technologies from existing CNV callers.

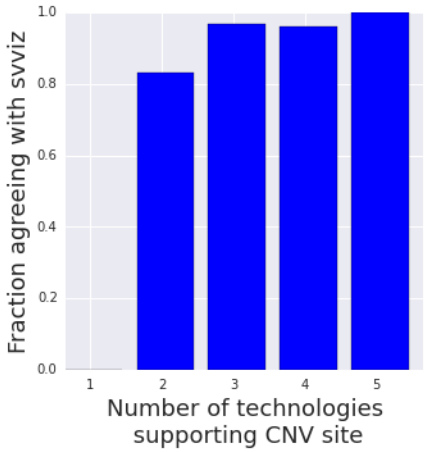


Fig. 10. Agreement with svviz classifications for genome-wide sites with varying support from orthogonal technologies. We only compare sites with CN0, CN1 or CN2 classifications from both methods.

3. Discussion

We show that individuals with no background in genomics can be trained to accurately classify and thereby curate copy number variants. This is possible because the classification of CNVs based on images of aligned NGS reads is ultimately a pattern recognition problem, and even non-experts with limited training can excel at recognizing these patterns. As soliciting expert participation is prohibitively more difficult than non-expert participation (evident in the small amount of expert data we were able to collect), the ability to use non-experts enables crowdsourcing on a substantially larger scale. Deployment of manual curation on the ever growing body of clinical samples would likely require this adaptation as the volume will quickly exceed the capacity of experts. In this study, the larger scale afforded by non-expert workers allowed us to curate thousands of putative CNVs across the entire genome of a single individual from the Genome In A Bottle reference collection.

We are able to use non-expert classifications by using confidence scores to recognize the limit of their abilities. For many applications, such as deriving gold standard labels to improve machine learning methods, it is more critical to determine which classifications are trusted than to classify everything correctly. As machine learning approaches are increasingly adopted to solve genomic problems, crowdsourcing can provide an avenue to derive trusted training sets at high throughput for low cost.

While we have shown that crowdsourcing can be used to generate high confidence labels for CNVs, there are several limitations to our study. First, the set of CNVs we present is not a complete set for the GIAB Ashkenazim son (NA24385), but instead a set of the highest confidence sites. Further, we only know that a CNV is segregating at the site, but we do not know its exact position or size. One broader limitation of crowdsourcing is that people can be consistent but wrong, however this limitation is shared by other approaches such as ensemble-based computational methods. In the current framework, our high confidence classifications are also enriched for sites that are overall easier to classify. However, there are many ways to increase confidence for more difficult questions by scaling the number of workers, augmenting training schemes, improving confidence metrics or considering alternative experimental designs such as those that incorporate both experts and non-experts depending on the particular question's difficulty. Nevertheless, we are confident that our crowdsourced, genome-wide set of curated CNVs will prove valuable to methods developers working to improve CNV calling algorithms.

Many possibilities exist for improving and expanding on this proof-of-concept study demonstrating the crowdsourcing curation of genomic variants. Incorporating images from additional technologies, such as long-read sequencing, could likely identify additional high confidence sites and remove some errors from using only short reads. Additional work might also use input from users about the precision of breakpoints. Other types of images could also be used, such as dot plots from assembly-assembly alignments and svviz images with reads mapped to reference and alternate alleles. These additional methods may help non-experts classify more difficult types of structural variants, like complex changes, insertions, inversions, and translo-

cations, as well as variants in difficult, repetitive regions of the genome.

We use Google's high throughput crowdsourcing platform, but as additional crowdsourcing platforms become available at low cost, soliciting participation from the crowd will become progressively easier. By using strategic experimental design, crowdsourcing can be a productive avenue to compete with and improve upon computational methods in difficult areas of genomics. Copy number variation, a domain where many experts still use manual inspection, is just one of these many areas. We provide a resource of high quality copy number variant classifications for a reference genome as a result of our study but ultimately see the potential expand far beyond these results.

Data Access

All Supplementary Methods, Figures and Data are available at `ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/technical/CrowdVariant_SupplementaryInfo/`. We provide the scores for each putative copy number variant site and label the high quality sites. All raw worker answers for both non-experts and experts are available as well.

References

Abyzov, A., Urban, A. E., Snyder, M., & Gerstein, M. (2011, jun). CNVnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. *Genome research*, *21*(6), 974–984.

Chaisson, M. J. P., Huddleston, J., Dennis, M. Y., Sudmant, P. H., Malig, M., Hormozdiari, F., ... Eichler, E. E. (2014, nov). Resolving the complexity of the human genome using single-molecule sequencing. *Nature*, *517*(7536), 608–611.

Cooper, S., Khatib, F., Treuille, A., Barbero, J., Lee, J., Beenen, M., ... Players, F. (2010). Predicting protein structures with a multiplayer online game. *Nature*, *466*(7307), 756–760.

DeLong, E. R., DeLong, D. M., & Clarke-Pearson, D. L. (1988, sep). Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, *44*(3), 837–45.

Drmanac, R., Sparks, A. B., Callow, M. J., Halpern, A. L., Burns, N. L., Kermani, B. G., ... Reid, C. A. (2010, jan). Human genome sequencing using unchained base reads on self-assembling DNA nanoarrays. *Science (New York, N.Y.)*, *327*(5961), 78–81.

English, A. C., Salerno, W. J., Hampton, O. A., Gonzaga-Jauregui, C., Ambreth, S., Ritter, D. I., ... Gibbs, R. A. (2015). Assessing structural variation in a personal genome-towards a human reference diploid genome. *BMC genomics*, *16*(1), 286.

Feuk, L., Carson, A. R., & Scherer, S. W. (2006). Structural variation in the human genome. *Nature Reviews Genetics*, *7*(2), 85–97.

Garrison, E., & Marth, G. (2012, jul). Haplotype-based variant detection from short-read sequencing.

Haghighi, A., Krier, J. B., Toth-Petroczy, A., Cassa, C. A., Frank, N. Y., Carmichael, N., ... Womens Hospital Project, B. G. M. (2018, aug 13). An integrated clinical program and crowdsourcing strategy for genomic sequencing and mendelian disease gene discovery. *NPJ Genom Med*, *3*, 21. Retrieved 2018-09-30, from `http://www.nature.com/articles/s41525-018-0060-9` doi: 10.1038/s41525-018-0060-9

Hormozdiari, F., Hajirasouliha, I., McPherson, A., Eichler, E. E., & Sahinalp, S. C. (2011, dec). Simultaneous structural variation discovery among multiple paired-end sequenced genomes. *Genome research*, *21*(12), 2203–2212.

Iqbal, Z., Caccamo, M., Turner, I., Flicek, P., & McVean, G. (2012, jan). De novo assembly and genotyping of variants using colored de Bruijn graphs. *Nature Genetics*, *44*(2), 226–232.

Ishikawa, ST and Gulick, V. (2012). Clickworkers interactive: towards a robust crowdsourcing tool for collecting scientific data. In *Lunar and planetary science conference* (pp. 2–3).

Kidd, J. M., Cooper, G. M., Donahue, W. F., Hayden, H. S., Sampas, N., Graves, T., ... Eichler, E. E. (2008, may). Mapping and sequencing of structural variation from eight human genomes. *Nature*, *453*(7191), 56–64.

Mak, A. C. Y., Lai, Y. Y. Y., Lam, E. T., Kwok, T.-P., Leung, A. K. Y., Poon, A., ... Kwok, P.-Y. (2016). Genome-Wide Structural Variation Detection by Genome Mapping on Nanochannel Arrays. *Genetics*, *202*(1).

Mills, R. E., Walter, K., Stewart, C., Handsaker, R. E., Chen, K., Alkan, C., ... Korbel, J. O. (2011, feb). Mapping copy number variation by population-scale genome sequencing. *Nature*, *470*(7332), 59–65.

Mohiyuddin, M., Mu, J. C., Li, J., Bani Asadi, N., Gerstein, M. B., Abyzov, A., ... Lam, H. Y. K. (2015, aug). MetaSV: an accurate and integrative structural-variant caller for next generation sequencing. *Bioinformatics (Oxford, England)*, *31*(16), 2741–2744.

Nattestad, M., & Schatz, M. C. (2016, oct). Assemblytics: a web analytics tool for the detection of variants from an assembly. *Bioinformatics (Oxford, England)*, *32*(19), 3021–3023.

Pinto, D., Darvishi, K., Shi, X., Rajan, D., Rigler, D., Fitzgerald, T., ... Feuk, L. (2011). Comprehensive assessment of array-based platforms and calling algorithms for detection of copy number variants. *Nature biotechnology*, *29*(6), 512–20.

Prather, E. E., Cormier, S., Wallace, C. S., Lintott, C., Jordan Raddick, M., & Smith, A. (2013). Measuring the conceptual understandings of citizen scientists participating in zooniverse projects: A first approach. *Astronomy Education Review*, *12*(1).

Raddick, M. J., Bracey, G., Gay, P. L., Lintott, C. J., Murray, P., Schawinski, K., ... Vandenberg, J. (2010). Galaxy Zoo: Exploring the Motivations of Citizen Science Volunteers. *Astronomy Education Review*, *9*(1), 010103.

Redon, R., Ishikawa, S., Fitch, K. R., Feuk, L., Perry, G. H., Andrews, T. D., ... Hurles, M. E. (2006). Global variation in copy number in the human genome. *Nature*, *444*(7118), 444–54.

Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.-C., & Müller, M. (2011, jan). pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC bioinformatics*, *12*(1), 77.

Scherer, S. W., Lee, C., Birney, E., Altshuler, D. M., Eichler, E. E., Carter, N. P., ... Feuk, L. (2007). Challenges and standards in integrating surveys of structural variation. *Nature genetics*, *39*(7 Suppl), S7–S15.

Spies, N., Zook, J. M., Salit, M., & Sidow, A. (2015, jul). Svviz: A read viewer for validating structural variants. *Bioinformatics*, *31*(24), 3994–3996.

Sudmant, P. H., Rausch, T., Gardner, E. J., Handsaker, R. E., Abyzov, A., Huddleston, J., ... Korbel, J. O. (2015). An integrated map of structural variation in 2,504 human genomes. *Nature*, *526*(7571), 75–81.

Tattini, L., D'Aurizio, R., & Magi, A. (2015). Detection of Genomic Structural Variants from Next-Generation Sequencing Data. *Frontiers in bioengineering and biotechnology*, *3*(June), 92.

Thorvaldsdóttir, H., Robinson, J. T., & Mesirov, J. P. (2013, mar). Integrative Genomics Viewer (IGV): High-performance genomics data visualization and exploration. *Briefings in Bioinformatics*, *14*(2), 178–192.

Zook, J. M., Catoe, D., McDaniel, J., Vang, L., Spies, N., Sidow, A., ... Salit, M. (2016, jun). Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Scientific data*, *3*, 160025.

A repository of microbial marker genes related to human health and diseases for host phenotype prediction using microbiome data

Wontack Han and Yuzhen Ye †

Computer Science Department, Indiana University, Bloomington, IN 47408, USA
†*E-mail: yye@indiana.edu*

The microbiome research is going through an evolutionary transition from focusing on the characterization of reference microbiomes associated with different environments/hosts to the translational applications, including using microbiome for disease diagnosis, improving the efficacy of cancer treatments, and prevention of diseases (e.g., using probiotics). Microbial markers have been identified from microbiome data derived from cohorts of patients with different diseases, treatment responsiveness, etc, and often predictors based on these markers were built for predicting host phenotype given a microbiome dataset (e.g., to predict if a person has type 2 diabetes given his or her microbiome data). Unfortunately, these microbial markers and predictors are often not published so are not reusable by others. In this paper, we report the curation of a repository of microbial marker genes and predictors built from these markers for microbiome-based prediction of host phenotype, and a computational pipeline called Mi2P (from Microbiome to Phenotype) for using the repository. As an initial effort, we focus on microbial marker genes related to two diseases, type 2 diabetes and liver cirrhosis, and immunotherapy efficacy for two types of cancer, non-small-cell lung cancer (NSCLC) and renal cell carcinoma (RCC). We characterized the marker genes from metagenomic data using our recently developed subtractive assembly approach. We showed that predictors built from these microbial marker genes can provide fast and reasonably accurate prediction of host phenotype given microbiome data. As understanding and making use of microbiome data (our second genome) is becoming vital as we move forward in this age of precision health and precision medicine, we believe that such a repository will be useful for enabling translational applications of microbiome data.

Keywords: microbiome; microbial marker gene; type 2 diabetes; liver cirrhosis; immunotherapy efficacy.

1. Introduction

Recent studies of microbiomes (i.e., communities of microorganisms) have shaped a new view of the biological world in which various microbial organisms play important roles in the health of humans, animals, plants, and the environment.[1–4] Metagenome-wide association studies[5] have enabled the high-resolution discovery of associations between the microbiome and human diseases, including type 2 diabetes,[6] liver cirrhosis,[7] atherosclerotic cardiovascular disease,[8] colorectal cancer[9] and rheumatoid arthritis.[10] The announcement of the National Microbiome Initiative (NMI) on May 13, 2016, marks a milestone in microbiome research. The NMI aims

to advance the understanding of microbiome behavior and enable protection and restoration of healthy microbiome function. Development of computational tools for interpretation and integration of meta-omics data will be key to advancing the field and ultimately achieving the goal of the NMI.

Unlike traditional microbial genomic sequencing projects, metagenomics attempts to directly characterize the entire collection of genes within an environmental sample (i.e., the metagenome) and analyze their biochemical activities and complex interactions.[11,12] Landmark progress in metagenomics occurred in 2004[13,14] when two research groups published results from large-scale environmental sequencing projects. Many more metagenomic projects have been conducted or are ongoing, representing broadened applications from ecology and environmental sciences[15] to the chemical industry[16] and human health.[17] Metagenomics, in principle, enables the study of any microbial organism, including the large number of microorganisms that cannot be isolated or are difficult to grow in a lab. More importantly, microbes, by nature, live in communities where they interact with each other by exchanging nutrients, metabolites, and signaling molecules. Metagenomics enables the characterization of microbes in natural environments, addressing important biological questions related to microbial environments such as the diversity of microbes in different environments,[18] microbial (and microbe-host) interactions,[19] and the environmental and evolutionary processes.[20]

Earlier metagenomics studies focused on the characterization of reference microbiomes associated with different environments/hosts. Recent studies shift the emphasis to the translational applications, including using microbiome for disease diagnosis, improving the efficacy of cancer treatments (including cancer chemotherapy and immunotherapy), and prevention of diseases (e.g., using probiotics).[21] Gut bacterium *Eggerthella lenta* was found to be able to manipulate cardiac drug inactivation.[22] Harnessing the host immune system constitutes a promising cancer therapeutic because of its potential to specifically target tumor cells while limiting harm to normal tissues. Recent clinical success has fueled the enthusiasm about immunotherapy using antibodies that block immune inhibitory pathways, specifically, the CTLA-4 and the PD-1/PD-L1 axis.[22,23] The gut microbiota plays an important role in shaping hosts immune responses,[24] so there is no surprise that a few recent studies have shown that intestinal microbiota (and some particular microbial species/strains) can mediate immune activation in response to chemotherapeutic agents and immunotherapy. Sivan and colleagues[25] found that commensal Bifidobacterium promotes antitumor immunity and facilitates anti PD-L1 efficacy. They also found that oral administration of Bifidobacterium alone improved tumor control to the same degree as anti PD-L1 therapy (checkpoint blockade), and combination treatment nearly abolished tumor outgrowth. Gut microbiota can also modulate the actions of chemotherapeutic drugs used in cancer and other disease, reducing the toxicity of chemotherapeutic compounds and improve their efficacy.[26] A working knowledge of the microbiome (our second genome[27]) is vital as we move forward in this age of precision health and precision medicine,[28] especially in the area of cancer research, which aims at effective treatments for various kinds of cancer based on the knowledge of genetics, biology of the disease and host-microbiome interactions.[29]

The success of the translational applications of microbiome data relies on the character-

ization of differential markers (species, genes, biological pathways, among others) that can differentiate different groups of microbiome data (e.g., healthy individuals versus patients, treatment responders versus non-responders). It is also important to understand factors influencing the gut microbiome and strategies to manipulate the microbiome to augment therapeutic responses and disease prevention.[30]

To derive microbial markers that are associated with a specific host phenotype (e.g., healthy versus diseased), a key task is to compare two groups of microbiome (e.g., one group of microbiome data derived from healthy individuals versus a group derived from patients) to detect *consistent* differences (e.g., species or genes) between the groups, considering the large inter- and intra-individual variations of the microbiome.[31] The typical analysis workflow is to assemble and annotate metagenomic datasets individually or as a whole, followed by statistical tests to identify differentially abundant species/genes. The subtractive assembly approaches we previously developed, subtractive assembly (SA)[32] and concurrent subtractive assembly (CoSA) approach,[33] are *de novo* assembly approaches for comparative metagenomics that first detect differential reads between two groups of metagenomes and then only assemble these reads. When evaluated using simulated and real type 2 diabetes microbiome datasets,[33] our subtractive assembly approaches reduce the datasets up front, which also result in better characterization of the differential genes.

Recent studies have revealed microbial markers for disease diagnosis and other purposes, and predictors built based on these markers have achieved promising accuracy for predictions. The pitfall of most of these studies is that the microbial markers and predictors built from these markers are not made available for others to use. For example, Qin et al.[7] constructed a support vector machine discriminator based on microbiome data for liver cirrhosis prediction using 15 gene markers, achieving impressive accuracy, with AUC (area under the receiver operating characteristic curve) of 0.918 and 0.838, respectively, for training and leave-one-out cross-validation. Although the authors listed the identities of these 15 genes in a supplementary table (Supp Table 12 in[7]), they did not release the gene sequences, nor the discriminator they built. It makes it impossible for others to use their marker genes and predictors. Using our recently developed computational approach CoSA,[33] we re-analyzed several large collections of publicly available microbiome datasets, in an attempt to create a repository of microbial marker genes and the predictors built from these marker genes for translational applications of microbiome data (e.g., to predict if a cancer patient is likely to be responsive to PD-1 blockage treatment given his/her microbiome data). We note there is no shortage of microbiome repositories; instances include the Human Microbiome Project repository (`http://hmpdacc.org`) and the MG-RAST server (`https://www.mg-rast.org`). However, there is no repository of bacterial marker genes and predictors for microbiome-based predictions to the best of our knowledge. As a proof of concept, we focused on two diseases, type 2 diabetes and liver cirrhosis, and two types of cancers. We first extracted microbial marker genes from these microbiome datasets, then built predictors using these genes, and finally created a repository of the marker genes and predictors, as well as a companion computational pipeline for using this repository.

2. Methods

2.1. *Microbiome datasets*

We focus on microbial genes related with two diseases and the treatment efficacy of two types of cancer:

(a) T2D (type 2 diabetes). We used the T2D cohort from a study,[6] which contains microbiome data from two groups of 70-year-old European women, one group of 50 with T2D and the other a matched group of healthy controls (NGT group; 43 participants). We previously used this cohort for testing our subtractive assembly approaches.[32,33]
(b) Cirrhosis (liver cirrhosis). Qin et al.[7] derived metagenomic datasets from 98 Chinese patients with liver cirrhosis and 83 healthy individuals as training datasets to infer marker genes and build a predictor, and microbiome data from additional 25 patients and 31 healthy controls as validation datasets. Similarly, we used their training datasets for characterization of marker genes and training of predictors, and their validation datasets for independent tests of the predictors for liver cirrhosis.
(c) NSCLC (non-small-cell lung cancer). It has been shown that gut bacteria can affect patient responses to cancer immunotherapy (e.g., immune checkpoint inhibitors ICIs that target the PD-1/PD-L1 axis). Routy et al.[34] found that primary resistance to ICIs can be attributed to abnormal gut microbiome composition, and fecal microbiota transplantation (FMT) from cancer patients who responded to ICIs into germ-free or antibiotic-treated mice ameliorated the antitumor effects of PD-1 blockade, whereas FMT from non-responding patients failed to do so. They sequenced the microbiome of the stool samples at diagnosis, and showed correlations between clinical responses to ICIs and relative abundance of *Akkermansia muciniphila*. We used microbiome datasets from this study, which includes 32 non-responders and 33 responders, aiming to infer marker genes that can be used to distinguish responders from non-responders.
(d) RCC (renal cell carcinoma). We used datasets from the same study[34] that involve 20 non-responders versus 42 responders to a different cancer type, renal cell carcinoma.

Table 1 summarizes the microbiome datasets that were re-analyzed in this paper.

Table 1: Summary of the microbiome datasets for training the predictors.

Abr.	Disease	Reference	# of samples	Total base pairs (bps)
T2D	Type 2 diabetes	[6]	93	225 GB
Cirrhosis	Liver cirrhosis	[7]	181	817 GB
NSCLC	Non-small-cell lung cancer	[34]	65	153 GB
RCC	Renal cell carcinoma	[34]	62	147 GB

2.2. *Microbial gene characterization and quantification*

For each collection of above mentioned microbiome datasets, we first applied CoSA to assemble genes that are potentially differential between the groups (i.e., for the T2D collection and the liver collection, the patient group versus group of healthy individuals, and for the NSCLC and RCC collections, responders versus non-responders). These genes were then subject to feature selection. Using selected marker genes, different machine learning (ML) approaches were employed to build predictors for microbiome-based host phenotype prediction. We refer the readers to our previous publications[32,33] for details about our subtractive assembly approach CoSA. Briefly, the CoSA approach uses a Wilcoxon rank-sum (WRS) test to detect k-mers that are differentially abundant between two groups of microbiomes (CoSA uses KMC2[35] for k-mer counting, and employs the "mannwhitneyutest" function from ALGLIB (`http://www.alglib.net`) for the test). It then uses identified differential k-mers to extract reads (by a voting strategy) that are likely from the sub-metagenome with consistent abundance differences between the groups of microbiomes. Further, CoSA attempts to reduce the redundancy of reads (from abundant common species) by excluding reads containing abundant k-mers. Extracted reads are then assembled using MegaHit,[36] and genes are predicted from the assembled contigs using FragGeneScan.[37] The quantification of the genes in each microbiome is done by reads mapping of shotgun reads onto the genes using Bowtie 2.[38] We counted a gene's abundance based on the counts of both uniquely and multiplely mapped reads (the contribution of multiplely mapped reads to a gene was computed according to the proportion of the read counts divided by the gene's unique abundance[7]). The read counts were then normalized per kilobase of gene per million of reads in each sample.

2.3. *Inference of microbial marker genes using machine learning approaches*

Microbial genes assembled and quantified mentioned above for the different microbiome datasets were used as candidate features for selecting microbial marker genes and for training predictors for microbiome-based host phenotype prediction (see Figure 1(a)). For feature selection, we first applied a q-value cutoff and then used two different feature selection methods (tree-based feature selection and L1-based feature selection) to select a smaller number of microbial genes, and used them as microbial marker genes. We tried different ML algorithms for phenotype prediction, including Support Vector Machines (SVM), Random Forests (RF), Decision Trees (DT), Neural Networks (NN), and K-nearest Neighbor (KN) approach, along with different cross-validation strategies. We used the scikit-learn (`http://scikit-learn.org`) implementation of these ML approaches in this study. We tested RF with 10, 100 and 1000 trees and KN with 20 neighbors. For NN, we used Bernoulli Restricted Boltzmann Machine (RBM) with 3200 binary hidden units. We used the default settings for SVM and DT.

2.4. *Mi2P: from microbiome to phenotype*

We created a repository of above mentioned microbial marker genes and predictors built from the marker genes. We also developed a computational pipeline called Mi2P (which stands for "from Microbiome to Phenotype") for users to use this repository. As shown in Figure

1(b), Mi2P is composed of three main steps: 1) mapping of metagenomic sequencing reads onto the marker genes using Bowtie 2;[38] 2) quantification of the marker genes based on read counts, using both uniquely and multiplely mapped reads (see 2.2); and 3) the estimated gene abundances are used as input features to the microbiome-based phenotype predictors. A wrapper script is included in the pipeline for the one-stop use of our pipeline, which takes a metagenomic dataset as the input, and reports prediction as the main output. It also outputs some intermediate results including the estimated gene abundances. Mi2P is available as open source software for download at sourceforge (`https://sourceforge.net/projects/mi2p/`).

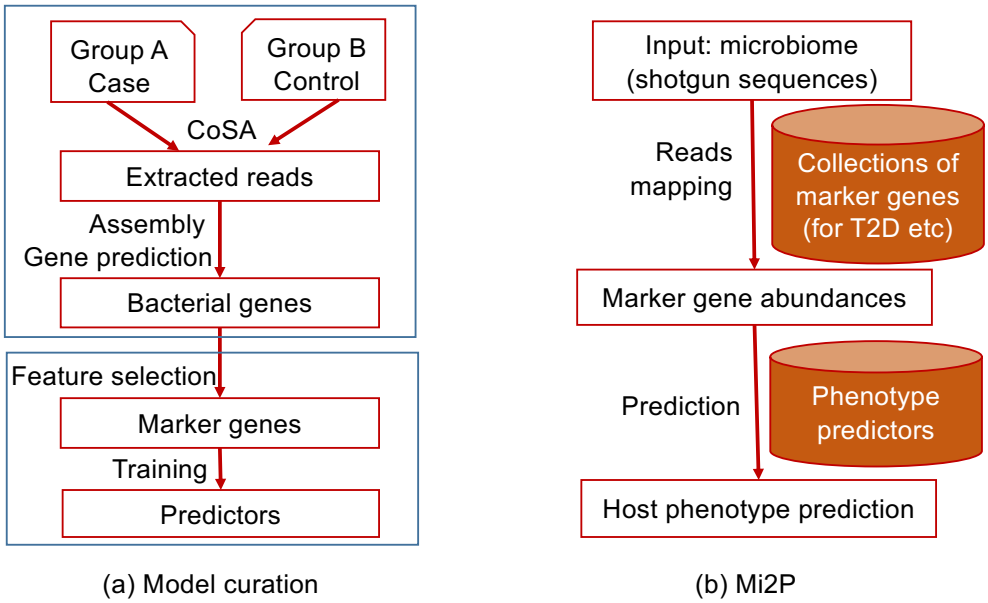


Fig. 1: Schematic representations of the model curation based on CoSA (a) and Mi2P (Microbiome to Phenotype) pipeline (b).

3. Results

3.1. *Accuracy of microbiome-based predictors*

We built predictors for predicting host phenotype based on the microbiome data. We evaluated the accuracy of the predictors using different cross-validation strategies and ML approaches. Furthermore, we tested two different feature selection approaches (tree-based and L1-based) with liver cirrhosis data sets. Since we have already reported the performance of T2D prediction in our previous publications,[32,33] we focused on reporting the results for liver cirrhosis and cancer treatment responsiveness prediction based on microbiome data in this paper.

Figure 2 shows the ROC plots for liver cirrhosis prediction using different ML approaches and feature selection methods. The figure shows that RF achieved better predictions than

SVM approach. It also shows that predictors built from genes selected using the tree-based feature selection method performed better as compared to L1-based feature selection method. We therefore chose the tree-based feature selection as the default approach in our pipeline.

Table 2 summarizes the accuracy of the predictors we built for liver cirrhosis. Our SVM based predictor achieved comparable performance as the predictor reported in Qin et al..[7] However, our RF based predictor achieved significantly better predictions with higher AUCs. We speculate that the accuracy improvement is a result of the combination of more marker genes and a different machine learning approach (RF). We note that we tested RF using different numbers of trees, including 10, 100 and 1000. We found that RF with 100 trees and 1000 trees achieved slightly better predictions than RF with 10 trees. Balancing running time and accuracy, we chose RF with 100 trees.

Table 2: Accuracy of microbiome-based predictors for liver cirrhosis.

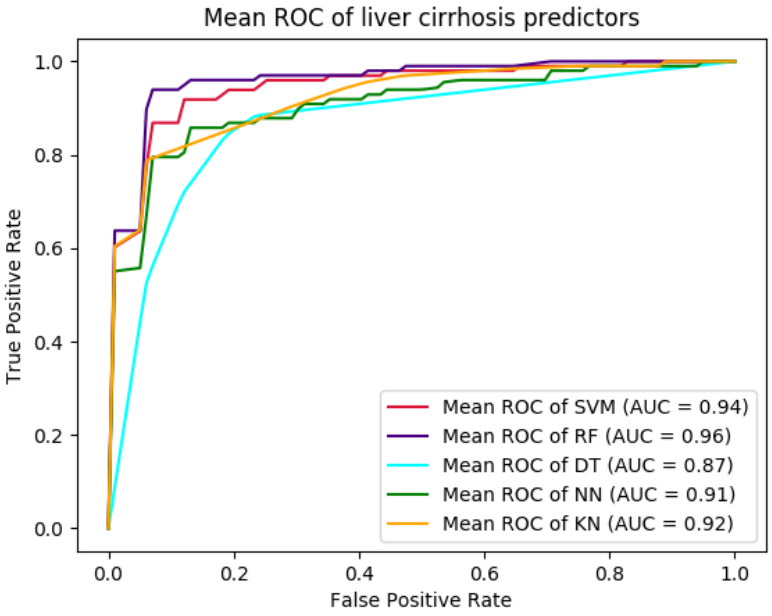
	methods	# of marker genes	SVM	RF (100 trees)	NN	KN
cross[a]	Qin et al.	15[c]	0.84 [c]	N/A	N/A	N/A
	Our approach	46	0.92	0.92	0.88	0.71
validation[b]	Qin et al.	15[c]	0.84 [c]	N/A	N/A	N/A
	Our approach	46	0.83	0.93	0.81	0.72

[a]: the "cross" columns show the leave-one-out validation result (see Figure 2 (a) for 5 fold cross-validation results). [b]: validation using microbiome data unseen in the training of the predictor. [c]: numbers taken from the paper.[7]

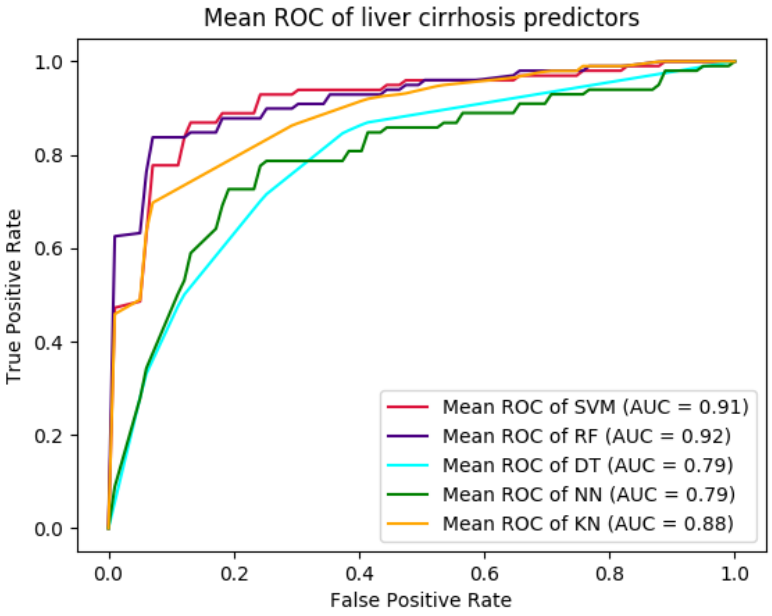
Table 3 summarizes the accuracy for predicting immunotherapy responders versus non-responders based on microbiome data. Correlations between clinical responses to immunotherapy (ICI) and the relative abundance of *Akkermansia muciniphila* were reported in,[34] however, no predictors were built by the authors. Here, we built predictors for immunotherapy responsiveness using the RF approach with a small collection of marker genes, which achieved reasonably accurate predictions for NSCLC. Predictions of RCC based on microbiome data were less accurate. We tested RF predictors with different trees, and results show that RF with 100 trees performed relatively well for both cancers, similar to prediction of liver cirrhosis. Therefore, we chose RF predictors with 100 trees for immunotherapy resposiveness prediction to include in our Mi2P package. We note that we also applied SVM approach to this dataset, which however achieved much worse predictions (AUC = 0.61) than the RF predictors.

3.2. *Microbial marker genes*

We include the sequences of microbial marker genes (both proteins and gene sequences), along with their annotations (by hmmscan[39]) in the Mi2P package. Table 4 shows a few examples



(a) Tree-based feature selection



(b) L1-based feature selection

Fig. 2: Receiver operating characteristic (ROC) plots of the liver cirrhosis predictors using different ML approaches. We also tested two feature selection methods: tree-based feature selection and L1-based feature selection, and the results are shown in (a) and (b), respectively. The ROC curves were averaged over five cross validation results.

Table 3: Accuracy of microbiome-based prediction of responders versus non-responders to cancer treatment using RF (with 10, 100, and 1000 trees), DT and NN approaches.

Cancer type	# of marker genes	RF 10	RF 100	RF 1000	DT mean AUC	NN mean AUC
NSCLC	116	0.86	0.91	0.89	0.72	0.81
RCC	85	0.84	0.83	0.81	0.79	0.78

identified from the liver cirrhosis cohort. These marker genes can be either more abundant in healthy individuals (i.e., depleted in liver cirrhosis microbiomes), or more abundant in liver cirrhosis microbiomes. We also note that a significant fraction of genes have no functional annotations according to hmmscan search (or annotated to a domain without functional annotations, such as DUF3829): 0 out of 5 (0%) depleted genes, and 4 out of 41 (10%) enriched genes in liver cirrohosis microbiomes have no functional annotations.

Table 4: Examples of microbial marker genes for liver cirrhosis prediction.

Gene id	Putative function	Pfam domain
	Depleted in liver cirrhosis microbiome	
H_k99_23554_31_534_	Tripartite ATP-independent periplasmic transporters	DctQ
H_k99_23763_1365_1613_	Helix-turn-helix domain	HTH_31
H_k99_38620_1_453_+	Acyltransferase family	Acyl_transf_3
H_k99_59586_373_654_-	Amidohydrolase	Amidohydro_2
H_k99_64410_1_617_-	REC lobe of CRISPR-associated endonuclease Cas9	Cas9_REC
	Enriched in liver cirrhosis microbiome	
L_k99_1592_1_390_-	Polysaccharide biosynthesis C-terminal domain	Polysacc_synt_C
L_k99_7366_1_565_-	Carbon starvation protein CstA	CstA
L_k99_13622_1_326_+	Septation ring formation regulator, EzrA	EzrA
L_k99_52773_82_623_+	Sodium:sulfate symporter transmembrane region	Na_sulph_symp
L_k99_52825_1_408_+	D-isomer specific 2-hydroxyacid dehydrogenase	2-Hacid_dh_C

3.3. *Running time of Mi2P pipeline*

We provide a wrapper script in Mi2P pipeline for users to employ our repository of microbial marker genes and predictors. We show that this pipeline gives fast prediction of host phenotype from a query microbiome dataset (of shotgun sequences), thanks to the relatively small number

of microbial marker genes that need to be considered. For example, on a linux computer (with Intel(R) Xeon(R) CPU E5-2623 v3 @ 3.00GHz), running the pipeline for two test datasets, one from the liver cirrhosis collection (ERR528314 with 3 Gbps), and the other one from the NSCLC collection (ERR2213736 with 2 Gbps) each took less than 6 min to complete.

4. Discussion

Our current repository of microbial marker genes and predictors is rather limited, covering only four host phenotypes. We plan to apply the same analysis to more collections of microbiome datasets associated with human diseases and treatment efficacy. We believe there will be no shortage of such datasets due to the soaring interests in microbiome research associated with human health and diseases. In addition, we will seek to collect microbial marker genes using other approaches (e.g., based on the literature search) to enrich our repository.

A challenging problem in making our repository of microbial maker genes and predictors useful will be the generalization issue, due to both the biological complexity (e.g., stratification of the samples that were used to build the classifiers) and technical complexity (e.g., overfitting of the predictors). The generalization issue is a general problem in machine learning, and methods have been proposed to alleviate the problem. We will explore some of the existing approaches to address this challenge. In addition, we will explore approaches to provide confidence of predictions, rather than to simply provide yes or no prediction.

Further studies of the microbial marker genes will be needed to understand why they are important for microbiome-host interaction, contributing to the host phenotype. We also note that a significant fraction of the identified marker genes are of unknown functions. We will exploit different homology- and context-based approaches to predict the functions of these genes. Boosted by the accumulation of microbial genomes and metagenomes, a few new methods, including our own guilt-by-association approach (the community profiling approach), have been developed for functional annotation of microbial genes.[40,41] We plan to utilize these approaches in our future research.

Acknowledgments

This work was supported by the NIH grant 1R01AI108888 to Ye, and partially supported by the Indiana University Precision Health Initiative.

References

1. S. Haase, A. Haghikia, N. Wilck, D. N. Muller and R. A. Linker, *Immunology* **154**, 230 (Jun 2018).
2. L. Zhao, F. Zhang, X. Ding, G. Wu, Y. Y. Lam, X. Wang, H. Fu, X. Xue, C. Lu, J. Ma, L. Yu, C. Xu, Z. Ren, Y. Xu, S. Xu, H. Shen, X. Zhu, Y. Shi, Q. Shen, W. Dong, R. Liu, Y. Ling, Y. Zeng, X. Wang, Q. Zhang, J. Wang, L. Wang, Y. Wu, B. Zeng, H. Wei, M. Zhang, Y. Peng and C. Zhang, *Science* **359**, 1151 (03 2018).
3. Z. Dai, O. O. Coker, G. Nakatsu, W. K. K. Wu, L. Zhao, Z. Chen, F. K. L. Chan, K. Kristiansen, J. J. Y. Sung, S. H. Wong and J. Yu, *Microbiome* **6**, p. 70 (Apr 2018).
4. A. Altamirano-Barrera, M. Uribe, N. C. Chavez-Tapia and N. Nuno-Lambarri, *J. Nutr. Biochem.* **60**, 1 (Mar 2018).

5. J. Wang and H. Jia, *Nat. Rev. Microbiol.* **14**, 508 (08 2016).
6. J. Qin, Y. Li, Z. Cai, S. Li, J. Zhu, F. Zhang, S. Liang, W. Zhang, Y. Guan, D. Shen, Y. Peng, D. Zhang, Z. Jie, W. Wu, Y. Qin, W. Xue, J. Li, L. Han, D. Lu, P. Wu, Y. Dai, X. Sun, Z. Li, A. Tang, S. Zhong, X. Li, W. Chen, R. Xu, M. Wang, Q. Feng, M. Gong, J. Yu, Y. Zhang, M. Zhang, T. Hansen, G. Sanchez, J. Raes, G. Falony, S. Okuda, M. Almeida, E. LeChatelier, P. Renault, N. Pons, J. M. Batto, Z. Zhang, H. Chen, R. Yang, W. Zheng, S. Li, H. Yang, J. Wang, S. D. Ehrlich, R. Nielsen, O. Pedersen, K. Kristiansen and J. Wang, *Nature* **490**, 55 (Oct 2012).
7. N. Qin, F. Yang, A. Li, E. Prifti, Y. Chen, L. Shao, J. Guo, E. Le Chatelier, J. Yao, L. Wu, J. Zhou, S. Ni, L. Liu, N. Pons, J. M. Batto, S. P. Kennedy, P. Leonard, C. Yuan, W. Ding, Y. Chen, X. Hu, B. Zheng, G. Qian, W. Xu, S. D. Ehrlich, S. Zheng and L. Li, *Nature* **513**, 59 (Sep 2014).
8. Z. Jie, H. Xia, S. L. Zhong, Q. Feng, S. Li, S. Liang, H. Zhong, Z. Liu, Y. Gao, H. Zhao, D. Zhang, Z. Su, Z. Fang, Z. Lan, J. Li, L. Xiao, J. Li, R. Li, X. Li, F. Li, H. Ren, Y. Huang, Y. Peng, G. Li, B. Wen, B. Dong, J. Y. Chen, Q. S. Geng, Z. W. Zhang, H. Yang, J. Wang, J. Wang, X. Zhang, L. Madsen, S. Brix, G. Ning, X. Xu, X. Liu, Y. Hou, H. Jia, K. He and K. Kristiansen, *Nat Commun* **8**, p. 845 (10 2017).
9. G. Zeller, J. Tap, A. Y. Voigt, S. Sunagawa, J. R. Kultima, P. I. Costea, A. Amiot, J. Bohm, F. Brunetti, N. Habermann, R. Hercog, M. Koch, A. Luciani, D. R. Mende, M. A. Schneider, P. Schrotz-King, C. Tournigand, J. Tran Van Nhieu, T. Yamada, J. Zimmermann, V. Benes, M. Kloor, C. M. Ulrich, M. von Knebel Doeberitz, I. Sobhani and P. Bork, *Mol. Syst. Biol.* **10**, p. 766 (Nov 2014).
10. X. Zhang, D. Zhang, H. Jia, Q. Feng, D. Wang, D. Liang, X. Wu, J. Li, L. Tang, Y. Li, Z. Lan, B. Chen, Y. Li, H. Zhong, H. Xie, Z. Jie, W. Chen, S. Tang, X. Xu, X. Wang, X. Cai, S. Liu, Y. Xia, J. Li, X. Qiao, J. Y. Al-Aama, H. Chen, L. Wang, Q. J. Wu, F. Zhang, W. Zheng, Y. Li, M. Zhang, G. Luo, W. Xue, L. Xiao, J. Li, W. Chen, X. Xu, Y. Yin, H. Yang, J. Wang, K. Kristiansen, L. Liu, T. Li, Q. Huang, Y. Li and J. Wang, *Nat. Med.* **21**, 895 (Aug 2015).
11. J. Handelsman, M. R. Rondon, S. F. Brady, J. Clardy and R. M. Goodman, *Chem. Biol.* **5**, R245 (Oct 1998).
12. T. Thomas, J. Gilbert and F. Meyer, *Microb Inform Exp* **2**, p. 3 (Feb 2012).
13. G. W. Tyson, J. Chapman, P. Hugenholtz, E. E. Allen, R. J. Ram, P. M. Richardson, V. V. Solovyev, E. M. Rubin, D. S. Rokhsar and J. F. Banfield, *Nature* **428**, 37 (Mar 2004).
14. J. Hu and J. L. Blanchard, *Mol. Biol. Evol.* **26**, 5 (Jan 2009).
15. E. A. Dinsdale, O. Pantos, S. Smriga, R. A. Edwards, F. Angly, L. Wegley, M. Hatay, D. Hall, E. Brown, M. Haynes, L. Krause, E. Sala, S. A. Sandin, R. V. Thurber, B. L. Willis, F. Azam, N. Knowlton and F. Rohwer, *PLoS ONE* **3**, p. e1584 (Feb 2008).
16. F. Branco dos Santos, W. M. de Vos and B. Teusink, *Curr. Opin. Biotechnol.* **24**, 200 (Apr 2013).
17. T. H. consortium, *Nature* **486**, 207 (Jun 2012).
18. E. Stulberg, D. Fravel, L. M. Proctor, D. M. Murray, J. LoTempio, L. Chrisey, J. Garland, K. Goodwin, J. Graber, M. C. Harris, S. Jackson, M. Mishkind, D. M. Porterfield and A. Records, *Nat Microbiol* **1**, p. 15015 (Jan 2016).
19. M. B. Burns, E. Montassier, J. Abrahante, S. Priya, D. E. Niccum, A. Khoruts, T. K. Starr, D. Knights and R. Blekhman, *PLoS Genet.* **14**, p. e1007376 (Jun 2018).
20. S. D. Hooper, J. Raes, K. U. Foerstner, E. D. Harrington, D. Dalevi and P. Bork, *PLoS ONE* **3**, p. e2607 (Jul 2008).
21. P. Joglekar and J. A. Segre, *Cell* **169**, 378 (04 2017).
22. H. J. Haiser, D. B. Gootenberg, K. Chatman, G. Sirasani, E. P. Balskus and P. J. Turnbaugh, *Science* **341**, 295 (Jul 2013).

23. O. Hamid, C. Robert, A. Daud, F. S. Hodi, W. J. Hwu, R. Kefford, J. D. Wolchok, P. Hersey, R. W. Joseph, J. S. Weber, R. Dronca, T. C. Gangadhar, A. Patnaik, H. Zarour, A. M. Joshua, K. Gergich, J. Elassaiss-Schaap, A. Algazi, C. Mateus, P. Boasberg, P. C. Tumeh, B. Chmielowski, S. W. Ebbinghaus, X. N. Li, S. P. Kang and A. Ribas, *N. Engl. J. Med.* **369**, 134 (Jul 2013).
24. I. I. Ivanov and K. Honda, *Cell Host Microbe* **12**, 496 (Oct 2012).
25. A. Sivan, L. Corrales, N. Hubert, J. B. Williams, K. Aquino-Michaels, Z. M. Earley, F. W. Benyamin, Y. M. Lei, B. Jabri, M. L. Alegre, E. B. Chang and T. F. Gajewski, *Science* **350**, 1084 (Nov 2015).
26. J. L. Alexander, I. D. Wilson, J. Teare, J. R. Marchesi, J. K. Nicholson and J. M. Kinross, *Nat Rev Gastroenterol Hepatol* **14**, 356 (Jun 2017).
27. B. Zhu, X. Wang and L. Li, *Protein Cell* **1**, 718 (Aug 2010).
28. V. Gopalakrishnan, B. A. Helmink, C. N. Spencer, A. Reuben and J. A. Wargo, *Cancer Cell* **33**, 570 (Apr 2018).
29. A. V. Contreras, B. Cocom-Chan, G. Hernandez-Montes, T. Portillo-Bobadilla and O. Resendis-Antonio, *Front Physiol* **7**, p. 606 (2016).
30. T. R. Simms-Waldrip and A. Y. Koh, *Cell Host Microbe* **23**, 423 (04 2018).
31. T. Hisada, K. Endoh and K. Kuriki, *Arch. Microbiol.* **197**, 919 (Sep 2015).
32. M. Wang, T. G. Doak and Y. Ye, *Genome Biol.* **16**, p. 243 (Nov 2015).
33. W. Han, M. Wang and Y. Ye, *Res Comput Mol Biol* **2017**, 18 (2017).
34. B. Routy, E. Le Chatelier, L. Derosa, C. P. M. Duong, M. T. Alou, R. Daillere, A. Fluckiger, M. Messaoudene, C. Rauber, M. P. Roberti, M. Fidelle, C. Flament, V. Poirier-Colame, P. Opolon, C. Klein, K. Iribarren, L. Mondragon, N. Jacquelot, B. Qu, G. Ferrere, C. Clemenson, L. Mezquita, J. R. Masip, C. Naltet, S. Brosseau, C. Kaderbhai, C. Richard, H. Rizvi, F. Levenez, N. Galleron, B. Quinquis, N. Pons, B. Ryffel, V. Minard-Colin, P. Gonin, J. C. Soria, E. Deutsch, Y. Loriot, F. Ghiringhelli, G. Zalcman, F. Goldwasser, B. Escudier, M. D. Hellmann, A. Eggermont, D. Raoult, L. Albiges, G. Kroemer and L. Zitvogel, *Science* **359**, 91 (Jan 2018).
35. S. Deorowicz, M. Kokot, S. Grabowski and A. Debudaj-Grabysz, *Bioinformatics* **31**, 1569 (May 2015).
36. D. Li, R. Luo, C.-M. Liu, C.-M. Leung, H.-F. Ting, K. Sadakane, H. Yamashita and T.-W. Lam, *Methods* **102**, 3 (2016).
37. M. Rho, H. Tang and Y. Ye, *Nucleic Acids Res.* **38**, p. e191 (Nov 2010).
38. B. Langmead and S. L. Salzberg, *Nat. Methods* **9**, 357 (Mar 2012).
39. R. D. Finn, J. Clements and S. R. Eddy, *Nucleic Acids Res.* **39**, 29 (Jul 2011).
40. D. Jiao, W. Han and Y. Ye, *Methods* **129**, 8 (10 2017).
41. C. Beck, H. Knoop and R. Steuer, *PLoS Genet.* **14**, p. e1007239 (03 2018).

AICM: A Genuine Framework for Correcting Inconsistency Between Large Pharmacogenomics Datasets

Zhiyue Tom Hu[†], Yuting Ye[2], Patrick A. Newbury[3], Haiyan Huang [4#], Bin Chen [5#]
[† 2 4]*Department of Biostatistics, University of California, Berkeley*
[4]*Department of Statistics, University of California, Berkeley*
E-mails: {zyhu95, yeyt, hyh0110}@berkeley.edu
[3 5]*Department of Pediatrics and Human Development, Michigan State University*
[3 5]*Department of Pharmacology and Toxicology, Michigan State University*
E-mails: newburyp@msu.edu, Bin.Chen@hc.msu.edu
[#]*: corresponding author*

The inconsistency of open pharmacogenomics datasets produced by different studies limits the usage of such datasets in many tasks, such as biomarker discovery. Investigation of multiple pharmacogenomics datasets confirmed that the pairwise sensitivity data correlation between drugs, or rows, across different studies (drug-wise) is relatively low, while the pairwise sensitivity data correlation between cell-lines, or columns, across different studies (cell-wise) is considerably strong. This common interesting observation across multiple pharmacogenomics datasets suggests the existence of subtle consistency among the different studies (i.e., strong cell-wise correlation). However, significant noises are also shown (i.e., weak drug-wise correlation) and have prevented researchers from comfortably using the data directly. Motivated by this observation, we propose a novel framework for addressing the inconsistency between large-scale pharmacogenomics data sets. Our method can significantly boost the drug-wise correlation and can be easily applied to re-summarized and normalized datasets proposed by others. We also investigate our algorithm based on many different criteria to demonstrate that the corrected datasets are not only consistent, but also biologically meaningful. Eventually, we propose to extend our main algorithm into a framework, so that in the future when more datasets become publicly available, our framework can hopefully offer a "ground-truth" guidance for references.

Keywords: Pharmacogenomics Datasets; Precision Medicine; Biomarker Discovery

1. Introduction

One goal of precision medicine is to select optimal therapies for individual cancer patients based on individual molecular biomarkers identified from clinical trials.[1–3] Molecular biomarkers for many cancer drugs are currently quite limited, and it takes many years to identify and validate a biomarker for a single drug in clinical trials.[4,5] Recent pharmacogenomics studies, where drugs are tested against panels of molecularly characterized cancer cell lines, enabled large-scale identification of various types of molecular biomarkers by correlating drug sensitivity with molecular profiles of pre-treatment cancer cell lines.[6–10] These biomarkers are expected to predict the chance that cancer cells will respond to individual drugs.

There have been a handful of similar pharmacogenomic studies since Cancer Cell Line Encyclopedia (CCLE)[7] and Genomics of Cancer Genome Project (CGP)[11] were published in 2012 by the Broad Institute and Sanger Institute, respectively. CCLE included sensitivity data

for 1046 cell lines and 24 compounds; CGP included data for almost 700 cell lines and 138 compounds. The following Broad Institute's Cancer Therapeutics Response Portal (CTRPv2) dataset included 860 cell lines and 481 compounds.[8,12,13] The dataset from the Institute for Molecular Medicine Finland (FIMM) included 50 cell lines and 52 compounds.[14] The new version of Genomics of Drug Sensitivity in Cancer (GDSC1000) dataset included 1001 cell lines and 251 compounds. There have also been similar pharmacogenomics studies specific to particular cancers including acute myeloid leukemia.[15–17]

Each dataset is essentially a data matrix, where each row represents one drug, each column represent one cell line, and values are sensitivity measures derived from dose-response curves. IC50 (concentration at which the drug inhibited 50% of the maximum cellular growth) and AUC (area under the activity curve measuring dose response) are commonly used as sensitivity measures. However, recent re-investigation of published pharmacogenomics data has revealed the inconsistency of drug sensitivity data among different studies, raising the concern of using them for biomarker discovery.[18,19] In the recent comparison of drug sensitivity measures between CGP and CCLE for 15 drugs tested on the 471 shared cell lines, the vast majority of drugs yielded poor concordance (median Spearman's rank correlation of 0.28 and 0.35 for IC50 and AUC, respectively).[18]

There have been numerous attempts to address this issue. Mpindi et al. proposed to increase the consistency through harmonizing the readout and drug concentration range.[20] They re-analyzed the dose–response data using a standardized AUC response metric. They found high concordance between FIMM and CCLE and reasoned that similar experimental protocols were applied, including the same readout, similar controls. Bouhaddou et al. calculated a common viability metric across a shared log10-dose range, and computed slope, AUC values and found the new matrix could lead to better consistency.[21] Hafner et al. proposed another metric called GR50 to summarize drug sensitivity and demonstrated its superiority in assessing the effects of drugs in dividing cells.[22] Most proposed ideas focused on forming better summarization metric and/or standardizing experiments and data processing pipeline. Unfortunately, standardization methods cannot address the inconsistency issues of existing datasets. Re-summarization methods rely heavily on the assumption that the raw data is correct. But since datasets produced under similar experimental protocols are more consistent with each other, there surely exists some technical noises on the raw data.[20] Hence when the overlapping part between datasets grows bigger and the noise sources become more complex, these methods might not work well. Note that most of the studies have focused on the overlaps between CCLE and other datasets, which only contain very limited number of drugs. Novel computational methods correcting large-scale summarized data are therefore in urgent need.

Studies confirmed that drug-wise correlation is poor, but the cell-wise correlation is considerably strong (for example: overlapping cell lines between CTRPv2 and GDSC1000 have a median Spearman's correlation of 0.553), suggesting the underlying consistency of pharmacogenomics datasets. Inspired by this observation, we developed a novel computational method Alternating Imputation and Correction Method (AICM). Through purely correcting data based on their cell-wise correlation, AICM significantly improves the drug-wise correlation and hence makes the datasets more credible in future work. Furthermore, since AICM

works on summarized data, it can easily concatenate with all previous methods proposed to improve the summarization of raw data — just run on the re-summarized data. To the best of our knowledge, this is the first method that leverages cell-wise information into correcting data to address such challenge. We release the code and corrected datasets to the community[a].

2. Method

2.1. *Method overview*

The main goal is to increase the drug-wise correlation between two datasets, denoted as $A, B \in \mathbb{R}^{n\times p}$ — n drugs and p cell lines — for convenience. We denote the ith **row** of matrix A as $A_{[i,:]}$, then the goal can be formalized into the following problem:

$$\max_{f,g} \quad \sum_{i=1}^{n} \text{Corr}\left(f(A)_{[i,:]}, g(B)_{[i,:]}\right) \tag{1}$$

This is a more generalized idea than Renyi's correlation as we define f, g not functions but **operations** such that $f, g : \mathbb{R}^{n\times p} \to \mathbb{R}^{n'\times p'}$, where $n', p' \in \mathbb{Z}_+$. Operations include using a new summarization metric to re-summarize raw data and subsampling the data.

Now, since cell-wise correlation is consistently more concordant across different studies than drug-wise correlation, we can raise one natural question: can we rely on the cell-wise information to correct the datasets so that the drug-wise correlation will also be improved? We denote A^j as the jth **column** of A and A^J as the union of all column A^j such that $j \in J$, then more precisely, we want to develop some operation f, g such that

$$\max_{f,g} \quad \sum_{i=1}^{n} \text{Corr}\left(f(A|A^J, B^J)_{[i,:]}, g(B|A^J, B^J)_{[i,:]}\right) \quad J \subseteq \bigcup_{k=1}^{p}\{k\} \tag{2}$$

$$\text{s.t.} \quad \|f(A) - A\| \le \epsilon_A, \quad \|f(B) - B\| \le \epsilon_B \tag{3}$$

where $(\cdot|A^J, B^J), J \subseteq \bigcup_{k=1}^{p}\{k\}$ means either partial or all corresponding column information of A and B is given. $\|\cdot\|$ in (3) denotes an arbitrary matrix norm, and ϵ_A, ϵ_B are some arbitrary tolerance that we allow maximum departure from the original values. We have found that there are considerably large amount of missing data in these datasets. Surprisingly, with some simple linear regression based imputation of these missing data based solely on the cell-wise information, we found increase in drug-wise correlation. This confirmed our hypothesis that cell-wise information can be utilized to correct the datasets. Thus, AICM is developed to accomplish this goal by randomly dropping the parts of one dataset's column and re-fit based on another dataset's corresponding column with a simple linear regression with ℓ_∞ norm regularization. ℓ_∞ norm is leveraged to regularize large departure from the original data as it bounds the maximum departure of fitted values from original values. The corrected values are subject to a hard threshold assuming that the data are not completely destroyed by noises, so that the corrected data shall not depart too far from the original value. By repeating such regression process interactively between two datasets, AICM hopes to reveal the true information shared in between these datasets and hence increase the drug-wise consistency.

[a]https://github.com/tomwhoooo/aicm

2.2. *Algorithm*

The main idea is as described above: we uniformly randomly drop the values from one matrix (response matrix) and use the other matrix's column (variable matrix) to impute dropped values. We then threshold the imputed values into the final correction by some proportional threshold with respect to the original values of the response matrix. We iteratively repeat this process by swapping the role of response and variable between two matrices. Below are the hyperparameters for the algorithm:

- max iterations ($iter \in \mathbb{Z}_+$): how many iterations the alternating imputation and correction need to be run.
- dropping rate ($r \in (0, 1)$): what percent of the data from the response matrix should be dropped each iteration
- regularization term ($\lambda_r \in \mathbb{R}_+$): how much the original value should be taken into account during the regression process
- hard proportional constraint ($\lambda_h \in (0, 1)$): how many percentage points percent the imputed data can depart from the original value absolutely

And the full algorithm is described in detail as in Algorithm 1. We use a simple linear regression with ℓ_∞ norm (Eq 4) regularization for fitting process. Besides this, one can always use other fitting methods. For example, if one believes sparsity needs to be incorporated, one can use more weights and an ℓ_1 norm, or if one believes there needs to be some group effects across cell lines, one can use an ℓ_1 and ℓ_2 norm penalty. These ideas are similar to the idea of Lasso and Elastic Net.[23,24] However, it is suggested that the objective function of this fitting process should remain convex, since solving non-convex problems would highly likely lead to a local extrema (or even a saddle point) and thus cause disastrous variations among trials.

2.3. *Remarks*

Although the whole iterative procedures are not convex, the main objective function (4) is convex and hence the solution of this function would be a global minimum with an appropriate solver. Thus (4) can be solved efficiently and accurately by various methods such as proximal gradient algorithm and alternating direction of multipliers (ADMM).[25,26] They have well-established convergence theorems and are available in many open-source (i.e. SCS[27]) and industrial solvers.[28]

In the next section, we will show the results of our algorithm on real datasets, as well as synthetic datasets to demonstrate our method significantly increases drug-wise correlation remarkably and does not artificially increase the correlation under certain assumption. We will also show the result is indeed biologically meaningful.

3. Results and Discussion

3.1. *Synthetic datasets*

The alternative correction procedure (**Swap**) in AICM essentially agglomerates two datasets. It inevitably gives rise to the concern that the corrected datasets are forced to be similar

Algorithm 1 Alternating Imputation and Correction Method (AICM)

Hyperparameter: Dropping rate r, maximum iteration $iter$, regularization term λ_r, and hard constraint term λ_h.

Input: Two data matrices, of both n drugs and p cell-lines with summarized sensitivity data, denote as $A, B \in \mathbb{R}^{n\times p}$. We denote jth **column** of two matrices as a^j, b^j, $j \in \{1, 2, \ldots, p\}$ respectively. We denote the entry at ith row and jth column as A_{ij} and B_{ij} respectively, $\{i, j\} \in \{1, 2, \ldots, n\} \times \{1, 2, \ldots, p\}$.

Initialization: For each $j \in \{1, 2, \ldots, p\}$, for all $i \in \{1, 2, \ldots n\}$ such that B_{ij} is missing while A_{ij} is not, we denote such set as B^{NA}_{ij}, we fit a linear model such that α_j, β_j maximizes $\|b^j - \alpha_j a^j + \beta_j\|_2$ and then impute the missing values as $B^{\text{NA}}_{ij} = \alpha_j A_{ij} + \beta_j$. Then swap the role of A and B and repeat the above process. Now we have two matrices with same missing indices.

for k in $\{1, 2, \ldots Iter\}$ **do**

Swap: $A \to B, B \to A$.

Drop: Randomly drop $r \times n \times p$ data uniformly from A, we denote the indices of the dropped data as $\mathcal{D} \subseteq \{1, 2, \ldots, n\} \times \{1, 2, \ldots, p\}$, and hence dropped data as a set $A^{\text{DR}} := \left\{\bigcup_{\{i,j\}\in\mathcal{D}} A_{ij}\right\}$. In a similar fashion, we denote dropped data of **column** k as $a^k_{\text{DR}} := \left\{\bigcup_{\{i,j\}\in\mathcal{D},\forall i \text{ s.t. } j=k} A_{ij}\right\}$, we denote the corresponding data in kth column of B as b^k_{ADR}. We fit a set of parameters $\alpha_j \in \mathbb{R}, \beta_j \in \mathbb{R}$ for each j with the following objective function:

$$\min_{\alpha_j,\beta_j} \quad \frac{1}{n}\|b^j - (\alpha_j a^j + \beta_j)\|_2 + \lambda_r \|a^j_{\text{DR}} - (\alpha_j b^j_{\text{ADR}} + \beta_j)\|_\infty \tag{4}$$

Correction: Set $a^j_{\text{DR}} = \alpha_j b^j_{\text{ADR}} + \beta_j$ for each j. We denote the set of corrected value as $\{A^{\text{IMP}}\} = \bigcup_{j=1}^{p} \{a^j_{\text{DR}}\}$.

Threshold: For $\{i, j\} \in \mathcal{D}$, we set $\{A^{\text{IMP}}\}_{ij}$ to

$$\{A^{\text{IMP}}\}_{ij} = \max\left(\min\left(A_{ij}, (1-\lambda_h)A_{ij}\right), (1+\lambda_h)A_{ij}\right) \tag{5}$$

end for

regardless of the ground truth. For example, one easily questions whether AICM improves the between-group correlation of placebo – it functions as white noise, thus is expected to be uncorrelated between one dataset and another. In addition, the induced randomness (**Drop**) in AICM might well shake one's confidence in the stability and reliability of this method. In this section, we utilize synthetic datasets to demonstrate that AICM are free of these hypothetical troubles.

In the most ideal scenario, where there exist no technical or biological noises, the drug sensitivity matrices are expected to be the same across distinct research teams. For simplicity, we assume that the ground truth can be separated into the drug part and the cell part. Then, the observed matrix can be modelled as

$$M = \alpha \mathbf{1} \cdot \mathbf{1}^T + \mathbf{a} \cdot \mathbf{b}^T + W, \tag{6}$$

where α is the baseline, $\mathbf{a} \in \mathbb{R}^n$ contains the information about the n drugs, $\mathbf{b} \in \mathbb{R}^p$ summarizes

the structure of the cell lines. The matrix $\alpha \mathbf{1} \cdot \mathbf{1}^T + \mathbf{a} \cdot \mathbf{b}^T$ represents the ground truth of the drug sensitivities. We simulate the ineffective drugs as uncorrelated rows by setting the top m entries of $\mathbf{a}$ to 0's while the other rows associated with non-zero values (hence correlated) in $\mathbf{a}$ are regarded as effective drugs. $W \in \mathbb{R}^{n \times p}$ is a random matrix from a matrix normal distribution which reflects the composite of noise. In this study, we set $n = 50$, $p = 40$, $m = 10$. The details of the data generation process are deferred to supplementary material[b].

We apply AICM to the synthetic datasets with 30 different combinations of hyperparameters *iter* and λ_h: $iter \in \{20, 40, 80, 100, 120, 140\}$ and $\lambda_h \in \{0.05, 0.1, 0.15, 0.2, 0.25\}$, and repeat the method for 20 times for each combination. With careful selection, we take $(iter, \lambda_h) = (80, 0.1)$ because this combination gives acceptable reduction on correlations between first ten uncorrelated rows and strong increase of correlations between correlated rows as demonstrated (see Figure 1). In addition, $\lambda_h = 0.1$ is a conservative control of the correction step. Note that the normalized distances between the two matrices and the ground truth are reduced to 1.188 and 1.170 respectively after correction (the distances are 1.272 and 1.267 before correction). The decrease in distance is relatively significant, given the fact that we put a hard proportional threshold at 10% for each individual value. Therefore, AICM does help reduce the noise in the observed matrices. Furthermore, the Spearman's correlation median of the correlated rows is increased to 0.390 from 0.219 with standard deviation 0.021, while the Spearman's correlation median of uncorrelated rows is reduced to 0.084 from 0.095 with standard deviation 0.010. It indicates that the result is insensitive to the randomness of the dropping procedure in AICM. In Figure 2, the actual shift of the correlation distributions is displayed. On top of incremental correlations of correlated rows, there appear to be reduced correlations of uncorrelated rows after using AICM. It implies that our method not only enhances the real signals, but also exposes the fake ones. Thus, the original concern is eliminated on indiscriminately blending signals between datasets.

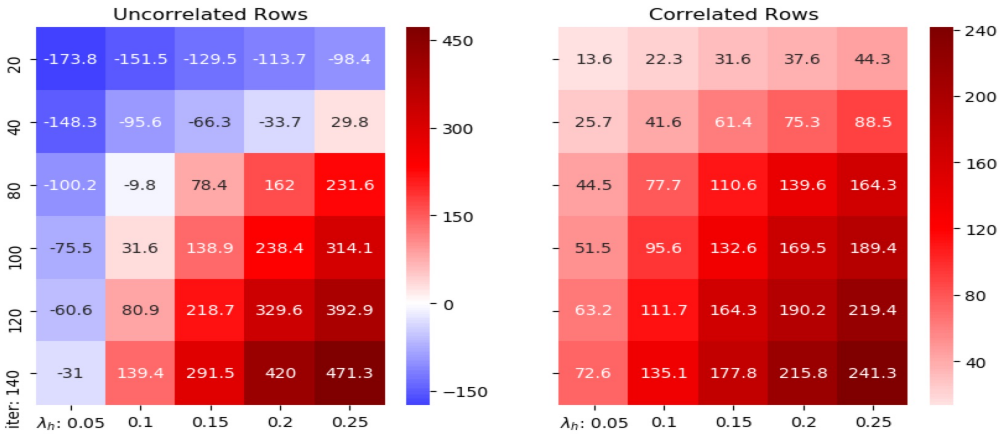


Fig. 1: The percentage change (%) of the medians of the correlations on synthetic datasets with different parameters. x-axis is *iter* and y-axis is λ_h.

[b]https://github.com/tomwhoooo/aicm/blob/master/paper_supp

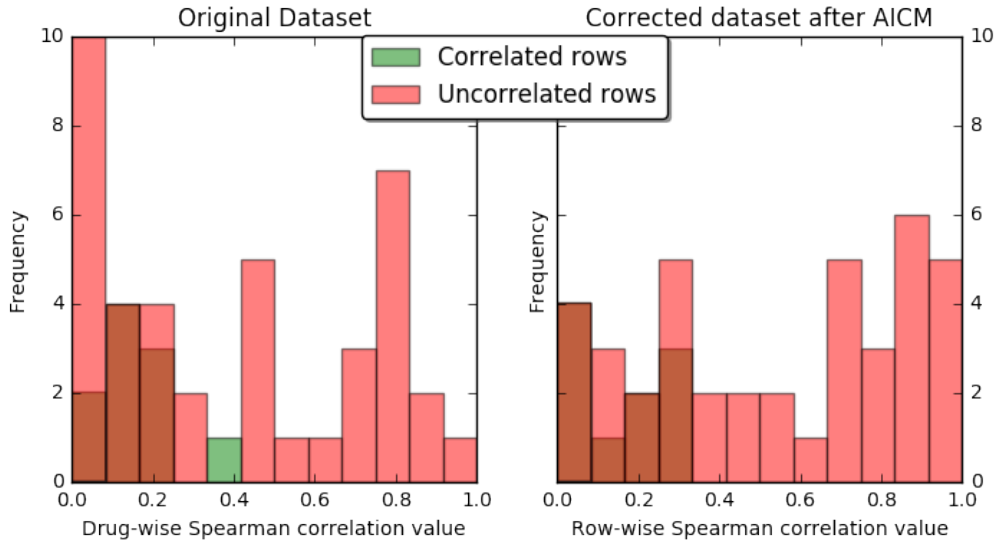


Fig. 2: Distribution of drug-wise correlations between the synthetic datasets before AICM is applied and after. *Note that the darker green bars denote overlap of uncorrelated rows and correlated rows in this histogram.*

3.2. *Real datasets*

We choose the three largest datasets in PharmacoGX: CTRPv2, GDSC1000, and FIMM as case studies.[8,11,13,19] Drug names are compared by first converting to InChIKey via the webchem R package.[29] For the GDSC1000 dataset, 60 InChIKeys are subsequently manually retrieved from PubChem. A Python script is prepared and used to retrieve generic cell line "Accession numbers" from Cellosaurus.[30] Given that not all cell lines returned Accession numbers, we remove symbols, spaces, and case from the names of the remaining cell lines for improved matching between datasets. For each of the three datasets, their respective IC50 and AUC data are obtained from PharmacoGx. Duplicate experiments are removed from CTRPv2 and GDSC1000 by removing all instances of a certain culture medium. Finally, the six dataframes are filtered for matching cell lines and drugs between each other, yielding 12 dataframes which contain IC50 and AUC between all 3 datasets.

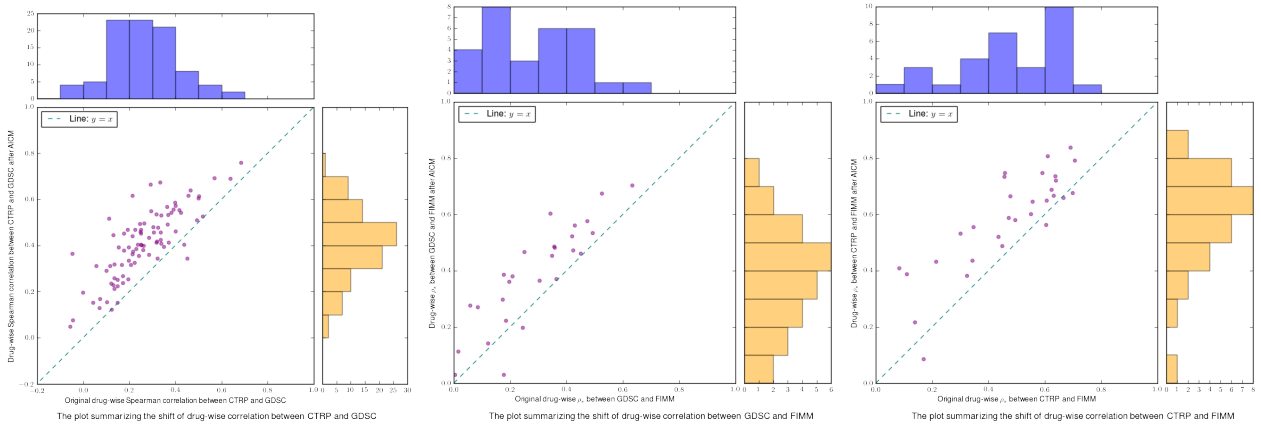
With the optimal hyperparameters fetched from synthetic data, we demonstrate the shift of Spearman's correlation between 90 drugs overlapping between GDSC1000 and CTRPv2 after AICM is deployed in Figure 3a. The data uses AUC summarization. It is clear that after AICM is deployed, the two datasets become more concordant with each other — this can be observed from both individual drug scatter plot and overall distribution. We also demonstrate two similar graphs between 30 overlapping drugs between CTRPv2 and FIMM, 29 overlapping drugs between GDSC1000 and FIMM with AUC summarization in Figure 3b and 3c.

Note that when we calculate the correlation, the original values that are missing are discarded from both matrices for fair comparison. Brief statistics of the original and post-correction drug-wise Spearman's correlation can be found in Table 1. For significance, we

used the cutoff of one-sided test at p-value 0.05 using the significance test of Spearman's correlation proposed by Jerrold Zar.[31] The values present what percentage of drugs is significant across two datasets.

Datasets	Mean		Median		Significant		Size	
	Before	After	Before	After	Before	After	Drug	Cell
CTRPv2 & GDSC1000	0.261	0.410	0.249	0.411	63.33%	90.00%	90	566
CTRPv2 & FIMM	0.485	0.624	0.468	0.585	70.00%	93.33%	30	41
GDSC1000 & FIMM	0.250	0.352	0.278	0.380	27.59%	55.17%	29	47

Table 1: Brief statistics of the original and post-correction drug-wise Spearman's correlation

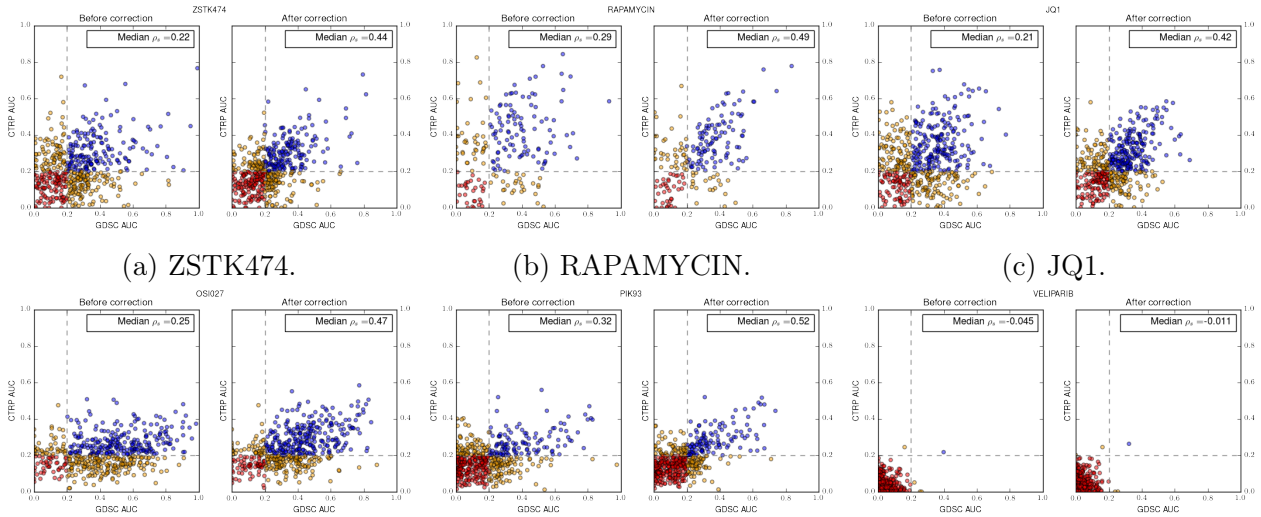


(a) GDSC vs CTRP (b) GDSC vs FIMM (c) CTRP vs FIMM

Fig. 3: The shift of Spearman's correlation, both individually and as a distribution, of common drugs between specified datasets before and after AICM is run.

We demonstrate the scatter plots of some individual drug's effect on cell lines before and after AICM correction in Figure 4, we can indeed see the scatter plots become more concordant across datasets. We color the plots in a similar fashion as Safikhani et al.: we use blue (sensitive) to denote both datasets ≥ 0.2 and red (resistant) for both ≤ 0.2; orange denotes inconsistency.[19] We pay particular interest to drugs that show significant improvement and drugs that show little improvement. We can see that drugs such as ZSTK474. Rapamycin, JQ1, OSI027 and PIK93 show significant improvement. Although Velaparib shows little improvement, it is known to be a very selective PARP inhibitor; it is not effective in any of cancer cell lines examined in this study. Thus it would be meaningless and artificial to increase the correlation across two datasets.

We also present the scatter plots of some drugs shared by all three datasets: CTRPv2, GDSC1000 and FIMM. We can see that in both 5a and 5b, the two graphs on the right consistently demonstrate more similar pattern than the two graphs on the left, which confirms that the variation across multiple datasets is alleviated after AICM is deployed – AICM indeed

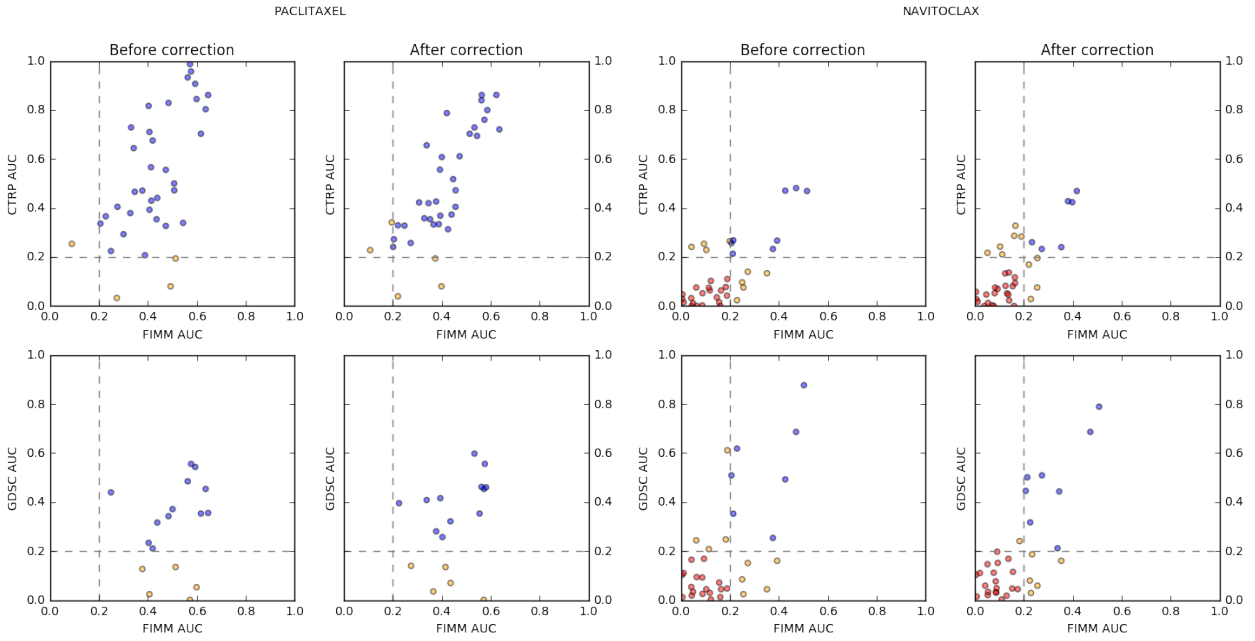


(a) ZSTK474. (b) RAPAMYCIN. (c) JQ1.

(d) OSI027. (e) PIK93. (f) Velaparib.

Fig. 4: Individual drugs with respect to individual cell lines before and after AICM is deployed. First five demonstrate drugs whose correlations are significantly improved and the last one demonstrates a drug whose correlation is poorly improved.

recovers some meaningful signals.



(a) Drug Paclitaxel. (b) Drug Navitoclax.

Fig. 5: Overlapping drugs across three datasets.

4. Conclusions and Future Work

In this work, we develop a genuine algorithm by alternatively dropping and fitting cell-wise data and succeeds in improving the drug-wise correlation. The algorithm is flexible to incorporate different ideas. For example, one can replace the fitting process with other regression methods if one had different assumptions in mind. We have shown that with appropriate hyperparameters chosen, AICM can improve the drug-wise correlation across different studies and that the increase in correlation is indeed concordant and biologically meaningful.

We realize the limitation of AICM's dependence on the overlapping of existing data, while such data is rather rare. We did not include experiment on CCLE dataset primarily because it has very limited drug overlap with other existing datasets. Also, AICM currently does not purport to correct sensitivity data of new drugs. Future work will be to extend such algorithm into a complete framework. AICM is able to scale to reasonable amount of datasets. When a new dataset is coming in, say X, we can conduct AICM procedure between this dataset and each existing dataset, say $Y_1, Y_2, \ldots Y_n$, yield n corrected datasets, $\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_n$. Afterward, we can do an average on corrected to specify the corrected new dataset, i.e. $\tilde{X} = \frac{1}{n}\sum_{i=1}^{n} \tilde{X}_i$. We will maintain a database of corrected existing drugs and cells, and when more data comes in, we will be able to incorporate it. We hope as more data comes in, the database would asymptotically become more accurate of reflecting true relationship between drugs and cell lines and can thus serve as a ground-truth guidance. As for new drugs, we will develop either a generative algorithm or a clustering algorithm, i.e. getting the latent distribution where drug is "generated" or cluster it based on existing features, and find similar existing drugs in hope of some practical guidance. We believe our corrected datasets will facilitate biomarker discovery.

Acknowledgments

This work is supported by R21 TR001743 and K01 ES028047 and the MSU Global Impact Initiative. We would thank Anthony Sciarini for providing the pipeline to fetch the cell-line generic names. We would also thank Ryan Lovett and Chris Paciorek for all helps received on cluster computing issues.

References

1. F. S. Collins and H. Varmus. A new initiative on precision medicine. *N. Engl. J. Med.*, 372(9):793–795, Feb 2015.
2. D. R. Lowy and F. S. Collins. Aiming High–Changing the Trajectory for Cancer. *N. Engl. J. Med.*, 374(20):1901–1904, May 2016.
3. B. Chen and A. J. Butte. Leveraging big data to transform target selection and drug discovery. *Clin. Pharmacol. Ther.*, 99(3):285–297, Mar 2016.
4. G. Yothers, M. J. O'Connell, M. Lee, M. Lopatin, K. M. Clark-Langone, C. Millward, S. Paik, S. Sharif, S. Shak, and N. Wolmark. Validation of the 12-gene colon cancer recurrence score in NSABP C-07 as a predictor of recurrence in patients with stage II and III colon cancer treated with fluorouracil and leucovorin (FU/LV) and FU/LV plus oxaliplatin. *J. Clin. Oncol.*, 31(36):4512–4519, Dec 2013.
5. A. de Gramont, S. Watson, L. M. Ellis, J. Rodon, J. Tabernero, A. de Gramont, and S. R.

Hamilton. Pragmatic issues in biomarker evaluation for targeted therapies in cancer. *Nat Rev Clin Oncol*, 12(4):197–212, Apr 2015.

6. M. J. Garnett and et al. Systematic identification of genomic markers of drug sensitivity in cancer cells. *Nature*, 483(7391):570–575, Mar 2012.
7. J. Barretina, G. Caponigro, N. Stransky, K. Venkatesan, A. A. Margolin, S. Kim, C. J. Wilson, J. Lehar, G. V. Kryukov, D. Sonkin, A. Reddy, M. Liu, L. Murray, M. F. Berger, J. E. Monahan, P. Morais, J. Meltzer, A. Korejwa, J. Jane-Valbuena, F. A. Mapa, J. Thibault, E. Bric-Furlong, P. Raman, A. Shipway, I. H. Engels, J. Cheng, G. K. Yu, J. Yu, P. Aspesi, M. de Silva, K. Jagtap, M. D. Jones, L. Wang, C. Hatton, E. Palescandolo, S. Gupta, S. Mahan, C. Sougnez, R. C. Onofrio, T. Liefeld, L. MacConaill, W. Winckler, M. Reich, N. Li, J. P. Mesirov, S. B. Gabriel, G. Getz, K. Ardlie, V. Chan, V. E. Myer, B. L. Weber, J. Porter, M. Warmuth, P. Finan, J. L. Harris, M. Meyerson, T. R. Golub, M. P. Morrissey, W. R. Sellers, R. Schlegel, and L. A. Garraway. The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603–607, Mar 2012.
8. A. Basu, N. E. Bodycombe, J. H. Cheah, E. V. Price, K. Liu, G. I. Schaefer, R. Y. Ebright, M. L. Stewart, D. Ito, S. Wang, A. L. Bracha, T. Liefeld, M. Wawer, J. C. Gilbert, A. J. Wilson, N. Stransky, G. V. Kryukov, V. Dancik, J. Barretina, L. A. Garraway, C. S. Hon, B. Munoz, J. A. Bittker, B. R. Stockwell, D. Khabele, A. M. Stern, P. A. Clemons, A. F. Shamji, and S. L. Schreiber. An interactive resource to identify cancer genetic and lineage dependencies targeted by small molecules. *Cell*, 154(5):1151–1161, Aug 2013.
9. F. Iorio, T. A. Knijnenburg, D. J. Vis, G. R. Bignell, M. P. Menden, M. Schubert, N. Aben, E. Goncalves, S. Barthorpe, H. Lightfoot, T. Cokelaer, P. Greninger, E. van Dyk, H. Chang, H. de Silva, H. Heyn, X. Deng, R. K. Egan, Q. Liu, T. Mironenko, X. Mitropoulos, L. Richardson, J. Wang, T. Zhang, S. Moran, S. Sayols, M. Soleimani, D. Tamborero, N. Lopez-Bigas, P. Ross-Macdonald, M. Esteller, N. S. Gray, D. A. Haber, M. R. Stratton, C. H. Benes, L. F. A. Wessels, J. Saez-Rodriguez, U. McDermott, and M. J. Garnett. A Landscape of Pharmacogenomic Interactions in Cancer. *Cell*, 166(3):740–754, Jul 2016.
10. M. Niepel, M. Hafner, E. A. Pace, M. Chung, D. H. Chai, L. Zhou, B. Schoeberl, and P. K. Sorger. Profiles of Basal and stimulated receptor signaling networks predict drug response in breast cancer lines. *Sci Signal*, 6(294):ra84, Sep 2013.
11. W. Yang, J. Soares, P. Greninger, E. J. Edelman, H. Lightfoot, S. Forbes, N. Bindal, D. Beare, J. A. Smith, I. R. Thompson, S. Ramaswamy, P. A. Futreal, D. A. Haber, M. R. Stratton, C. Benes, U. McDermott, and M. J. Garnett. Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic Acids Res.*, 41(Database issue):D955–961, Jan 2013.
12. M. G. Rees, B. Seashore-Ludlow, J. H. Cheah, D. J. Adams, E. V. Price, S. Gill, S. Javaid, M. E. Coletti, V. L. Jones, N. E. Bodycombe, C. K. Soule, B. Alexander, A. Li, P. Montgomery, J. D. Kotz, C. S. Hon, B. Munoz, T. Liefeld, V. Dan?ik, D. A. Haber, C. B. Clish, J. A. Bittker, M. Palmer, B. K. Wagner, P. A. Clemons, A. F. Shamji, and S. L. Schreiber. Correlating chemical sensitivity and basal gene expression reveals mechanism of action. *Nat. Chem. Biol.*, 12(2):109–116, Feb 2016.
13. B. Seashore-Ludlow, M. G. Rees, J. H. Cheah, M. Cokol, E. V. Price, M. E. Coletti, V. Jones, N. E. Bodycombe, C. K. Soule, J. Gould, B. Alexander, A. Li, P. Montgomery, M. J. Wawer, N. Kuru, J. D. Kotz, C. S. Hon, B. Munoz, T. Liefeld, V. Dan?ik, J. A. Bittker, M. Palmer, J. E. Bradner, A. F. Shamji, P. A. Clemons, and S. L. Schreiber. Harnessing Connectivity in a Large-Scale Small-Molecule Sensitivity Dataset. *Cancer Discov*, 5(11):1210–1223, Nov 2015.
14. B. Yadav, T. Pemovska, A. Szwajda, E. Kulesskiy, M. Kontro, R. Karjalainen, M. M. Majumder, D. Malani, A. Murumagi, J. Knowles, K. Porkka, C. Heckman, O. Kallioniemi, K. Wennerberg, and T. Aittokallio. Quantitative scoring of differential drug sensitivity for individually optimized

anticancer therapies. *Sci Rep*, 4:5193, Jun 2014.
15. R. Marcotte, A. Sayad, K. R. Brown, F. Sanchez-Garcia, J. Reimand, M. Haider, C. Virtanen, J. E. Bradner, G. D. Bader, G. B. Mills, D. Pe'er, J. Moffat, and B. G. Neel. Functional Genomic Landscape of Human Breast Cancer Drivers, Vulnerabilities, and Resistance. *Cell*, 164(1-2):293–309, Jan 2016.
16. A. Daemen, O. L. Griffith, L. M. Heiser, N. J. Wang, O. M. Enache, Z. Sanborn, F. Pepin, S. Durinck, J. E. Korkola, M. Griffith, J. S. Hur, N. Huh, J. Chung, L. Cope, M. J. Fackler, C. Umbricht, S. Sukumar, P. Seth, V. P. Sukhatme, L. R. Jakkula, Y. Lu, G. B. Mills, R. J. Cho, E. A. Collisson, L. J. van't Veer, P. T. Spellman, and J. W. Gray. Modeling precision treatment of breast cancer. *Genome Biol.*, 14(10):R110, 2013.
17. S. I. Lee, S. Celik, B. A. Logsdon, S. M. Lundberg, T. J. Martins, V. G. Oehler, E. H. Estey, C. P. Miller, S. Chien, J. Dai, A. Saxena, C. A. Blau, and P. S. Becker. A machine learning approach to integrate big data for precision medicine in acute myeloid leukemia. *Nat Commun*, 9(1):42, 01 2018.
18. B. Haibe-Kains, N. El-Hachem, N. J. Birkbak, A. C. Jin, A. H. Beck, H. J. Aerts, and J. Quackenbush. Inconsistency in large pharmacogenomic studies. *Nature*, 504(7480):389–393, Dec 2013.
19. Z. Safikhani, P. Smirnov, M. Freeman, N. El-Hachem, A. She, Q. Rene, A. Goldenberg, N. J. Birkbak, C. Hatzis, L. Shi, A. H. Beck, H. J. W. L. Aerts, J. Quackenbush, and B. Haibe-Kains. Revisiting inconsistency in large pharmacogenomic studies. *F1000Res*, 5:2333, 2016.
20. John Patrick Mpindi, Bhagwan Yadav, Päivi Östling, Prson Gautam, Disha Malani, Astrid Murumägi, Akira Hirasawa, Sara Kangaspeska, Krister Wennerberg, Olli Kallioniemi, and Tero Aittokallio. Consistency in drug response profiling. *Nature*, 540:E5 EP –, 11 2016.
21. Mehdi Bouhaddou, Matthew S. DiStefano, Eric A. Riesel, Emilce Carrasco, Hadassa Y. Holzapfel, DeAnalisa C. Jones, Gregory R. Smith, Alan D. Stern, Sulaiman S. Somani, T. Victoria Thompson, and Marc R. Birtwistle. Drug response consistency in ccle and cgp. *Nature*, 540:E9 EP –, 11 2016.
22. M. Hafner, M. Niepel, M. Chung, and P. K. Sorger. Growth rate inhibition metrics correct for confounders in measuring sensitivity to cancer drugs. *Nat. Methods*, 13(6):521–527, 06 2016.
23. Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.
24. Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.
25. Neal Parikh and Stephen Boyd. Proximal algorithms. *Found. Trends Optim.*, 1(3):127–239, January 2014.
26. Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011.
27. B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. SCS: Splitting conic solver, version 2.0.2, November 2017.
28. Tianyi Lin, Shiqian Ma, and Shuzhong Zhang. On the global linear convergence of the admm with multiblock variables. *SIAM Journal on Optimization*, 25:1478–1497, 2015.
29. George Nicola, Tiqing Liu, and Michael K. Gilson. Public domain databases for medicinal chemistry. *Journal of Medicinal Chemistry*, 55(16):6987–7002, 2012. PMID: 22731701.
30. A. Bairoch. The Cellosaurus, a Cell-Line Knowledge Resource. *J Biomol Tech*, May 2018.
31. Jerrold H. Zar. Significance testing of the spearman rank correlation coefficient. *Journal of the American Statistical Association*, 67:578–580, 1972.

Outgroup Machine Learning Approach Identifies Single Nucleotide Variants in Noncoding DNA Associated with Autism Spectrum Disorder

Maya Varma[1], Kelley Marie Paskov[2], Jae-Yoon Jung[2, 5], Brianna Sierra Chrisman[3], Nate Tyler Stockham[4], Peter Yigitcan Washington[3], Dennis Paul Wall [2, 5*]

Departments of Computer Science[1], Biomedical Data Science[2], Bioengineering[3], Neuroscience[4] and Pediatrics[5], Stanford University
Stanford, CA 94305, USA
Email: dpwall@stanford.edu

Autism spectrum disorder (ASD) is a heritable neurodevelopmental disorder affecting 1 in 59 children. While noncoding genetic variation has been shown to play a major role in many complex disorders, the contribution of these regions to ASD susceptibility remains unclear. Genetic analyses of ASD typically use unaffected family members as controls; however, we hypothesize that this method does not effectively elevate variant signal in the noncoding region due to family members having subclinical phenotypes arising from common genetic mechanisms. In this study, we use a separate, unrelated outgroup of individuals with progressive supranuclear palsy (PSP), a neurodegenerative condition with no known etiological overlap with ASD, as a control population. We use whole genome sequencing data from a large cohort of 2182 children with ASD and 379 controls with PSP, sequenced at the same facility with the same machines and variant calling pipeline, in order to investigate the role of noncoding variation in the ASD phenotype. We analyze seven major types of noncoding variants: microRNAs, human accelerated regions, hypersensitive sites, transcription factor binding sites, DNA repeat sequences, simple repeat sequences, and CpG islands. After identifying and removing batch effects between the two groups, we trained an ℓ_1-regularized logistic regression classifier to predict ASD status from each set of variants. The classifier trained on simple repeat sequences performed well on a held-out test set (AUC-ROC = 0.960); this classifier was also able to differentiate ASD cases from controls when applied to a completely independent dataset (AUC-ROC = 0.960). This suggests that variation in simple repeat regions is predictive of the ASD phenotype and may contribute to ASD risk. Our results show the importance of the noncoding region and the utility of independent control groups in effectively linking genetic variation to disease phenotype for complex disorders.

Keywords: Autism Spectrum Disorder; noncoding region; tissue-specific microRNAs; human accelerated regions; hypersensitive sites; transcription factor binding sites; DNA repeat sequences; simple repeat sequences; CpG islands; batch effects

*Corresponding author

1. Introduction

Autism spectrum disorder (ASD) is a neurodevelopmental disorder characterized by social impairments, communication difficulties, and restricted and repetitive patterns of behavior. ASD usually manifests in infants and children and presents a wide range of symptoms that vary from person to person. Currently, 1 in 59 children in the United States are affected, and prevalence rates are expected to increase drastically over the next decade.[1] ASD is known to be highly genetic with a concordance rate between monozygotic twins of 77-99%.[2,3] The genetic architecture of the disorder is known to be complex, with an estimated 1000 genes involved in disease susceptibility, spanning common, rare, and de novo variants.[4,5]

Models exploring the genetic basis of ASD typically focus on protein-coding genes; however, coding sequences account for only 1.5% of human DNA. The remaining segments of DNA are comprised of noncoding regions, which have been shown to play an important role in many genetic disorders. For example, recessive mutations in the PTF1A gene enhancer can cause pancreatic agenesis,[6] a common mutation in the RET enhancer increases risk for Hirschprung disease,[7] and mutations in topologically associating chromatin domains can cause limb malformation.[8] Furthermore, a meta-analysis of over a thousand genetic association studies showed that most of the disease-associated single nucleotide variants identified by genome wide association studies (GWAS) lie in the noncoding region.[9]

However, the contribution of noncoding variants to ASD still remains unclear. A recent analysis of whole genome sequences of 516 children with ASD and their unaffected family members concluded that individuals with ASD tend to have significantly more de novo mutations in noncoding regions. The study evaluated two noncoding regions: untranslated regions (UTRs) of genes and conserved transcription factor binding sites that map to sites of DNase I hypersensitivity.[10] However, a separate evaluation of the same dataset concluded that although individuals with ASD possessed a small excess of de novo mutations in noncoding regions, there were no significant results across over 50,000 regulatory classes after multiple testing correction.[11]

As shown by these studies, population genetic analyses typically classify unaffected family members as controls. However, we hypothesize that this assumption does not effectively elevate variant signal from the genome for ASD cohorts. For example, close relatives of individuals with ASD often exhibit autistic behaviors, such as social deficits and delayed speech.[12,13] Thus, it is possible that family members possess a subclinical phenotype of ASD that may arise from genomic features shared with their affected children. Also, the diagnostic criteria for ASD were modified in 2013 with the release of the fifth edition of the Diagnostic and Statistical Manual of Mental Disorders. Most parents would have been evaluated using an earlier version of diagnostic criteria, making it possible that some would qualify for an ASD diagnosis by modern clinical standards.

In order to address this issue and to exacerbate signal in the noncoding region, we introduce a separate outgroup of patients with progressive supranuclear palsy (PSP), a neurodegenerative condition that causes difficulty with movement and thought.[14] We chose this group of control patients because there is no known etiological overlap or comorbidity between PSP and ASD, and PSP is generally not heritable. There are some familial cases caused by a mutation

in at least one copy of the gene MAPT on chromosome 17, but this is the only gene currently known to be linked with PSP.[15] No patients in the control group exhibit symptoms of ASD. In this work, we use whole genome sequencing data from 2182 children with ASD and 379 PSP controls to investigate the role of noncoding variants in ASD susceptibility.

This study focuses on seven major noncoding regions: tissue specific microRNAs, human accelerated regions, hypersensitive sites, transcription factor binding sites, DNA repeat sequences, simple repeat sequences, and CpG islands. *Tissue-specific microRNAs* play important roles in the regulation of mRNA expression and the development of neurons, and recent studies have implicated a total of 219 microRNAs in the development of ASD.[16] *Human accelerated regions*, which consist of only 49 highly-conserved segments in DNA, have been shown to regulate neural activity, with de novo copy number variations in these regions enriched in individuals with ASD.[17] *Hypersensitive sites* are regulatory regions that are sensitive to cleavage by nucleases, and de novo mutations in these regions are significantly enriched in ASD probands.[18] *Transcription-factor binding sites* are located in the noncoding regions of genes and assist in the regulation of transcription; variants in binding sites in MEGF10 and TCF4 have been associated with ASD and other intellectual disabilities.[19,20] *DNA Repeat sequences* and *simple repeat sequences* are sequences of repeating base pairs, distinguished by the length of the repeating pattern, that have been linked to neuronal differentiation and brain development.[21] Finally, *CpG islands*, which consist of regions with high frequencies of the cytosine and guanine base pairs, can have higher rates of methylation in individuals with ASD.[22]

2. Methods

2.1. *Data and Preprocessing*

We analyzed 30x-coverage whole genome sequencing data from the Hartwell Foundation's Autism Research and Technology Initiative (iHART); iHART has amassed data from 1006 multiplex families, each with at least two ASD-affected children. We also analyzed 30x-coverage whole genome sequencing data from 379 patients diagnosed with PSP. In order to limit batch effects due to inconsistencies in sequencing methodologies, we sequenced both populations at the New York Genome Center with Illumina HiSeq X instruments and utilized the same GATK variant calling pipeline; in addition, there is no sample overlap between the cohorts.

Chromosome coordinate lists for the seven noncoding regions were downloaded from the UCSC Genome Browser and the Regulatory Elements Database.[23,24] Quality control was performed on the variant call format (VCF) files by removing all variants with high excess heterozygosity scores, which typically indicate sequencing artifacts or consanguinity within the population. We then filtered the variant-call format files to extract all variants within these regions that were present in both the PSP and ASD populations. We also removed all variants with a large proportion (greater than 20%) of missing sites.

2.2. *Accounting for Batch Effects*

Batch effects present a major challenge when combining whole genome sequencing data across cohorts, resulting in many false positive associations.[25] Batch effects can result from almost

any step in the whole genome sequencing procedure, including library preparation, sequencing machine or center, sequencing depth, and variant calling pipelines.[26] Several methods have been developed to mitigate these effects, but these procedures focus on reducing batch effects for datasets collected and analyzed independently.[27,28] In our case, care was taken to sequence our ASD case and PSP control samples at the same center with the same platform and to analyze them using identical variant calling pipelines. In order to detect the more subtle batch effects that may remain, we expand on the method used by the UK10K project, detecting batch effects using a genome-wide association test with batch (ASD and PSP) as the phenotype.[29] To do this, we performed a chi-squared test for each variant, comparing the number of individuals with homozygous reference, heterozygous, homozygous alternate, and missing genotypes between the two datasets. Any variants with a batch association p-value below 0.05 after applying a Bonferroni multiple testing correction were discarded, resulting in the removal of approximately 5% of variants. Figure 1 shows the number of variants within each region that passed our preprocessing and batch effect filters.

Tissue-Specific miRNA	Human Accelerated Regions	Hypersensitive Sites	Transcription Factor Binding Sites	DNA Repeat Sequences	Simple Repeat Sequences	CpG Islands
1564	647	577,900	325,003	684,487	232,193	168,953

Fig. 1. Number of noncoding variants of each type after applying preprocessing filters and removing variants affected by batch effects.

2.3. *Feature Representation and Logistic Regression Classifier*

We designed a machine learning approach to determine if variation within noncoding regions could be utilized to predict ASD. In order to capture variant information from both the ASD and PSP populations, we constructed binary feature matrices for each of the seven noncoding regions. Each matrix includes 2561 rows corresponding to the 379 PSP control patients and 2182 ASD case patients; the columns represent the variants (shown in Fig. 1) associated with the region. We set each cell of the matrix as 1 if the individual expressed an alteration at the variant site (either heterozygous or homozygous alternate) and as 0 if the variant matched the reference sequence. Since several of these feature matrices included over one billion elements, all matrices were encoded in a customized sparse representation to ensure that machine learning would remain computationally tractable.

We created a logistic regression classifier with ℓ_1 regularization in order to encourage the use of the smallest possible number of relevant features. 80% of the individuals in the dataset were randomly selected for inclusion in the training set, and the remaining 20% were added to the held-out test set; train and test sets were divided by family, so there is no familial overlap between sets. In order to address class imbalance between the case and control populations, we adjusted classifier weights such that they are inversely proportional to class sizes. We ran 5-fold cross validation in order to tune the level of regularization (represented by λ). Then,

we evaluated performance on the held-out test set by measuring F_1 scores, precision, recall, and AUC-ROC.

We extracted the top-ranked variants from each of the seven noncoding regions for further analysis by selecting the five variants from each classifier with the highest positive regression coefficient values as well as the five variants with the lowest negative coefficient values. We also confirmed that these variants were highly-ranked across multiple folds in our cross-validation tests.

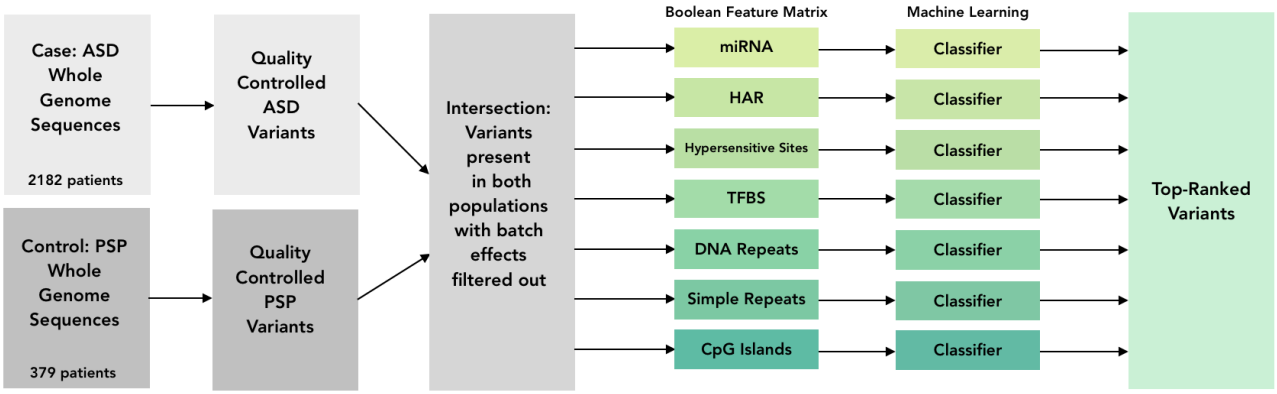


Fig. 2. *Machine learning pipeline.* Variants were called separately for cases and controls. The variant calls were then merged and a batch-effect filter was applied. Feature matrices were created for each of the seven noncoding regions and served as input to ℓ_1-regularized logistic regression classifiers. Finally, the top-ranked features were extracted from each classifier.

2.4. *Validation*

We validated the performance of our classifier using a held-out test set composed of 20% of the individuals from both cohorts. To demonstrate that our classifier can generalize, we also measured performance of our trained models on a completely independent cohort consisting of 517 ASD patients from the Simons Simplex Collection[30] and 2054 control individuals from the 1000 Genomes Project.[31] These cohorts were sequenced at different depths on different machines; however, the same GATK variant calling pipeline was utilized. We use this cohort to show that our classifier can effectively generalize to new populations and that we have adequately addressed batch effects in our training data.

Next, we devised a bootstrap test in order to determine if the seven groups of features used in this analysis were relevant predictors of ASD status when compared to random variants. To do so, we randomly sampled from the set of variants called in both the PSP and ASD cohorts. Feature matrices were designed according to the same procedures outlined in sections 2.1 and 2.2, and classifiers were trained on the random variants using the procedure outlined in section 2.3. This process was repeated between 20 and 100 times to obtain 95% confidence intervals. We ran separate bootstrap tests using different numbers of variants in order to account for the wide range in sizes of our variant sets; bootstrap test sizes range from 10^2 to 10^6 variants.

We also ran several tests to ensure that our logistic regression classifier was not biased by

population stratification. Ethnicity is responsible for much of the variation in human genomes, so to ensure that population substructure was not confounding our results, we examined performance separately for Europeans and non-Europeans in our test set. Autism is also sex-biased, with males about 4 times more likely to be affected than females; in order to verify that our results are robust to differences in the sex chromosomes, we also examined test performance on males and females separately.

Finally, we evaluated the biological functions of top-ranked variants in order to determine potential correlation with the ASD phenotype.

3. Results

3.1. *Classifier Performance*

Results from the logistic regression classifier as well as top-ranked variants are summarized in Figure 3. The classifier was evaluated on a held-out test set and was able to differentiate between ASD and PSP with high accuracy, with AUC-ROC values ranging from 0.600 to 0.960. The logistic regression classifier trained on variants located in simple repeat sequences showed the best performance out of all seven variant sets.

	miRNA	HAR	Hypersensitive Sites	TFBS	DNA Repeats	Simple Repeats	CpG Islands
	λ = 10: 110 variants AUC-ROC = 0.602 Precision = 0.889 Recall = 0.619 F_1 Score = 0.730	λ = 10: 108 variants AUC-ROC = 0.600 Precision = 0.893 Recall = 0.548 F_1 Score = 0.679	λ = 10: 614 variants AUC-ROC = 0.891 Precision = 0.933 Recall = 0.922 F_1 Score = 0.928	λ = 10: 637 variants AUC-ROC = 0.774 Precision = 0.888 Recall = 0.896 F_1 Score = 0.892	λ = 10: 649 variants AUC-ROC = 0.852 Precision = 0.898 Recall = 0.932 F_1 Score = 0.915	λ = 10: 519 variants AUC-ROC = 0.960 Precision = 0.949 Recall = 0.958 F_1 Score = 0.953	λ = 10: 522 variants AUC-ROC = 0.850 Precision = 0.924 Recall = 0.915 F_1 Score = 0.920
Top-Ranked Variants	Positive: 1-200938662 3-124950150 4-83551007 4-185678110 8-11702375 Negative: 1-56961756 2-32380330 9-14086349 12-6928569 X-153609616	Positive: 4-138785309 4-182253283 16-78992353 20-708998 20-61733540 Negative: 1-3089839 1-81623829 9-2621560 12-92757463 16-5508166	Positive: 1-17426602 2-215085206 11-63902879 15-42187492 16-1537926 Negative: 1-39900230 1-151762599 2-11797152 12-132339648 19-1361712	Positive: 2-119593844 5-160684599 8-114307607 11-124235672 19-30841145 Negative: 9-119245085 14-100995452 16-1894991 18-5600042 X-145430634	Positive: 3-63405151 5-20981037 6-13509234 12-92626545 20-18174324 Negative: 5-155993630 6-122479014 7-14626211 7-128128428 15-91429519	Positive 3-30550980 14-37565015 17-11206720 22-27486124 X-3127935 Negative: 7-137369693 8-26074016 10-49883667 X-55147362 X-143750718	Positive: 1-47082513 8-102506074 14-91731023 18-29304254 19-2137000 Negative: 6-131949293 13-37006117 15-82338172 18-33077673 19-46095110

Fig. 3. *Machine learning results.* We performed ℓ_1-regularized logistic regression for each noncoding region. AUC-ROC, precision, recall, and F_1 score show performance evaluated on the held-out test set. λ values for each noncoding region, as well as the number of remaining variants with nonzero coefficients remaining after feature selection, are listed. The 10 top-ranked variants for each classifier are listed in GRCh37 coordinates; the presence of variants with positive coefficient scores and the absence of variants with negative coefficient scores are likely to suggest the ASD phenotype.

3.2. *Bootstrap Test*

To determine whether the seven types of noncoding regions we tested are more predictive of ASD status than random sets of variants, we performed a bootstrap test. Figure 4 shows the

95% confidence interval for AUC-ROC performance of random variant sets of various sizes on the held-out test set. As the number of variants used for prediction increases, the AUC values achieved by the classifier also increase. This is expected because as we incorporate more variants into our classifier, we become increasingly likely to by chance include ASD-associated variants or variants in linkage-disequilibrium with autism-associated variants. Furthermore, as the number of variants included in the classifier increases, any subtle batch effects missed by our filtering procedure will begin to influence results.

We see that after accounting for variant set size, the microRNA, human accelerated region, and CpG island variant sets perform within the bootstrapped 95% confidence interval. Hypersensitive sites, transcription factor binding sites, and DNA repeat sequences all perform worse than random variant sets. These noncoding regions may not be associated with ASD, or our batch effect correction procedure may have been too stringent and removed important autism-associated signal. The classifier trained on simple repeat sequences is the only variant set that significantly outperforms the random bootstrap with a Bonferonni corrected p-value (accounting for the 7 tests performed) of 0.0287. This suggests that genetic variation within simple repeats may be associated with ASD risk.

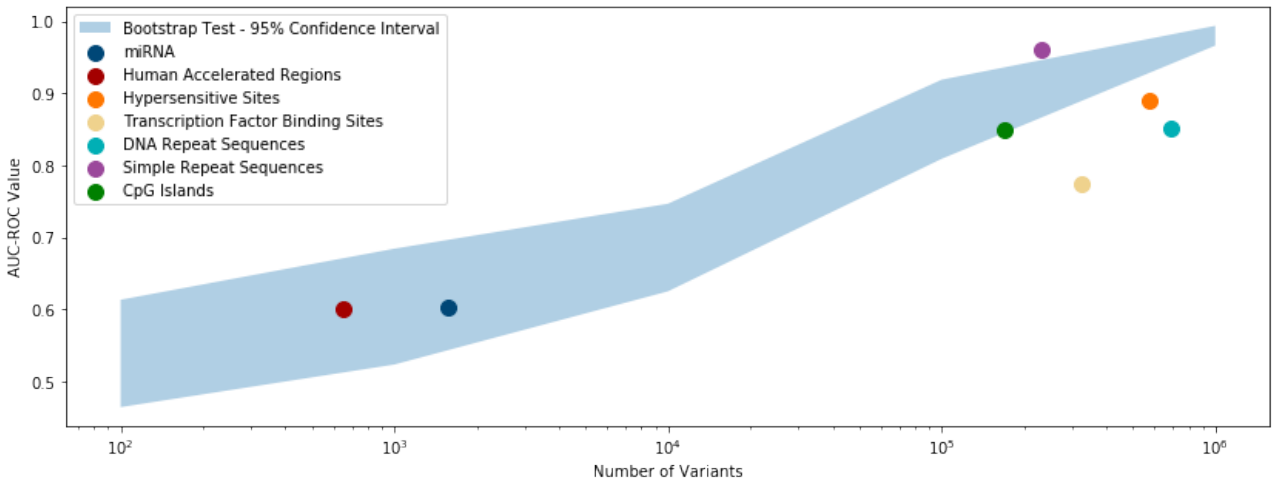


Fig. 4. *Evaluating prediction performance of noncoding regions.* The blue shaded region shows the 95% confidence interval for AUC-ROC performance of randomly selected sets of variants. As the number of variants provided to the model increases, performance increases as well. Six of the noncoding regions we studied performed at or below the bootstrapped models. However, the simple repeat sequences variants significantly outperformed the bootstrap, suggesting that these noncoding variants may be associated with ASD.

3.3. *Performance on an Independent Test Set*

In order to measure generalization ability, all seven classifiers were evaluated on an independent test set consisting of ASD patients from the Simons Simplex Collection and control individuals from the 1000 Genomes Project. AUC-ROC values ranged from 0.361 to 0.960, with most of the models suffering from a degradation in performance. However, the model trained on simple

repeat sequences maintained a large AUC-ROC, consistent with the hypothesis that this region contains relevant signal for differentiating ASD and neurotypical individuals. These results are in agreement with our bootstrap analysis.

	miRNA	HAR	Hypersensitive Sites	TFBS	DNA Repeats	Simple Repeats	CpG Islands
Independent Test Set (SSC ASD + 1000 Genomes)	0.361	0.375	0.593	0.351	0.682	0.960	0.589

Fig. 5. *Performance on an independent test set* This figure includes AUC-ROC values from validation on an independent cohort consisting of individuals from the Simon's Simplex Collection and the 1000 Genomes Project. Only the classifier trained on simple repeat sequences is able to generalize.

3.4. *Accounting for Population Substructure and Sex Differences*

To show that our classifier trained on simple repeat sequences is robust to population substructure, we analyzed the population composition of our case and control groups. Figure 6 shows our case and control populations superimposed on ethnicity profiles from the 1000 Genomes Project. Our PSP population is predominantly of European descent, while the iHART population is more diverse.

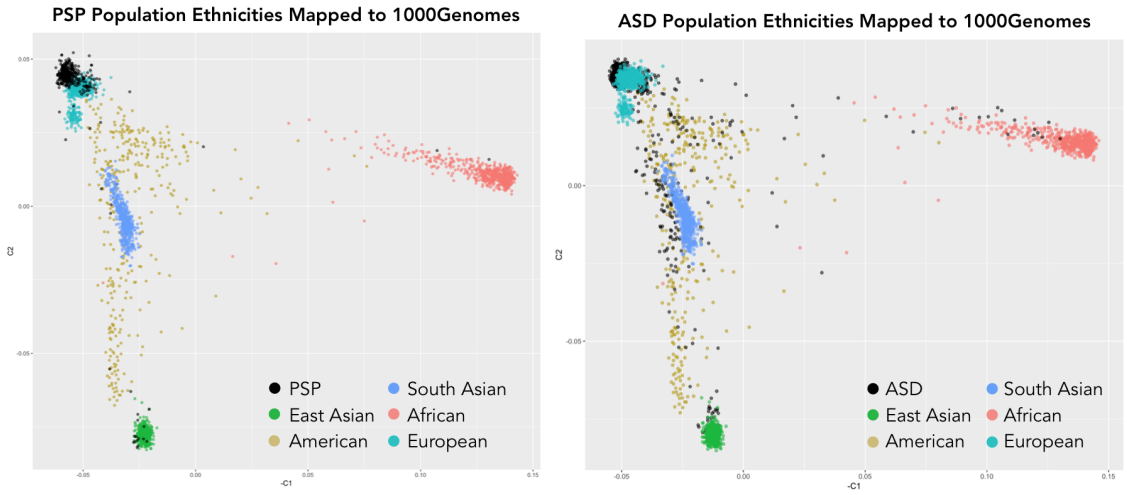


Fig. 6. *Population compositions of PSP and ASD cohorts* These plots map the PSP and ASD populations to a principal components plot of the 1000 Genomes population in order to identify the ethnicity of individuals in our datasets.

In order to ensure that this classifier is not biased by ethnicity, we evaluated its test performance on individuals of European and non-European descent separately. Figure 7 shows that it performs equally well on individuals of European or non-European ancestry, increasing our confidence that our results are not confounded by population substructure. We also evaluated differences in classification performance between males and females, also shown in Figure 7.

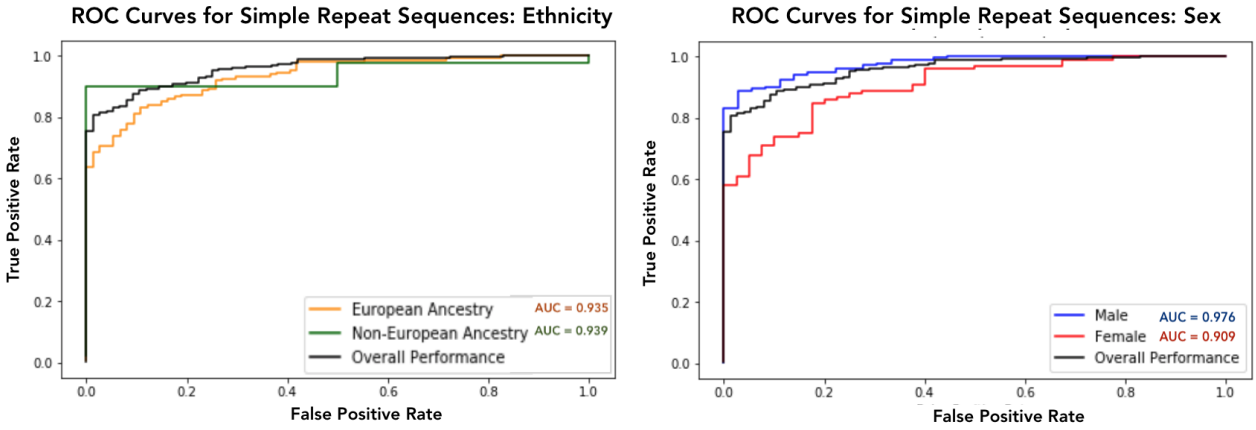


Fig. 7. *ROC curves for the classifier trained on simple repeat sequences across four splits of the held-out test set* The plots show that the classifier yields similar results on the European and non-European population. However, classifier performance is higher across males than females.

Our classifier is better able to predict ASD affected status in males than in females. This is interesting because ASD has a strong male bias with male children being four times more likely to develop autism than female children.[32]

3.5. *Biological Functions*

We evaluated the biological functions of all 70 top-ranked variants in order to identify potential correlations with the ASD phenotype. Since each variant either occurs in the intronic region of a gene or in an intergenic region between two genes, we generated a comprehensive list of genes associated with top-ranked variants. This resulted in a set of 98 genes, which we utilized to evaluate biological evidence. In the tissue-specific microRNA regions, a variant at position 200,938,662 in chromosome 1 is located in the intronic region of KIF21B, a gene that regulates synapse function and morphology of neurons; this gene is also known to play a role in learning and memory.[33] A variant at position 124,950,150 in chromosome 3 is located in ZNF148, which has been linked with developmental delays.[34] A top-ranked variant in chromosome 12 is located in the intronic region of CD4, a gene expressed in regions of the brain that is known to be a mediator of neuronal damage.[35] In noncoding regions containing DNA repeat sequences, gene GFOD1 contains a variant at location 13,509,234 on chromosome 6 and has been linked with Attention Deficit-Hyperactivity Disorder, a common comorbid condition of ASD.[36] Similarly, a top-ranked variant in a simple repeat sequence in chromosome 7 is located within the intronic region of gene DGKI; this gene has been linked with dyslexia, which is also a comorbid condition of ASD.[37] In addition, a variant at chromosome 17 in a simple repeat region is located within gene SHISA6, a regulator of synaptic transmission.[38]

In order to analyze the relationship between the 98 identified genes and a set of 109 genes known to confer elevated ASD risk, we constructed a protein-protein interaction network in STRING, as shown in Figure 8.[39] Edges are derived from text-mining, experiments, databases, co-expression, neighborhood, gene fusion, and co-occurrence. The network showed that twenty

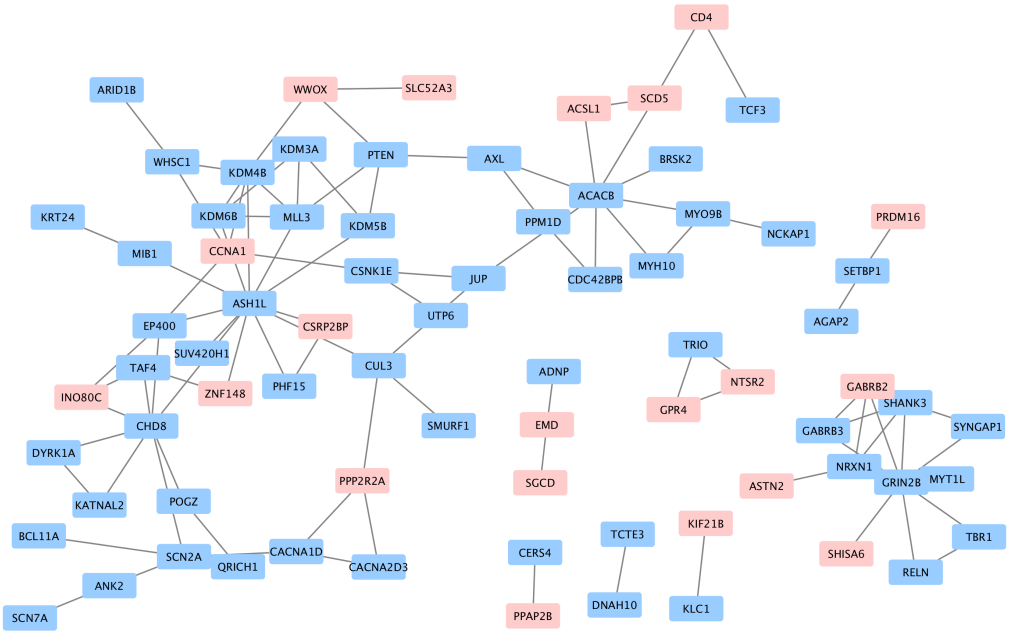


Fig. 8. *Gene interaction network* Interactions between genes previously linked with autism (in blue) and genes associated with the noncoding variants identified in this analysis (in pink) are shown in the figure. 20 identified genes interact closely with known ASD-risk genes. Notably, the gene CCNA1 is known to interact with 5 known ASD-linked genes.

newly-identified genes are closely connected to known ASD-linked genes.

4. Discussion

By utilizing outgroup machine learning to investigate the noncoding space, we were able to identify single nucleotide variants potentially associated with ASD. Biological validation of genes associated with top-ranked variants revealed a highly interconnected gene network, suggesting that identified genes interact closely with ASD-linked genes and may contribute to the ASD phenotype. Out of the seven regions analyzed in this work, the classifier trained on simple repeat regions demonstrated the strongest performance. Simple repeat sequences, also known as microsatellites, consist of repetitive sequences of one to ten base pairs; these regions are known to be extremely susceptible to mutations.[40] More than twenty neurodevelopmental and neurodegenerative conditions, many of which are comorbid with ASD, have been linked to unstable expansion of repeat sequences and consequent loss of protein function.[41] In addition, variation in promoter microsatellites of the gene AVPR1A has been implicated in increased susceptibility to ASD in an Irish population.[42] In this work, the classifier trained on simple repeat sequences significantly outperformed the random bootstrap test, indicating a potential correlation between variants in this region and the ASD phenotype; this was further supported by a biological analysis of top-ranked variants in simple repeat regions that revealed two genes associated with neural function.

Thus, our outgroup machine learning approach to elevate hidden signal in ASD genomes can effectively evaluate feature representations of the noncoding space; however, this method

has potential limitations, including batch effect correction and population stratification.

Current methods for addressing batch effects in whole genome sequencing data are meant to capture major differences in sequencing pipelines and are therefore quite stringent; the Type 2 Diabetes Consortium uses a series of quality control filters to identify batch effects resulting in a loss of 9.9% of called SNPs.[43] Our method for batch effect correction, adapted from the algorithm used by the UK10K Project,[29] is less conservative, discarding just under 5% of called SNPs. We believe this is appropriate since the batch effects in our dataset are much more subtle than those encountered by large consortia. Since our samples were sequenced at the same sequencing center with the same protocols and variant calling pipeline, we were able to control for many of the variables that could introduce batch effects. However, differences between populations in both cell type and the joint variant calling process could still create batch effect biases. The ASD samples were sequenced from lymphoblastoid cell lines while the PSP samples were sequenced from whole blood. Furthermore, while the same variant calling pipeline was used on both samples, GATK performs joint genotyping, a procedure that uses other samples in the cohort to resolve sequencing errors; since the two cohorts were run through the variant calling pipeline separately, subtle batch effects could have been introduced.

Regardless of batch effects, there remains the fundamental issue of population stratification in the merged dataset, especially since the initial cohorts were not drawn from the same ancestral or ethnic group. In order to establish a control for stratification, we created a null distribution by performing a bootstrap on successively larger variant sets, as reflected in Figure 4. High-performing null models likely do not reflect any neurological phenotype; rather, they represent the effect of divergent ancestry between the ASD and PSP cohorts. Interestingly, only the classifier trained on simple repeat sequences exceeded the null distribution for models of its size, suggesting a potential link with ASD.

Further analysis is needed to understand the biological consequences of these results. 40% of the top-ranked variants discovered in this analysis lie in intergenic regions; these may be enhancers to nearby genes, and we intend to explore associations between these variants and specific genes in a followup study. In addition, variants within simple repeat regions are challenging to call at low depth; in our current analysis, the top ten variants in simple repeat regions have an average read depth of 30.23 across the SSC dataset and an average read depth of 6.21 across the 1000 Genomes control dataset. In the future, we will validate our classifier using an independent test set sequenced at a higher depth of coverage.

5. Acknowledgments

This work was supported by the Hartwell Foundation award to D.P. Wall and the Hartwell Autism Research and Technology Initiative (iHART). This work was also supported by Bio-X and the Precision Health and Integrated Diagnostics (PHIND) Center at Stanford University.

References

1. E. Fombonne, *Journal of Child Psychology and Psychiatry* **59**, 717 (2018).
2. J. Hallmayer, S. Cleveland, A. Torres *et al.*, *Archives of General Psychiatry* **68**, 1095 (2011).
3. E. Colvert, B. Tick, F. McEwen, C. Stewart *et al.*, *JAMA Psychiatry* **72**, 415 (2015).

4. D. H. Geschwind *et al.*, *The Lancet Neurology* **14**, 1109 (2015).
5. G. Ramaswami and D. H. Geschwind, *Genetics of autism spectrum disorder*, 1 edn. (Elsevier B.V., 2018).
6. M. N. Weedon, I. Cebola, A.-M. Patch, S. E. Flanagan, E. De Franco, R. Caswell, S. A. Rodríguez-Seguí, C. Shaw-Smith, C. H. Cho, H. L. Allen *et al.*, *Nature genetics* **46**, p. 61 (2014).
7. E. S. Emison, A. S. McCallion, C. S. Kashuk, R. T. Bush, E. Grice, S. Lin, M. E. Portnoy, D. J. Cutler, E. D. Green and A. Chakravarti, *Nature* **434**, p. 857 (2005).
8. D. G. Lupiáñez, K. Kraft, V. Heinrich, P. Krawitz, F. Brancati, E. Klopocki, D. Horn, H. Kayserili, J. M. Opitz, R. Laxova *et al.*, *Cell* **161**, 1012 (2015).
9. R. Leslie, C. J. O'Donnell and A. D. Johnson, *Bioinformatics* **30**, 185 (2014).
10. T. Turner, P. Coe, D. Dickel, K. Hoekzema, B. Nelson, M. Zody, Z. Kronenberg, F. Hormozdiari, A. Raja, L. Pennacchio, R. Darnell and E. Eichler, *Cell* **171**, 710 (2017).
11. D. Werling, H. Brand, J. An *et al.*, *Nature Genetics* **50**, p. 727736 (2018).
12. J. Piven, *American Journal of Medical Genetics* **105**, 34 (2001).
13. K. De Groot and J. W. Van Strien, *Advances in Neurodevelopmental Disorders* **1**, 129 (2017).
14. H. R. Morris, *Neurodegeneration* , p. 72 (2017).
15. B. Borroni, C. Agosti, E. Manani, M. D. Luca and A. Padovani, *Current Medicinal Chemistry* **18** (2011).
16. S. Hicks and F. Middleton, *Front Psychiatry* **7**, p. 176 (2016).
17. R. Doan, B. Bae, B. Cubelos, M. Nieto and C. Walsh, *Cell* **167** (2016).
18. T. Turner, F. Hormozdiari, M. Duzyend, S. McClymont *et al.*, *American Journal of Human Genetics* **98**, 58 (2016).
19. X. Wu, J. Qin, Y. You *et al.*, *Scientific Reports* **7** (2017).
20. M. Forrest, M. Hill, D. Kavanagh *et al.*, *Schizophrenia Bulletin* (2017).
21. J. Fondon, E. Hammock, A. Hannan and D. King, *Neuroscience Trends* **8**, 328 (2008).
22. Y. Like, A. Hannan and J. Craig, *Frontiers in Neurology* **6**, p. 107 (2015).
23. W. Kent, C. Sugnet, T. Furey *et al.*, *Genome Research* **12**, 996 (2002).
24. N. Sheffield, R. Thurman, L. Song, A. Safi *et al.*, *Genome Research* **23**, 777 (2013).
25. J. T. Leek, R. B. Scharpf, H. C. Bravo, D. Simcha, B. Langmead, W. E. Johnson, D. Geman, K. Baggerly and R. A. Irizarry, *Nature Reviews Genetics* **11**, p. 733 (2010).
26. J. A. Tom, J. Reeder, W. F. Forrest *et al.*, *BMC bioinformatics* **18**, p. 351 (2017).
27. J. T. Leek, *Nucleic Acids Research* **42**, p. e161 (2014).
28. M. A. Taub, H. C. Bravo and R. A. Irizarry, *Genome medicine* **2**, p. 87 (2010).
29. U. Consortium *et al.*, *Nature* **526**, p. 82 (2015).
30. P. Chaste, L. Klei, S. J. Sanders *et al.*, *Biological Psychiatry* **77**, 775 (2015).
31. A. Auton, G. R. Abecasis, D. M. Altshuler *et al.*, *Nature* **526**, 68 (2015).
32. D. M. Werling and D. H. Geschwind, *Current opinion in neurology* **26**, p. 146 (2013).
33. M. Muhia, E. Thies, D. Labonte *et al.*, *Cell* **15**, 968 (2016).
34. S. Stevens, A. van Essen, C. van Ravenswaiij *et al.*, *Genome Medicine* **8**, p. 131 (2016).
35. S. C. Byram, M. J. Carson, C. A. DeBoy, C. J. Serpe, V. M. Sanders and K. J. Jones, *Journal of Neuroscience* **24**, 4333 (2004).
36. J. Lasky-Su, B. Neale, B. Franke *et al.*, *American Journal of Medicine* **147B**, 1345 (2008).
37. H. Matsson, K. Tammimies, M. Zucchelli *et al.*, *Behavioral Genetics* **41**, 134 (2011).
38. R. Klaassen, J. Stroeder, F. Coussen *et al.*, *Nature Communications* **7** (2016).
39. D. Szklarczyk, A. Franceschini, S. Wyder *et al.*, *Nucleic Acids Research* **43** (2015).
40. A. D. C. M. MLC Vieira, L Santini, *Genetics and Molecular Biology* **39**, 312 (2016).
41. J. Gatchel and H. Zoghbi, *Nature Reviews Genetics* **6**, 743 (2005).
42. K. Tansey, M. Hill, L. Cochrane, M. Gill, R. Anney and L. Gallagher, *Molecular Autism* **2** (2011).
43. C. Fuchsberger, J. Flannick, T. Teslovich *et al.*, *Nature* **536**, p. 41 (2016).

Detecting potential pleiotropy across cardiovascular and neurological diseases using univariate, bivariate, and multivariate methods on 43,870 individuals from the eMERGE network

Xinyuan Zhang*[1], Yogasudha Veturi*[2], Shefali Verma[2], William Bone[1], Anurag Verma[2], Anastasia Lucas[2], Scott Hebbring[3], Joshua C. Denny[4], Ian B. Stanaway[5], Gail P. Jarvik[5], David Crosslin[5], Eric B. Larson[6], Laura Rasmussen-Torvik[7], Sarah A. Pendergrass[8], Jordan W. Smoller[9], Hakon Hakonarson[10], Patrick Sleiman[10], Chunhua Weng[11], David Fasel[11], Wei-Qi Wei[12], Iftikhar Kullo[13], Daniel Schaid[14], Wendy K. Chung[15], Marylyn D. Ritchie†[2]

1. *Genomics and Computational Biology Graduate Group, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA 19104, USA*
2. *Department of Genetics and Institute for Biomedical Informatics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA 19104, USA*
3. *Center for Human Genetics, Marshfield Clinic, Marshfield, WI 54449, USA*
4. *Department of Medicine, Vanderbilt University, Nashville, TN 37235, USA*
5. *Departments of Medicine (Medical Genetics) and Genomic Sciences, University of Washington School of Medicine, Seattle, WA 98195, USA*
6. *Kaiser Permanente Washington Health Research Institute, Seattle, WA 98101, USA*
7. *Department of Preventive Medicine, Northwestern University Feinberg School of Medicine, Chicago, IL 60611, USA*
8. *Biomedical and Translational Informatics Institute, Geisinger Health System, Danville, PA 17822, USA*
9. *Psychiatric and Neurodevelopmental Genetics Unit, Massachusetts General Hospital, Boston, MA 02114, USA*
10. *Center for Applied Genomics, Children's Hospital of Philadelphia, PA 19104, USA*
11. *Department of Biomedical Informatics, Columbia University, New York, NY 10032, USA*
12. *Department of Biomedical Informatics in School of Medicine, Vanderbilt University, Nashville, TN 37230, USA*
13. *Division of Cardiovascular Diseases, Mayo Clinic, Rochester, MN 55905, USA*
14. *Division of Biomedical Statistics and Informatics, Department of Health Sciences Research, Mayo Clinic, Rochester, MN 55905, USA*
15. *Department of Pediatrics, Columbia University, New York, NY 10032, USA*

The link between cardiovascular diseases and neurological disorders has been widely observed in the aging population. Disease prevention and treatment rely on understanding the potential genetic nexus of multiple diseases in these categories. In this study, we were interested in detecting pleiotropy, or the phenomenon in which a genetic variant influences more than one phenotype. Marker-phenotype association approaches can be grouped into univariate, bivariate, and multivariate categories based on the number of phenotypes considered at one time. Here we applied one statistical method per category followed by an eQTL colocalization analysis to identify potential pleiotropic variants that contribute to the link between cardiovascular and neurological diseases. We performed our analyses on ~530,000 common SNPs coupled with 65 electronic health record (EHR)-based phenotypes in 43,870 unrelated European adults from the Electronic Medical Records and Genomics (eMERGE) network. There were 31 variants identified by all three methods that showed significant associations across late onset cardiac- and neurologic- diseases. We further investigated functional implications of gene expression on the detected "lead SNPs" via colocalization analysis, providing a deeper understanding of the discovered associations. In summary, we present the framework and landscape for detecting potential pleiotropy using univariate, bivariate, multivariate, and colocalization methods. Further exploration of these potentially pleiotropic genetic variants will work toward understanding disease causing mechanisms across cardiovascular and neurological

* Authors contributed equally to this work
† Corresponding author

diseases and may assist in considering disease prevention as well as drug repositioning in future research.

Keywords: Pleiotropy; Cardiovascular Diseases; Neurological Disorders; Univariate Analysis; Bivariate Analysis; Multivariate Analysis; Colocalization; eQTL.

1. Introduction

Cognitive decline has been observed in nearly 42% of elderly individuals at five years after cardiac surgery[1]. Of late, there has been increasing clinical evidence suggesting a link between cardiovascular and neurological diseases. To facilitate efficient disease prevention and treatment for cardiovascular and neurological diseases, it is imperative to understand the underlying, often unexplained, disease-causing mechanisms across multiple phenotypes. Pleiotropy is a phenomenon that can explain the influence of a specific allele on two or more unrelated phenotypes. While there has been evidence of polygenic pleiotropy (where multiple variants are causally associated with multiple traits) among cardiovascular[2] and neurological diseases[3], recent work has also demonstrated a genetic basis for the link *between* these disease groupings. In particular, there has been evidence of genetic overlap *between* cardiovascular disease and (a) multiple sclerosis[4] as well as (b) schizophrenia[5]. Large-scale genomics data coupled with electronic health record (EHR) data can enhance our ability to uncover novel cross phenotype associations and potentially pleiotropic variants (cross-phenotype association could also be an artifact of linkage disequilibrium (LD) or disease co-morbidities rather than true pleiotropy)[6]. In this study, we sought to identify common genetic variants that contribute to the link between diseases of the circulatory and nervous system using 43,870 unrelated European adults and 65 disease phenotypes from the Electronic Medical Records and Genomics (eMERGE) network.

Statistical approaches to detect pleiotropy across multiple phenotypes can be univariate (CPMA[7], ASSET[8], MultiMeta[9], GPA[10], MTAG[11], etc.), bivariate, and multivariate (MTMM[12], MultiPhen[13], GEMMA[14], mvLMM[15], mvBIMBAM[16], etc.) in addition to network-based approaches, among others[17]. Univariate methods (e.g. Phenome wide association studies or PheWAS) are a powerful way to characterize the effect of a genetic variant on each phenotype independently, and potential pleiotropy can be detected when the same SNP is found to be significantly associated with multiple phenotypes. This method has shown great success in identifying potential pleiotropy in several clinical genomics studies[18-23]. However, a limitation of univariate analysis is that it tests only one trait at a time, so it cannot be a formal test of pleiotropy. In contrast, bivariate analysis has been shown to have higher power over univariate analysis by analyzing pairs of phenotypes simultaneously[24]. Furthermore, because bivariate analysis can be structured to test the association of a trait with a variant, while adjusting for another trait's association with the variant, bivariate analyses can be constructed to formally test pleiotropy, and extended to multivariate traits to perform sequential tests for pleiotropic effects[25,26]. In this study, we used a bivariate analysis approach using summary-statistics from univariate analysis to test the hypothesis of "joint association" of a SNP with a trait pair while accounting for correlation in z-scores between the trait pair[24]. The alternative hypothesis here is that *at least* one of the two traits is significantly associated with a SNP marker. This implementation of bivariate analysis has suggested potential pleiotropy as well as hinted at underlying disease-causing mechanisms in many recent studies[27,28]. Finally, multivariate analysis is designed to test the joint association between genotype with multiple phenotypes in a single regression model. Multivariate analysis has been shown to have

increased power over univariate analysis in many scenarios, including when the genotype affects either a single phenotype or multiple correlated phenotypes[29,30]. We chose MultiPhen[13] to perform multivariate analysis because of its ability to handle binary phenotypes as well as its high power, as demonstrated via simulations[29]. In this paper, we refer to MultiPhen as multivariate analysis for the sake of convenience. Again, here the alternative hypothesis is that *at least one* of many traits is significantly associated with the SNP marker.

Since the "true" pleiotropic associations among cardiovascular diseases and neurological disorders are largely unknown, we applied three types of widely used methods to characterize the landscape of *potential* pleiotropy at genome-wide level[31,32]. To improve our confidence that the list of potential pleiotropic variants obtained across all three methods reflect a single causal variant instead of coincidental overlap, we performed statistical colocalization for these signals with gene expression datasets across all 48 available tissues from the Genotype-Tissue Expression (GTEx) consortium[33]. For instance, if a SNP colocalizes with an eQTL for traits A *and* B, it means that the same SNP associates with both: (a) gene expression and trait A, (b) gene expression and trait B. This can help us infer that the same SNP associates with both traits A and B and is likely pleiotropic. We found that many of the potentially pleiotropic signals associated with both disease groupings (diseases of the nervous and circulatory system) colocalized with eQTLs from the GTEx consortium (especially on chromosome 22) indicating that gene expression might be influencing risk of disease at those loci. This study is one of the first large-scale natural data applications and evaluation of univariate, bivariate, multivariate and colocalization methods in one comprehensive analysis. The overall study design is shown in Figure 1.

2. Methods

2.1. *eMERGE network*

In this study, we used data from the Electronic Medical Records and Genomics (eMERGE) network Phase III. The eMERGE network is a National Human Genome Research Institute (NHGRI) organized consortium to explore the utility of DNA biorepositories coupled with Electronic Health Record (EHR) systems for large-scale genomic research. The eMERGE network Phase III consists of 83,717 genotyped samples across multiple platforms that are imputed to Haplotype Reference Consortium 1.1 reference in genome build 37 covering ~39 million genetic variants. There are seven eMERGE adult sites included in our study: Marshfield Clinic Research Foundation, Vanderbilt University Medical Center, Kaiser Permanente Washington/University of Washington, Mayo Clinic, Northwestern University, Geisinger, and Harvard University.

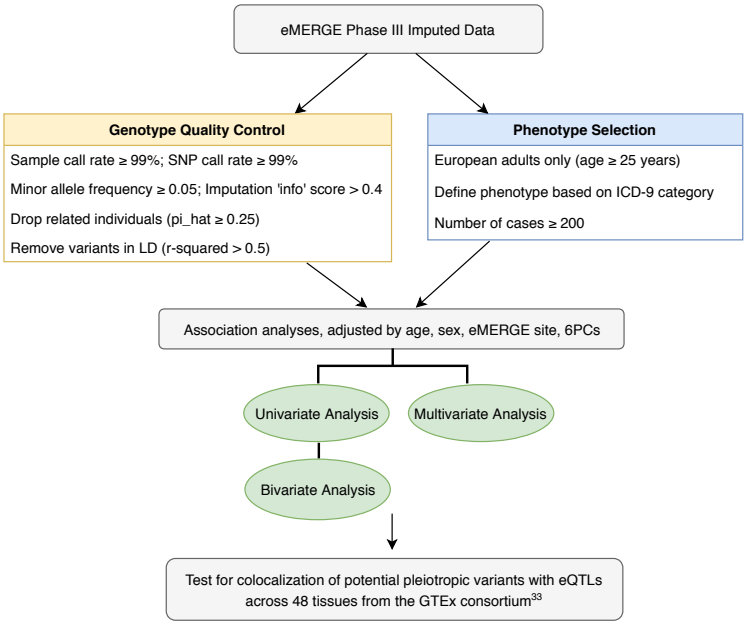


Figure 1. Overview of the analysis plan

2.2. *Genotypic Data and Quality Control*

eMERGE Phase III imputed genotypic data were cleaned following the "best-practice" quality control (QC) pipeline designed for imputed data[34]. We included genetic variants with genotype call rate ≥ 99% and sample call rate ≥ 99%. We selected common variants with minor allele frequency (MAF) ≥ 0.05. To account for sample relatedness, we dropped one of each related pair of individuals with pi_hat ≥ 0.25 (obtained from identity-by-descent estimation using PLINK[35]). We filtered out variants that had a linkage disequilibrium r^2 greater than 0.5 using a 100kb sliding window. We also filtered out the variants with a mean of imputation score less than or equal to 0.4. We further removed variants which have MAF difference greater than 0.1 compared to European population from 1000 Genomes Project[34]. After genotypic QC assessment and LD pruning, we had 54,942 unrelated individuals of European ancestry and 533,878 SNPs.

2.3. *Phenotype Definition and Selection Criteria*

2.3.1. *Phenotype Definition*

Cardiovascular and neurological phenotypes were defined using International Classification of Diseases, Ninth Revision (ICD-9) billing codes. We selected 98 ICD-9 codes from "Diseases of the circulatory systems" and "Diseases of nervous system and sense organs" as our primary phenotypes. Table 1 presents the major disease groups and corresponding ICD-9 codes. Of note, association analyses were performed using individual ICD-9 codes to define case/control status, and we used broader major disease categories for the purpose of presentation. The number of clinical visits per ICD-9 code per individual was used to define case-control status for each ICD-9 code: a case would be assigned if an individual had ≥ 3 instances; a control would be assigned if an individual had zero instances; an NA would be assigned if an individual had one or two instances[22].

2.3.2. *Phenotype Selection Criteria*

Our cohort comprised adults of European ancestry (age ≥ 25 years) from eMERGE network Phase III. We only used ICD-9 codes with more than or equal to 200 cases so as to increase statistical power of association tests[36]. As a result, a total of 65 cardiovascular and neurological ICD-9 based diagnoses and 43,870 individuals were included in our final round of association analyses. Individuals who have both cardiovascular and neurological disease were counted as cases for both. The sample size distribution of the 65 phenotypes is shown in Figure 2.

Table 1. Major group and ICD-9 category of neurological disorders and cardiovascular diseases

	Major Group	ICD-9 Codes
Circulatory System	Chronic rheumatic heart disease	393-398
	Hypertensive disease	401-405
	Ischemic heart disease	410-414
	Diseases of pulmonary circulation	415-417
	Other forms of heart disease	420-429
	Cerebrovascular disease	430-438
	Diseases of blood vessels	440-449
	Other diseases of circulatory system	451-459
Nervous System	Inflammatory diseases of the central nervous system	320-327
	Hereditary and degenerative diseases of the central nervous system	330-337
	Pain	338
	Disorders of the central nervous system	340-349
	Disorders of the peripheral nervous system	350-359

2.4. *Association Methods*

2.4.1. *Univariate Analysis*

We performed univariate logistic regression using 65 ICD-9 based diagnoses with 533,878 variants. We adjusted logistic regression models for sex, age, eMERGE site, and the first six principal components. We used PLINK 1.90 software[35] to perform the first round of univariate analysis because of its high computational efficiency. The logistic regression models converged for 33 out of 65 phenotypes. The major reason contributing to the non-convergence was the low sample sizes corresponding to some of the sites when we adjusted for eMERGE site (7 levels) as a categorical covariate. To address this, we used PLATO 2.1.0[37] to perform the second round of logistic regression tests on the remaining 32 phenotypes with the same set of covariates as before. Since PLATO implements an increased number of iterations compared to PLINK to find the best solution for logistic models, the software achieved convergence for all the remaining models. It should be noted that when both PLINK and PLATO converge, the results are concordant; these tools have been extensively compared previously[37].

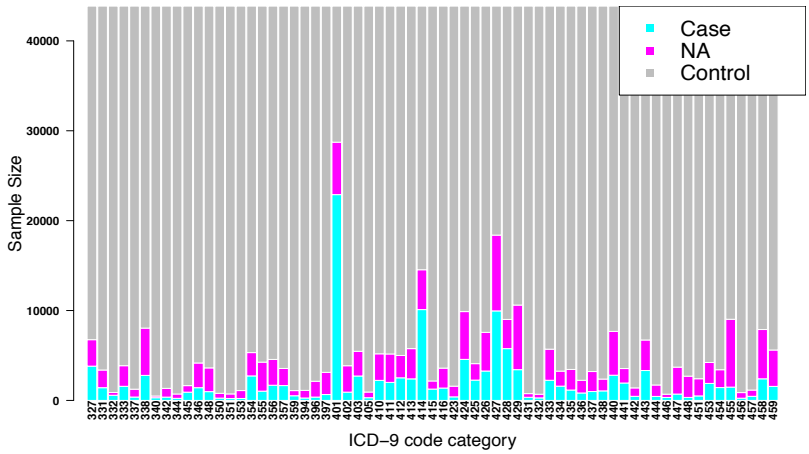


Figure 2. Sample size distribution for 65 ICD-9 disease categories

2.4.2. *Bivariate Analysis*

Bivariate analysis involved using summary-statistics (Z scores) from univariate analyses. We modeled our bivariate analysis protocol (with modifications) on the one followed by Siewert et al[27]. We first estimated mean and covariance of the Z scores obtained from univariate analyses for each of the 2080 pairs of phenotypes using all the available *LD-pruned* SNPs. This was done to ensure a null bivariate normal distribution of Z scores for each pair of phenotypes and to satisfy the "independence" assumption for hypothesis testing. Subsequently, we applied a p-value threshold of 0.005 on the univariate GWAS results and filtered out any SNPs that did not meet this threshold. We also filtered out SNPs with MAF = 0.5 to remove ambiguity pertaining to which allele was chosen as the referent allele in univariate analyses. Finally, we identified a list of common SNPs and estimated a p-value for each of 2,080 "pairs" of phenotypes using a chi-squared test with two degrees of freedom. Although we conducted a reduced number of tests, it should be noted that we corrected for multiple comparisons using the original "unfiltered" SNP set in order to control our type I error rate well.

2.4.3. *Multivariate Analysis*

We performed multivariate analysis using MultiPhen 2.0.2 R package[13]. MultiPhen analyzes multiple phenotypes jointly by testing linear combinations of phenotypes against each SNP using reverse ordinal regression. We adjusted for the same set of covariates as we did for univariate tests. By default, MultiPhen excludes individuals with at least one NA out of 65 phenotypes. Under this

scenario, the power of association tests would be limited as there would only be 7,535 individuals in total with extremely low case sample size per phenotype. Since we applied the "rule of three" to define a case, any person who had one or two instances of the occurrence of an ICD-9 code was set to missing (N/A). Because we did not want to drop so many individuals, we needed to fill in an alternative value for the N/A. For the purposes of multivariate analyses, these missing values were replaced by 0.5 to retain comparable sample size with univariate and bivariate analysis (sensitivity analyses on top significant SNPs yielded comparable results -- see Discussion). These individuals are *likely* cases since they have the ICD code in their record one or two times. A detailed evaluation of this replacement strategy will be conducted in the future to determine if a more optimal imputation strategy exists. Finally, to increase computational efficiency of MultiPhen, we parallelized the runs by splitting the genome into chunks of 10Mb each.

2.5. *Statistical Correction*

We implemented two Bonferroni correction calculation strategies to adjust for multiple testing when comparing the statistical performance of three types of methods. The Bonferroni threshold was calculated by dividing the level of significance by the number of tests. In the first strategy ("method-specific Bonferroni") we calculate Bonferroni threshold separately for each method. The derived significant thresholds for univariate, bivariate, multivariate testing were 1.44×10^{-9} [0.05/65*533878], 4.50×10^{-11} [0.05/(2080*533878)], and 9.37×10^{-8}[0.05/533878], respectively. We used an overly conservative significance threshold for bivariate analyses due to potential non-independence of tests (even after LD pruning). In the second strategy ("family-wise Bonferroni") we calculate Bonferroni threshold based on the total number of tests across all three methods. The derived significant threshold was 4.36×10^{-11} [0.05/(65*533878+2080*533878+533878)], and the criteria was applied across all three methods. Again, this correction is overly conservative given the correlation across the tests and methods but offers good control of the type I error rate.

2.6. *Colocalization*

Finally, we performed colocalization analysis to have greater confidence in our assessment of pleiotropy. We first obtained a list of potentially pleiotropic variants that cleared the "family-wise Bonferroni" multiple comparison threshold for univariate, bivariate and multivariate methods and narrowed down this list to SNPs that were associated with at least one disease from both nervous and circulatory systems. Finally, we ensured that for any given SNP, if one of the two traits in this circulatory-nervous trait pair had a univariate p-value that did not meet the "family-wise Bonferroni" threshold, it had a univariate -log10 p-value of at least 3. We termed the final list of SNPs as our "lead" SNPs. To test if these signals were being influenced by gene expression as well as driven by the same underlying variant, we performed statistical colocalization analyses using the "coloc" R package[38] between these signals and eQTLs (across all 48 available tissues) from the GTEx consortium[33]. We first obtained a 200KB window on either side of a "lead" SNP and looked for whether the lead SNP (or one in close LD with it) was an eQTL in a given tissue. If it was not an eQTL, that lead SNP was ignored. If it was an eQTL for a given tissue, we identified the corresponding "eGene" and obtained summary statistics from GTEx for all gene-variant associations in that 200KB window (either side). Note that we only chose the eGene that had the smallest p-value for a given eQTL from GTEx. Finally, for each phenotype with which the lead SNP is significantly associated, we performed statistical colocalization between the SNP and the

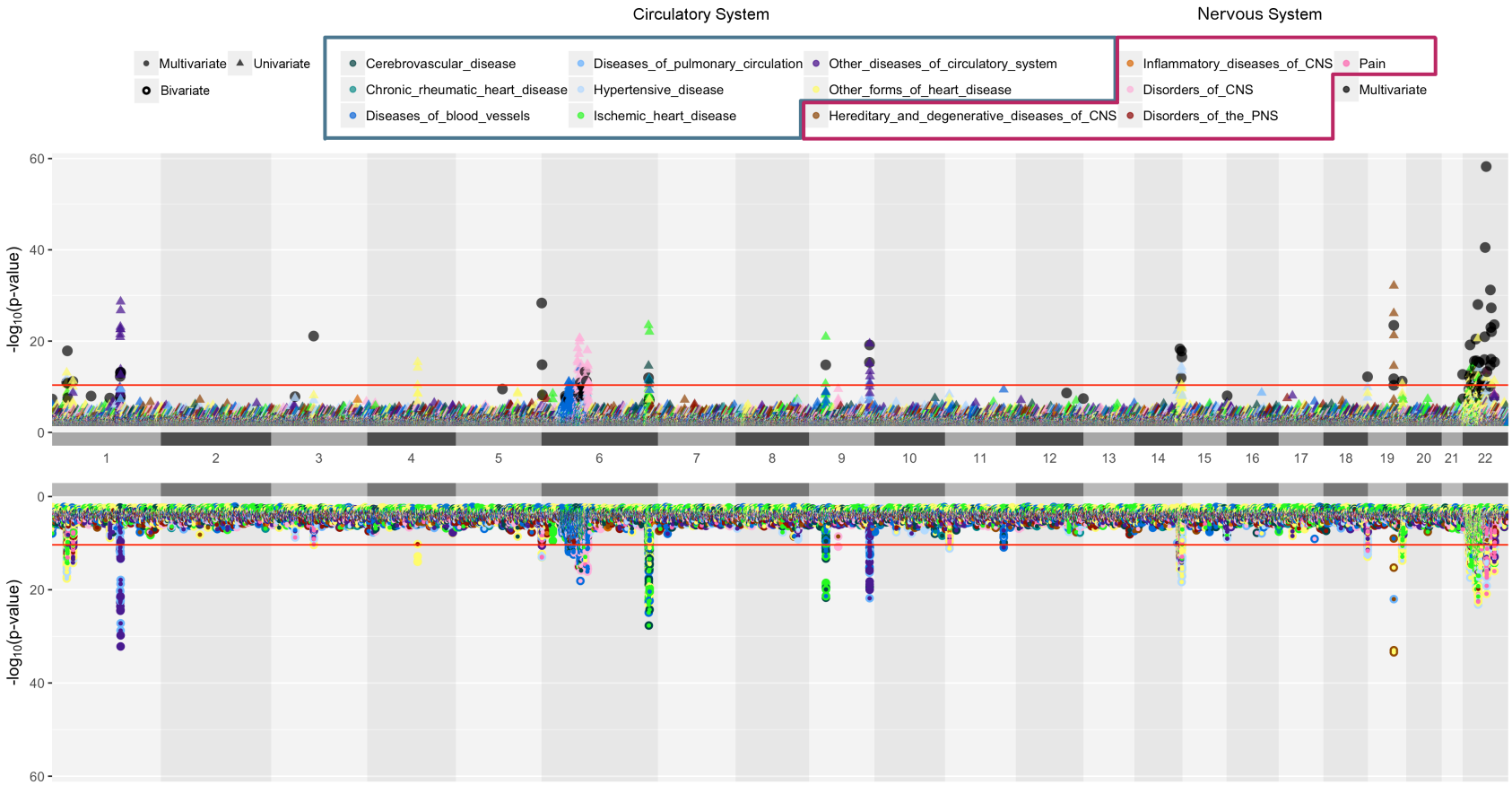


Figure 3. Univariate, Bivariate and Multivariate Results

A position-by-position comparison of genetic associations for univariate, bivariate and multivariate methods using code modified from Hudson R package (https://github.com/anastasia-lucas/hudson). The horizontal axis represents genomic locations by chromosome and the vertical axis represents $-\log_{10}$(p-value). Colors represent major disease groups of circulatory and nervous systems. The top plot presents univariate results with p-value less than 0.01 in triangles and multivariate results that passed "method-specific Bonferroni" threshold in black dots. The bottom plot present bivariate analysis results in a two-colored circle, denoting the two phenotypes with which a variant is associated with. The red lines in both plots are the "family-wise Bonferroni" threshold.

corresponding eQTL in that tissue. We set a coloc threshold of PP4/(PP3+PP4) > 0.8 to identify pleiotropic signals that are strongly influenced by gene expression. Here PP4 refers to the posterior probability that a single SNP associates with the phenotype as well as the gene expression whereas PP3 refers to the posterior probability of having two independent SNPs associate with either.

3. Results

3.1. *Landscape of Univariate, Bivariate and Multivariate Associations*

The landscape of univariate, bivariate, and multivariate association results is shown in Figure 3. There is an overall similar trend of association signals for univariate and bivariate analysis. We found that bivariate analysis identified more significant associations than univariate analysis when the correlation between phenotypes was low (less than 0.4). From the bottom half of Figure 3, we can see if the association signal from bivariate analyses comes from pairs of circulatory, nervous or circulatory-nervous traits. Black dots in Figure 3 represent the variants that passed "method-specific Bonferroni" significance from multivariate analysis. There are scenarios in which there is no significant association from univariate/bivariate analyses but significant results from multivariate analyses. Using "method-specific Bonferroni" threshold, univariate, bivariate, and multivariate methods detected 124, 108, and, 107 unique statistically significant SNPs, respectively; and there are 49 overlapping SNPs across three methods (data not shown). The number of variants detected at the more stringent "family-wise" threshold is given in Figure 4.

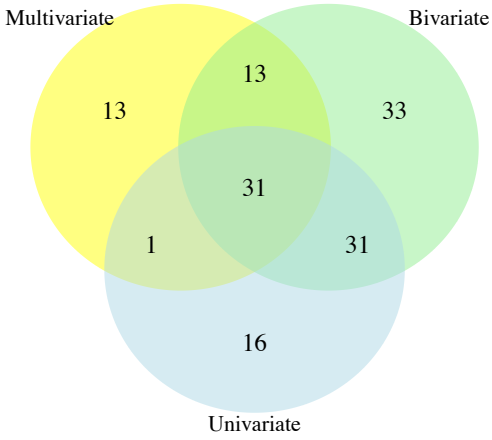


Figure 4. Venn diagram of the number of SNPs obtained at a "family-wise Bonferroni" threshold

3.2. *Variants associated with cardiovascular disease and neurological disorders*

Among the 31 "family-wise Bonferroni" SNPs across all three methods, we obtained 9 unique variants that are significantly associated with at least one cardiovascular disease and one neurological disorder from bivariate analysis that also "colocalized" with eQTLs across a host of tissues with a coloc PP4/(PP3+PP4) probability threshold of at least 0.8. Table 2 shows a comprehensive summary of these identified 9 variants. Our colocalization analyses revealed whether there was a shared variant underlying our potentially pleiotropic signals and whether gene expression may be influencing disease risk at these loci. For instance, the SNP at chromosome 1 and position 36822024 colocalized with eQTLs in the same 35 tissues for "Muscular dystrophies and other myopathies", "Pain" and "Other conditions of the brain" (neurological phenotypes) as well as "Heart failure", "Essential hypertension", "Cardiac dysrhythmias" and "Hypotension" (cardiovascular phenotypes) (eGenes: *EVA1B, TRAPPC3*). This means that rs10796883 influences 4 different cardiovascular disease categories, 3 different neurological disease categories as well as gene expression for *EVA1B* and *TRAPPC3* eGenes across 35 different tissues. Likewise, the variant on chromosome 22 position 22947156 colocalized with eQTLs in 4 tissues (Brain-cerebellum, testis, transformed fibroblasts, small intestine ileum) for 4 different neurological phenotypes as well as 9

other cardiovascular phenotypes (eGenes*: IGLV3-21, GGTLC2*). Please refer to Supplementary table 1 at https://ritchielab.org/files/PSB2019/Veturi/Supplementary_Data_1.txt for a complete list of tissues in which each of the lead SNPs colocalizes with eQTLs.

Table 2. Potential pleiotropic SNPs and their associated disease groups

SNP	Circulatory NeglogP(Uni-variate)	Nervous NeglogP(Uni-variate)	NeglogP (Bi-variate)	NeglogP (Multi-variate)	Tissue count	eGenes
1:36822024 rs10796883	Cardiac_dysrhythmias(11.305)	Muscular_dystrophies_and_other_myopathies(4.921)	13.247	11.165	35	EVA1B, TRAPPC3
		Other_conditions_of_brain(3.451)	12.030		35	EVA1B, TRAPPC3
		Pain(4.151)	12.363		35	EVA1B, TRAPPC3
	Essential_hypertension(9.125)	Muscular_dystrophies_and_other_myopathies(4.921)	11.325		35	EVA1B, TRAPPC3
	Heart_failure(10.029)	Muscular_dystrophies_and_other_myopathies(4.921)	11.988		35	EVA1B, TRAPPC3
		Pain(4.151)	11.452		35	EVA1B, TRAPPC3
	Hypotension(8.660)	Muscular_dystrophies_and_other_myopathies(4.921)	10.699		35	EVA1B, TRAPPC3
6:32569056 rs9270779	Atherosclerosis(14.165)	Multiple_sclerosis(6.355)	18.112	10.861	8	HLA-DRB5, HLA-DRB9
		Parkinson's_disease(3.196)	15.097		11	HLA-DRB5, HLA-DRB9
	Occlusion_and_stenosis_of_precerebral_arteries(6.355)	Multiple_sclerosis(5.913)	10.400		7	HLA-DRB5, HLA-DRB9
	Other_peripheral_vascular_disease(6.355)	Multiple_sclerosis(7.442)	11.787		4	HLA-DRB5, HLA-DRB9
14:106995720 rs7160440	Cardiac_dysrhythmias(11.322)	Muscular_dystrophies_and_other_myopathies(4.394)	12.989	18.291	5	IGHV3-53,IGHV4-39, IGHV3-49
		Other_conditions_of_brain(3.726)	12.420		5	IGHV3-53,IGHV4-39, IGHV3-49
		Pain(6.297)	14.259		5	IGHV3-53,IGHV4-39, IGHV3-49
	Essential_hypertension(7.451)	Pain(6.297)	10.610		1	IGHV3-49
	Heart_failure(9.038)	Muscular_dystrophies_and_other_myopathies(4.394)	10.752		8	IGHV3-53,IGHV4-39, IGHV3-49, HOMER2P1
		Other_conditions_of_brain(3.726)	10.469		6	IGHV3-53,IGHV4-39, IGHV3-49
		Pain(6.297)	12.465		5	IGHV3-53,IGHV4-39, IGHV3-49
	Hypertensive_chronic_kidney_disease(8.116)	Pain(6.297)	11.623		5	IGHV3-53,IGHV4-39, IGHV3-49
	Hypotension(10.278)	Muscular_dystrophies_and_other_myopathies(4.394)	11.832		5	IGHV3-53,IGHV4-39, IGHV3-49
		Other_conditions_of_brain(3.726)	11.252		5	IGHV3-53,IGHV4-39, IGHV3-49
		Pain(6.297)	13.004		5	IGHV3-53,IGHV4-39, IGHV3-49
	Ill-defined_descriptions_and_complications_of_heart_disease(7.610)	Pain(6.297)	11.224		1	
22:22876236 rs361535	Other_forms_of_chronic_ischemic_heart_disease(4.985)	Inflammatory_and_toxic_neuropathy(14.211)	14.702	10.424	1	
22:22947156 rs2097594	Cardiac_dysrhythmias(10.930)	Inflammatory_and_toxic_neuropathy(3.011)	11.236	28.019	1	
		Muscular_dystrophies_and_other_myopathies(3.773)	12.116		1	
		Other_conditions_of_brain(3.328)	11.738		1	
		Pain(5.622)	13.348		1	
	Cardiomyopathy(12.330)	Inflammatory_and_toxic_neuropathy(3.011)	12.818		2	GGTLC2
		Muscular_dystrophies_and_other_myopathies(3.773)	13.768		2	IGLV3-21, GGTLC2
		Other_conditions_of_brain(3.328)	13.507		1	GGTLC2
		Pain(5.622)	15.503		2	GGTLC2
	Essential_hypertension(10.187)	Muscular_dystrophies_and_other_myopathies(3.773)	11.380		2	BCRP4
		Other_conditions_of_brain(3.328)	10.968			
		Pain(5.622)	12.386			
	Heart_failure(20.621)	Inflammatory_and_toxic_neuropathy(3.011)	19.807		2	GGTLC2
		Muscular_dystrophies_and_other_myopathies(3.773)	20.963		3	IGLV3-21, GGTLC2
		Other_conditions_of_brain(3.328)	21.000		2	GGTLC2
		Pain(5.622)	22.553		2	GGTLC2
	Hypertensive_chronic_kidney_disease(9.331)	Muscular_dystrophies_and_other_myopathies(3.773)	10.760		2	GGTLC2
		Pain(5.622)	12.119		2	GGTLC2
	Hypotension(9.778)	Muscular_dystrophies_and_other_myopathies(3.773)	10.883		2	GGTLC2
		Other_conditions_of_brain(3.328)	10.491		2	GGTLC2
		Pain(5.622)	12.026		2	GGTLC2
	Ill-defined_descriptions_and_complications_of_heart_disease(10.665)	Inflammatory_and_toxic_neuropathy(3.011)	10.863		2	GGTLC2
		Muscular_dystrophies_and_other_myopathies(3.773)	11.703		2	GGTLC2
		Other_conditions_of_brain(3.328)	11.478		2	GGTLC2
		Pain(5.622)	13.385		2	GGTLC2
	Other_diseases_of_endocardium(10.340)	Inflammatory_and_toxic_neuropathy(10.340)	11.032			
		Muscular_dystrophies_and_other_myopathies(10.340)	11.844			
		Other_conditions_of_brain(10.340)	11.617			
		Pain(5.622)	13.627			
	Other_forms_of_chronic_ischemic_heart_disease(11.873)	Inflammatory_and_toxic_neuropathy(11.873)	11.335			
		Muscular_dystrophies_and_other_myopathies(11.873)	12.690			
		Other_conditions_of_brain(11.873)	12.530			
		Pain(5.622)	14.168			
22:25420792 rs13056641	Cardiac_dysrhythmias(9.528)	Inflammatory_and_toxic_neuropathy(4.159)	10.817	40.505	11	KIAA1671, SGSM1, CRYBB2, CRYBB3, IGLL3P
		Organic_sleep_disorders(4.166)	10.687		1	IGLL3P
		Pain(4.590)	11.247		6	KIAA1671, IGLL3P
	Essential_hypertension(12.162)	Inflammatory_and_toxic_neuropathy(4.159)	12.620		16	KIAA1671, SGSM1, CRYBB2, CRYBB3, IGLL3P, BCRP3
		Organic_sleep_disorders(4.166)	12.521		1	IGLL3P
		Pain(4.590)	13.284		7	KIAA1671, IGLL3P
22:25436904 rs1040421	Angina_pectoris(3.067)	Pain(13.338)	15.015	58.239	7	KIAA1671, SGSM1, IGLL3P
	Atherosclerosis(5.075)	Pain(13.338)	15.580		8	KIAA1671, SGSM1, IGLL3P
	Cardiac_dysrhythmias(11.931)	Pain(13.338)	20.872		7	KIAA1671, SGSM1, IGLL3P
	Cardiomyopathy(4.939)	Pain(13.338)	15.904		8	KIAA1671, SGSM1, IGLL3P
	Conduction_disorders(5.764)	Pain(13.338)	16.372		5	KIAA1671, SGSM1, IGLL3P

	Essential_hypertension(10.303)	Pain(13.338)	19.175		8	KIAA1671, SGSM1, IGLL3P
	Heart_failure(7.101)	Pain(13.338)	17.129		8	KIAA1671, SGSM1, IGLL3P
	Hypertensive_chronic_kidney_disease(7.426)	Pain(13.338)	17.404		8	KIAA1671, SGSM1, IGLL3P
	Hypotension(6.693)	Pain(13.338)	16.037		4	KIAA1671, SGSM1, IGLL3P
	Other_diseases_of_endocardium(5.845)	Pain(13.338)	16.677		4	KIAA1671, SGSM1, IGLL3P
22:28250172 rs1997739	Cardiac_dysrhythmias(10.517)	Pain(4.966)	12.443	22.064	19	ZNRF3, TTC28-AS1
22:33079917 rs5749490	Cardiac_dysrhythmias(11.280)	Hereditary_and_idiopathic_peripheral_neuropathy(3.049)	11.884	23.601	9	FBXO7, SLC5A4-AS1
		Inflammatory_and_toxic_neuropathy(3.958)	12.254		2	FBXO7, SLC5A4-AS1
		Mononeuritis_of_lower_limb_and_unspecified_site(3.153)	12.242		2	FBXO7, SLC5A4-AS1
		Pain(8.424)	16.011		9	FBXO7, SLC5A4-AS1
	Hypertensive_chronic_kidney_disease(6.449)	Pain(8.424)	12.064		9	FBXO7, SLC5A4-AS1
	Hypertensive_heart_disease(4.191)	Pain(8.424)	10.592		10	FBXO7, SLC5A4-AS1
	Hypotension(8.197)	Pain(8.424)	12.959		3	FBXO7, SLC5A4-AS1

Notes: We left as missing in the table any eGene (Ensembl gene ID from GTEx) that did not have an HGNC symbol counterpart.

4. Discussion

In this study, we conducted EHR-based univariate, bivariate, and multivariate analyses on 43,870 adults of European ancestry from the eMERGE network using 65 cardiovascular and neurological ICD-9 disease categories. The aim of this study was to detect pleiotropic genetic variants that influence diseases of the circulatory and nervous systems. We also evaluated the performance of three types of methods for detecting pleiotropy.

We observed 79, 108, and, 58 unique variants, respectively that were detected by univariate, bivariate, and multivariate methods and 31 that overlapped among the three methods using a "family-wise Bonferroni" significance threshold. Univariate analysis suggests direct association between genetic variant and phenotype; bivariate association can offer insights into whether a variant is associated with a pair of phenotypes, whereas multivariate analysis is powerful in detecting if a variant is associated with multiple phenotypes. We took the intersection of the significant genetic variants across the three methods as our list of potential pleiotropic variants. Our colocalization analyses revealed 9 SNP variants associated with at least one disease from both, nervous and circulatory system that cleared the "family-wise Bonferroni" threshold for multivariate and bivariate analyses. Since we were looking at trait pairs here, we ensured that at least one of the two traits had a univariate p-value that cleared the "family-wise Bonferroni" threshold while the other trait had a univariate -log10 p-value of at least 3. Note that we conducted sensitivity analyses for MultiPhen on identified potentially pleiotropic variants in Table 2 when missing values were imputed with 0 and 1 (i.e. treated as controls or cases) in addition to 0.5 and observed no change in significance. To cross-check overlap between methods, we also performed multivariate analysis restricted to a pair of bivariate significant traits for the 9 potentially pleiotropic variants in Table 2 and found 100% consensus between bivariate and multivariate methods. These 9 variants showed strong evidence of colocalization with eQTLs across a host of tissue types (see Supplementary table 1) from the GTEx consortium[33], especially on chromosome 22.

Our results replicated previous association signals as well as detected novel associations. SNP at chromosome 6 position 32569056 (rs9270779) has been directly implicated in autonomic nervous system and has been shown to be associated with heart rate response to exercise in females suggesting it could be pleiotropic for the two disease groupings of interest[39]. Also, the corresponding eGenes for this SNP, *HLA-DRB5* and *HLA-DRB9* from colocalization analysis have been previously shown to be associated with multiple sclerosis. Among the 31 total SNP hits, the one at chromosome 19 position 45416741 (rs438811) is correlated with rs445925 (r^2=0.341), which has been shown to be clinically relevant to cardiovascular phenotypes[40]. This SNP is also located in the *APOC1/APOE* region, which has been shown to be associated with Alzheimer's disease[41]. Among novel potential

pleiotropic variants identified by all three methods *and* colocalization analysis, 6 out of 9 variants locate on chromosome 22, suggesting its potential crucial contribution to the link between cardiovascular and neurological diseases. In particular, the eGene *FBXO7* has been associated with multiple sclerosis[42] as well as heart disease[43]. As part of future work, we will conduct pathway analyses or conditional analyses to have confidence in a singular pleiotropic association or shared biology between these disease groupings.

The limitations of this study are that (1) using only ICD-9 codes instead of both ICD-9 and ICD-10 codes may have reduced the number of cases in our data; (2) the use of disease category instead of disease code as phenotype might have reduced the specificity of detected associations. We are planning to incorporate ICD-9 and ICD-10 codes to define primary phenotypes and examine disease heterogeneity in the future; (3) sample size considerations led to some diagnosis codes being left out of analyses; (4) given our very conservative multiple comparison thresholds, we have likely reported only a fraction of all potential pleiotropic signals, leading to type II errors, and (5) we were unable to investigate how many additional associated variants obtained using bivariate analyses in comparison to univariate and multivariate were "true positives". One way to investigate this would be to test for statistical colocalization on top bivariate analyses hits[27]. However, this necessitates that summary statistics be obtained from independent datasets which was not the case with our data. Replication of these signals in independent cohorts in future can help us address this limitation.

In summary, we provide a framework for future pleiotropy analyses in EHR data. Our work expands the pleiotropy detection framework from univariate methods (e.g. PheWAS) to bivariate and multivariate methods in large-scale real-world EHR data to detect a broader net of potentially pleiotropic signals across cardiovascular and neurological disorders. We also utilize colocalization analyses to enhance our understanding of the influence of gene expression on these potentially pleiotropic variants and consequently on disease risk. In future, we will also try to replicate the partially overlapping SNP signals in independent cohorts.

Acknowledgments

The eMERGE Network was initiated and funded by NHGRI through the following grants:

Phase III: U01HG8657 (Kaiser Permanente Washington (formerly known at GroupHealth) /University of Washington); U01HG8685 (Brigham and Women's Hospital); U01HG8672 (Vanderbilt University Medical Center); U01HG8666 (Cincinnati Children's Hospital Medical Center); U01HG6379 (Mayo Clinic); U01HG8679 (Geisinger Clinic); U01HG8680 (Columbia University Health Sciences); U01HG8684 (Children's Hospital of Philadelphia); U01HG8673 (Northwestern University); U01HG8701 (Vanderbilt University Medical Center serving as the Coordinating Center); U01HG8676 (Partners Healthcare/Broad Institute); and U01HG8664 (Baylor College of Medicine)
Phase II: U01HG006828 (Cincinnati Children's Hospital Medical Center/Boston Children's Hospital); U01HG006830 (Children's Hospital of Philadelphia); U01HG006389 (Essentia Institute of Rural Health, Marshfield Clinic Research Foundation and Pennsylvania State University); U01HG006382 (Geisinger Clinic); U01HG006375 (Group Health Cooperative/University of Washington); U01HG006379 (Mayo Clinic); U01HG006380 (Icahn School of Medicine at Mount Sinai); U01HG006388 (Northwestern University); U01HG006378 (Vanderbilt University Medical Center); and U01HG006385 (Vanderbilt University Medical Center serving as the Coordinating Center).
If the project includes data from the eMERGE imputed merged Phase I and Phase II dataset, please also add U01HG004438 (CIDR) and U01HG004424 (the Broad Institute) serving as Genotyping Centers. And/or The PGRNSeq dataset (eMERGE PGx), please also add U01HG004438 (CIDR) serving as a Sequencing Center.
Phase I: U01-HG-004610 (Kaiser Permanente Washington /University of Washington); U01-HG-004608 (Marshfield Clinic Research Foundation and Vanderbilt University Medical Center); U01-HG-04599 (Mayo Clinic); U01HG004609 (Northwestern University); U01-HG-04603 (Vanderbilt University Medical Center, also serving as the Administrative Coordinating Center); U01HG004438 (CIDR) and U01HG004424 (the Broad Institute) serving as Genotyping Centers.

References

1. Bruggemans, E. F. Cognitive dysfunction after cardiac surgery: Pathophysiological mechanisms and preventive strategies. *Neth Heart J* **21,** 70–73 (2012).
2. Webb, T. R. *et al.* Systematic Evaluation of Pleiotropy Identifies 6 Further Loci Associated with Coronary Artery Disease. *Journal of the American College of Cardiology* **69,** 823–836 (2017).
3. Ibanez, L. *et al.* Pleiotropic Effects of Variants in Dementia Genes in Parkinson Disease. *Front. Neurosci.* **12,** 633–10 (2018).
4. Wang, Y. *et al.* Genetic overlap between multiple sclerosis and several cardiovascular disease risk factors. *Mult Scler* **22,** 1783–1793 (2016).
5. Andreassen, O. A. *et al.* Genetic pleiotropy between multiple sclerosis and schizophrenia but not bipolar disorder: differential involvement of immune-related gene loci. *Mol Psychiatry* **20,** 207–214 (2014).
6. Ritchie, M. D. Large-Scale Analysis of Genetic and Clinical Patient Data. *Annu. Rev. Biomed. Data Sci.* **1,** 263-274 (2018).
7. Cotsapas, C. *et al.* Pervasive Sharing of Genetic Effects in Autoimmune Disease. *PLoS Genet* **7,** e1002254 (2011).
8. Bhattacharjee, S. *et al.* A Subset-Based Approach Improves Power and Interpretation for the Combined Analysis of Genetic Association Studies of Heterogeneous Traits. *The American Journal of Human Genetics* **90,** 821–835 (2012).
9. Vuckovic, D. *et al.* MultiMeta: an R package for meta-analyzing multi-phenotype genome-wide association studies. *Bioinformatics* **31,** 2754-2756 (2015).
10. Chung, D. *et al.* GPA: A Statistical Approach to Prioritizing GWAS Results by Integrating Pleiotropy and Annotation. *PLoS Genet* **10,** e1004787 (2014).
11. Turley, P. *et al.* Multi-trait analysis of genome-wide association summary statistics using MTAG. *Nat Genet* **50,** 229–237 (2018).
12. Korte, A. *et al.* A mixed-model approach for genome-wide association studies of correlated traits in structured populations. *Nat Genet* **44,** 1066–1071 (2012).
13. O'Reilly, P. F. *et al.* MultiPhen: Joint Model of Multiple Phenotypes Can Increase Discovery in GWAS. *PLoS ONE* **7,** e34861–12 (2012).
14. Zhou, X. *et al.* Efficient multivariate linear mixed model algorithms for genome-wide association studies. *Nature Methods* **11,** 407–409 (2014).
15. Furlotte, N. A. *et al.* Efficient Multiple Trait Association and Estimation of Genetic Correlation Using the Matrix-Variate Linear Mixed-Model. *Genetics* **200,** 114.171447–68 (2015).
16. Stephens, M. A Unified Framework for Association Analysis with Multiple Related Phenotypes. *PLoS ONE* **8,** e65245 (2013).
17. Hackinger, S. *et al.* Statistical methods to detect pleiotropy in human complex traits. *Open Biol.* **7,** 170125–13 (2017).
18. Verma, A. *et al.* PheWAS and Beyond: The Landscape of Associations with Medical Diagnoses and Clinical Measures across 38,662 Individuals from Geisinger. *The American Journal of Human Genetics* **102,** 592–608 (2018).
19. Pendergrass, S. A. *et al.* Phenome-Wide Association Study (PheWAS) for Detection of Pleiotropy within the Population Architecture using Genomics and Epidemiology (PAGE) Network. *PLoS Genet* **9,** e1003087–26 (2013).
20. Bastarache, L. *et al.* Systematic comparison of phenome-wide association study of electronic medical record data and genome-wide association study data. *Nature Biotechnology* **31,** 1102–1110 (2013).
21. Hall, M. A. *et al.* Detection of Pleiotropy through a Phenome-Wide Association Study (PheWAS) of Epidemiologic Data as Part of the Environmental Architecture for Genes Linked to Environment (EAGLE) Study. *PLoS Genet* **10,** e1004678–33 (2014).
22. Verma, A. *et al.* eMERGE Phenome-Wide Association Study (PheWAS) identifies clinical associations and pleiotropy for stop-gain variants. *BMC Medical Genomics* **9,** 1–7 (2016).
23. Denny, J. C. *et al.* Variants Near FOXE1 Are Associated with Hypothyroidism and Other Thyroid Conditions: Using Electronic Medical Records for Genome- and Phenome-wide Studies. *The American Journal of Human Genetics* **89,** 529–542 (2011).
24. Liu, Y. *et al.* Powerful Bivariate Genome-Wide Association Analyses Suggest the SOX6 Gene Influencing Both Obesity and Osteoporosis Phenotypes in Males. *PLoS ONE* **4,** e6827–8 (2009).
25. Schaid, D. J. *et al.* Multivariate generalized linear model for genetic pleiotropy. *Biostatistics* **5,** e553–18 (2017).
26. Schaid, D. J. *et al.* Statistical Methods for Testing Genetic Pleiotropy. *Genetics* **204,** 116.189308–497 (2016).
27. Siewert, K. M. *et al.* Bivariate GWAS scan identifies six novel loci associated with lipid levels and coronary artery disease. *bioRxiv* 1–27 (2018).
28. Medina-Gomez, C. *et al.* Bivariate genome-wide association meta-analysis of pediatric musculoskeletal traits reveals pleiotropic effects at the SREBF1/TOM1L2 locus. *Nature Communications* **8,** 1–10 (2017).
29. Porter, H. F. *et al.* Multivariate simulation framework reveals performance of multi-trait GWAS methods. *Nature Publishing Group* **7,** 1–12 (2017).
30. Galesloot, T. E. *et al.* A Comparison of Multivariate Genome-Wide Association Methods. *PLoS ONE* **9,** e95923–8 (2014).
31. Solovieff, N. *et al.* Pleiotropy in complex traits: challenges and strategies. *Nature Reviews Genetics* **14,** 483–495 (2013).
32. Zhu, Z. *et al.* Statistical power and utility of meta-analysis methods for cross-phenotype genome-wide association studies. *PLoS ONE* **13,** e0193256 (2018).
33. Carithers, L. J. *et al.* A Novel Approach to High-Quality Postmortem Tissue Procurement: The GTEx Project. *Biopreservation and Biobanking* **13,** 311–319 (2015).
34. Verma, S. *et al.* Imputation and quality control steps for combining multiple genome-wide datasets. *Frontiers in genetics* **5, 370** (2014).
35. Purcell, S. *et al.* PLINK: A Tool Set for Whole-Genome Association and Population-Based Linkage Analyses. *The American Journal of Human Genetics* **81,** 559–575 (2007).
36. Verma, A. *et al.* A simulation study investigating power estimates in phenome-wide association studies. *BMC Bioinformatics* **19,** 1–8 (2018).
37. Hall, M. A. *et al.* PLATO software provides analytic framework for investigating complexity beyond genome-wide association studies. *Nature Communications* 1–10 (2017).
38. Giambartolomei, C. *et al.* Bayesian Test for Colocalisation between Pairs of Genetic Association Studies Using Summary Statistics. *PLoS Genet* **10,** e1004383–15 (2014).
39. Ramirez, J. *et al*. Thirty loci identified for heart rate response to exercise and recovery implicate autonomic nervous system *Nature Communications* **9,** 2041-1723 (2018).
40. Allen, N. B. *et al.* Genetic loci associated with ideal cardiovascular health: A meta-analysis of genome-wide association studies. *American Heart Journal* **175,** 112–120 (2016).
41. Bertram, L. *et al*. Genome-wide association studies in Alzheimer's disease. *Human Molecular Genetics* **18**, R137–R145 (2009).
42. Burchell V. S. *et al*. The Parkinson's disease–linked proteins Fbxo7 and Parkin interact to mediate mitophagy. *Nature Neuroscience* **16**, 1257–1265 (2013).
43. Li, Y. *et al*. The Role of Proteasome in Heart Disease. *Biochim Biophys Acta*. **1809**, 141–149 (2011).

Integrating RNA expression and visual features for immune infiltrate prediction

Derek Reiman[1], Lingdao Sha[1], Irvin Ho[1], Timothy Tan[2], Denise Lau[1,†], and Aly A. Khan[1,3,†]

[1] *Tempus Labs, Chicago, IL 60654, USA,* [2] *Department of Pathology, Northwestern University, Chicago, IL 60637, USA,* [3] *Toyota Technological Institute at Chicago, Chicago, IL 60611, USA.*

[†] *Corresponding authors: denise@tempus.com; aakhan@ttic.edu*

Patient responses to cancer immunotherapy are shaped by their unique genomic landscape and tumor microenvironment. Clinical advances in immunotherapy are changing the treatment landscape by enhancing a patient's immune response to eliminate cancer cells. While this provides potentially beneficial treatment options for many patients, only a minority of these patients respond to immunotherapy. In this work, we examined RNA-seq data and digital pathology images from individual patient tumors to more accurately characterize the tumor-immune microenvironment. Several studies implicate an inflamed microenvironment and increased percentage of tumor infiltrating immune cells with better response to specific immunotherapies in certain cancer types. We developed NEXT (Neural-based models for integrating gene EXpression and visual Texture features) to more accurately model immune infiltration in solid tumors. To demonstrate the utility of the NEXT framework, we predicted immune infiltrates across four different cancer types and evaluated our predictions against expert pathology review. Our analyses demonstrate that integration of imaging features improves prediction of the immune infiltrate. Of note, this effect was preferentially observed for B cells and CD8 T cells. In sum, our work effectively integrates both RNA-seq and imaging data in a clinical setting and provides a more reliable and accurate prediction of the immune composition in individual patient tumors.

Keywords: Cancer immunology, digital pathology, immune infiltration, machine learning.

1. Introduction

Immune infiltration and its spatial organization within the tumor microenvironment has long been associated with cancer progression and clinical outcome.[1,2] The potential of the immune infiltrate as a prognostic biomarker has become increasingly relevant with the advent of cancer immunotherapies. Checkpoint blockade and other cancer immunotherapies can induce clinical responses in some cancer patients.[3,4] However, clinical responses are only observed in a proportion of patients and vary for different cancer types, suggesting that additional factors, such as the composition of the immune infiltrate, may be important determinants of clinical response.[5,6] Several clinical studies show the tumor immune microenvironment, particularly the presence or absence of key effector cells such as CD8 T cells, can affect tumor immune responses.[7,8] The challenge, then, is to develop accurate methods to characterize the immune infiltrate in cancer patients in a reproducible and cost effective manner in order to ultimately identify novel prognostic markers.

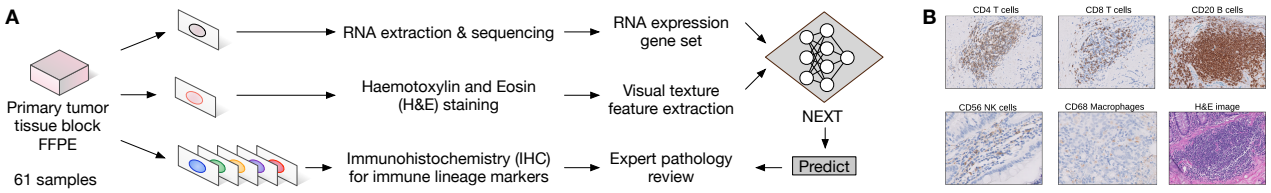


Fig. 1. Multi-modal approach used to train and validate NEXT for predicting the tumor immune infiltrate fraction and composition. Alternate slides cut from primary tumor FFPE blocks were used for RNA-seq or H&E staining. RNA expression data and visual features extracted from H&E slides were then fed into NEXT. IHC staining for lineage specific immune markers (CD20, CD4, CD8, CD68, CD56) was used by a pathologist to establish immune infiltration proportions.

Solid tumors are commonly infiltrated by adaptive and innate immune cells, including T and B lymphocytes, natural killer (NK) cells, and macrophages (MACs).[7,9] In the prevailing model, distinct effector cells in the tumor-immune microenvironment cooperate to present, recognize, and respond to tumor-specific antigens. However, several roadblocks exist for routine, accurate, and widespread pathological reporting of the immune infiltrate in tumor biopsies. Visual assessment after immunohistochemistry (IHC) staining for lineage specific markers remains the gold standard for evaluating immune cell infiltration in solid tumors. However, routine IHC assessment is not possible due to additional tissue sample requirements and the need for pathologist scoring. Alternatively, advances in genomic sequencing has facilitated implementation of RNA-sequencing (RNA-seq) in clinical medicine, but due to the inherent difficulty in deconvolving gene expression measurements into component immune cells, these approaches encounter significant ambiguity in reliably identifying correct immune proportions. Finally, emergent laboratory-based techniques, such as multiplex immunofluorescence, indexed flow cytometry, and single cell RNA-seq, require specialized labs and expertise, which limits widespread access.

We seek a middle ground by integrating coarse visual texture features from routine hematoxylin and eosin (H&E) staining of solid tumors used in cancer staging and diagnosis with bulk tumor RNA-seq to reduce ambiguity in predicting the immune infiltrate. In particular, this paper focuses on developing and applying a new framework, a neural network-based approach for integrating gene expression and visual texture features (NEXT) from solid tumor samples in a clinical laboratory setting (Fig. 1). We present implementations for predicting both the relative proportion of individual key effector immune cells and total fraction of the tumor immune infiltrate. Consequently, owing to the flexibility of our neural network-based approach, we are able to evaluate the integration of additional contextual features, such as estimates of the total fraction of immune infiltrate, to boost the prediction of immune cell-type proportions.

To test our model, we evaluated NEXT against current state-of-the-art methods for predicting the immune infiltrate as a proportion and benchmarked against expert pathologist review of IHC stained sections. Previous methods for predicting the immune infiltrate have either focused solely on RNA-based data or image-based data. Our approach is the first, to our knowledge, to utilize a multi-modal approach to refine RNA-based immune cell estimates by combining information from pathology imaging features.

2. Materials and Methods

2.1. *Data*

61 colorectal (n = 14), breast (n = 15), lung (n = 17) and pancreatic (n = 15) formalin-fixed paraffin-embedded (FFPE) solid tumor blocks were cut into alternating sections for RNA-seq, hematoxylin and eosin (H&E) staining, and immunohistochemistry (IHC) staining (Fig. 1A). Normalized read counts from RNA-seq for a specific panel of genes and visual texture features from H&E stained slides were generated and used as input for NEXT. Immune infiltrate predictions from NEXT were compared to pathologist expert review of IHC stained tumor sections using a panel of immune lineage markers (Fig. 1B).

2.1.1. *RNA extraction and sequencing*

Total nucleic acid was extracted from FFPE tumor tissue sections, macrodissected based on pathologist assessment of tumor cellularity, and proteinase K digested. Total nucleic acid was extracted with a Chemagic360 instrument using a source-specific magnetic bead protocol and stored at 4°C if less than 24 hours and -80°C if longer. RNA was purified from the total nucleic acid by DNase I digestion and magnetic bead purification. RNA was quantified by a Quant-iT picogreen dsDNA reagent Kit or Quant-iT Ribogreen RNA Kit (Life Technologies). Quality was confirmed using a LabChip GX Touch HT Genomic DNA Reagent Kit or LabChip RNA High HT Pico Sensitivity Reagent Kit (PerkinElmer).

The libraries were prepared using the KAPA RNA HyperPrep Kit. One hundred nanograms of RNA per tumor sample was fragmented with heat in the presence of magnesium to an average size of 200 base pairs. RNA underwent first strand cDNA synthesis using random primers, followed by combined second strand synthesis, A-tailing, adapter ligation, bead-based cleanup, and library amplification. After library preparation, samples were hybridized with the IDT xGEN Exome Research Panel. Target recovery was performed using Streptavidin-coated beads, followed by amplification using the KAPA HiFi Library Amplification Kit. The RNA libraries were sequenced to obtain an average of 90 million reads, minimum of 50 million reads, on an Illumina HiSeq 4000 System utilizing patterned flow cell technology.

After completion of sequencing, FASTQ files were uploaded to Amazon Web Services (AWS) which triggers the sequence analysis pipeline that uses the CRISP clinical RNA-seq pipeline[10] orchestrated by the JANE workflow tool (Tempus Labs, Inc.). CRISP performs pre-alignment QC, read grooming, alignment, post-align QC, and gene level quantification. The gene level counts from CRISP are then converted to TPMs (transcripts per million) to normalize for gene length and library size.

2.1.2. *Visual texture feature extraction*

H&E stained slide images were tiled and downsampled, generating overlapping square tiles with 210x210 microns in tile size and 30 microns in shifting strip size. Image tiles were downsampled by 4 on each edge, as 1 micron equals 4 pixels in size. Statistical features for each tile were generated and converted into 196 feature vectors, consisting of intensity and texture features. Image intensity features included the mean, standard deviation, and sum, where

applicable, for the gray level; red, green, blue layer; H&E stain layers; optical density (od) 3 channels; hue; and saturation. Texture features included zernike moments[11] (0-24 moments), threshold adjacency analysis[12] values (statistics 0-53), local binary patterns[13] (histogram bins 0-25), gray scale co-occurrence matrix[14] and difference of Gaussian[15] statistical measures. QuPath software[16] was utilized for histology slide visualization, tissue detection and tiling. Scikit-image, scikit-learn and mahotas python libraries[17] were used for image processing, feature generation and classification.

2.1.3. *Immunohistochemistry staining for lineage specific markers*

All FFPE slides were stained using the Leica Bond III automated IHC instrument and Leica reagents. The Leica antibody panel included: CD45 clone X16/99, CD4 clone 4B12, CD8 clone 4B11, CD20 clone L26, CD56 clone CD564, and CD68 clone 514H12. CD20 was used in a 1:200 dilution, but all other antibodies were purchased prediluted. Slides were deparaffinized using Dewax Solution. Heat induced epitope retrieval was used to reverse cross-linked proteins in the tissue by heating slides to 38 degrees Celsius and applying Epitope Retrieval Solution 1, a citrate-based solution with a pH of 6.0. The Bond Polymer Refine Detection kit was used for IHC staining and hematoxylin counterstaining. Slides were then dehydrated, mounted, and cover-slipped.

2.1.4. *Expert pathology review of histology slides*

The IHC and H&E stained slides were scored by a pathologist. The percent of each immune cell-type of interest (CD20+ B, CD4+ T, CD8+ T, CD68 MAC, CD56 NK cells) and total immune percentage (CD45) was determined by estimating the percent of cells that stained positive by IHC for the protein uniquely expressed by that cell-type. The pathologist was instructed to exclude staining of non-immune cells in their scoring. For instance, if 20% of all cells on a slide stained positively for CD20 B cells, but half of those positively staining cells were tumor cells, that sample would be scored as having 10% B cells. The percent tumor, stroma, and immune cells were estimated from evaluating the cell morphologies on their respective H&E slides. The relative abundance of the immune cell-types was determined by dividing the percent of the particular cell-type by the percent of total immune cells.

2.2. *NEXT architectures*

Neural networks can function as flexible and efficient learning models when integrating heterogeneous features, such as gene expression imaging features. The NEXT framework involves using a neural network-based architecture to integrate RNA-seq and imaging data. We used this framework in three separate architectures: NN-RNA, NN-RNA-image, and NN-Transfer (Fig. 2). Broadly, our framework was designed as a shallow neural network that consists of < 3 layers containing a set of neurons where each neuron is a weighted sum of the inputs in that layer. Non-linear activation functions are applied to the neurons to allow the model to find non-linear relationships between gene expression and imaging features. The output of a layer is then used as the input to the next layer. More specifically, given an input vector x, a

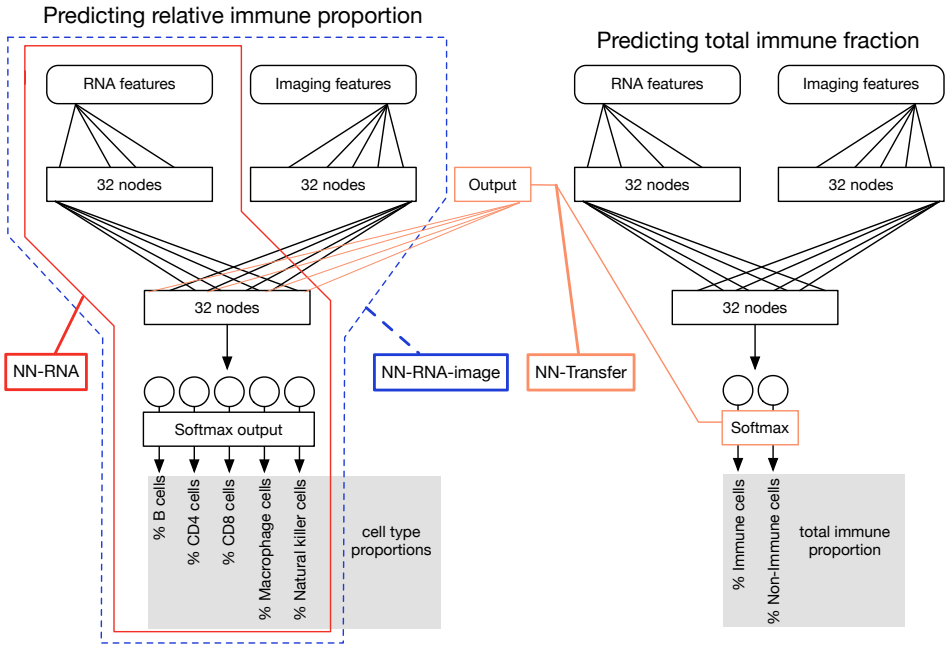


Fig. 2. NEXT architectures: NN-RNA (red), NN-RNA-image (blue), and NN-Transfer models (orange). The architecture takes in one or two inputs: RNA-seq gene expression alone (NN-RNA), or RNA-seq gene expression plus image features (NN-RNA-image). These inputs are passed into separate dense layers of 32 nodes in the first layer. The second layer contains a single dense layer of 32 nodes to integrate the information from the two sets of inputs. This layer is then fed into an output layer which uses the softmax activation to generate a probability distribution. These architectures can be used to predict relative immune cell-type abundances (left), or total fraction of tumor immune infiltrate (right). In the NN-Transfer model, we further boost the prediction of relative immune cell-type abundance through transfer learning by feeding the output of the total fraction of tumor immune infiltrate into the second layer of the model.

set of weights W, a bias term b, and an activation function ϕ, the output of the hidden layer, h, is calculated as:

$$h = \phi(Wx + b) \tag{1}$$

The neural network in this study was trained using both RNA-seq features and image features generated from image processing. The RNA-seq data was filtered using the LM22 gene list[18] and the TPM values were log transformed (feature size = 547). The image features included the mean and skewness values of the intensity and texture features across all tiles in an image (feature size = 392). In the first layer of the network, each set of features was used as inputs to their own fully connected layer which used the rectified linear unit (ReLU) activation function.

$$\text{ReLU}(x) = \max\{0, x\} \tag{2}$$

The second layer concatenated outputs of the modularized dense layers to create an integrated set of features. The values from this second layer were then passed to an output layer which used the softmax function to predict the desired immune proportion. The softmax function squashes an n-dimensional vector of real valued numbers into a new n-dimensional vector

with values in the range (0,1] and the sum of all the values is equal to one. More specifically, given a set of values for $Y = \{y_1, y_2, ...y_n\}$

$$\text{Softmax}(y_i|Y) = \frac{e^{y_i}}{\sum_j e^{y_j}} \tag{3}$$

Since our model was designed to predict a distribution, we trained it using the Kullback-Leibler divergence cost. The Kullback-Leibler divergence measures the divergence of a given probability distribution, Q, from a target probability distribution, P.

$$\text{KL}(P||Q) = \sum_i P(i)\, log\frac{P(i)}{Q(i)} \tag{4}$$

To prevent over-fitting of our model, we applied an L2 regularization to the weights for every layer. This regularizes the model by diffusing the weight vectors so that the model uses all of its weights rather than relying on a subset of higher valued weights. We also sought to enforce a shallow neural network architecture by reducing layer sizes until performance degradation was observed. Our final cost function for training was

$$C = \sum_i y_i\, log\frac{y_i}{\hat{y}_i} + \lambda \sum_L ||W^{(L)}||_2 \tag{5}$$

Here y_i is the true value for the probability of the ith output, $\hat{y}_i$ is the predicted probability for the ith output, λ is the L2 penalty coefficient, and $W^{(L)}$ are the weights for layer L.

The NN-RNA and NN-RNA-image architectures were trained to predict either the distribution of different immune cell-types in the sample or the total fraction of the tumor immune infiltrate. These models were trained using the ADAM optimizer for batch gradient descent with a learning rate of 0.005 and a λ value of 0.01. Models were trained and evaluated using leave-one-out cross validation. Specifically, for each left out example, we partitioned the other 60 samples into a training set of 40 and a validation set of 20. We then trained the model until the validation loss had not decreased within the last 5 epochs. We then predicted and reported the proportions of the single left out example.

After training the models, we evaluated if we could apply transfer learning by using one model to boost the other. For this, we used the outputs of the NN-Transfer model predicting the total fraction of the tumor immune infiltrate as additional inputs to the second layer of the NN-RNA-image model predicting the relative cell-type proportions. The NN-Transfer model was trained using the same methods and parameters described before.

3. Results

We tested the following four hypotheses. First, we tested whether NEXT could effectively learn and predict immune infiltration cell-type proportions from RNA-seq data (Section 3.1). Second, we tested whether integrating imaging features could further augment and improve infiltrate cell-type proportion prediction (Section 3.2). Third, we evaluated the flexibility of the NEXT architecture by predicting the total fraction of tumor immune infiltrate instead of the proportion of five key immune cell-types (Section 3.3). Finally, we tested the hypothesis that integrating estimates of the total fraction of immune infiltrate could yet further augment and improve prediction of the key immune cell-types (Section 3.4).

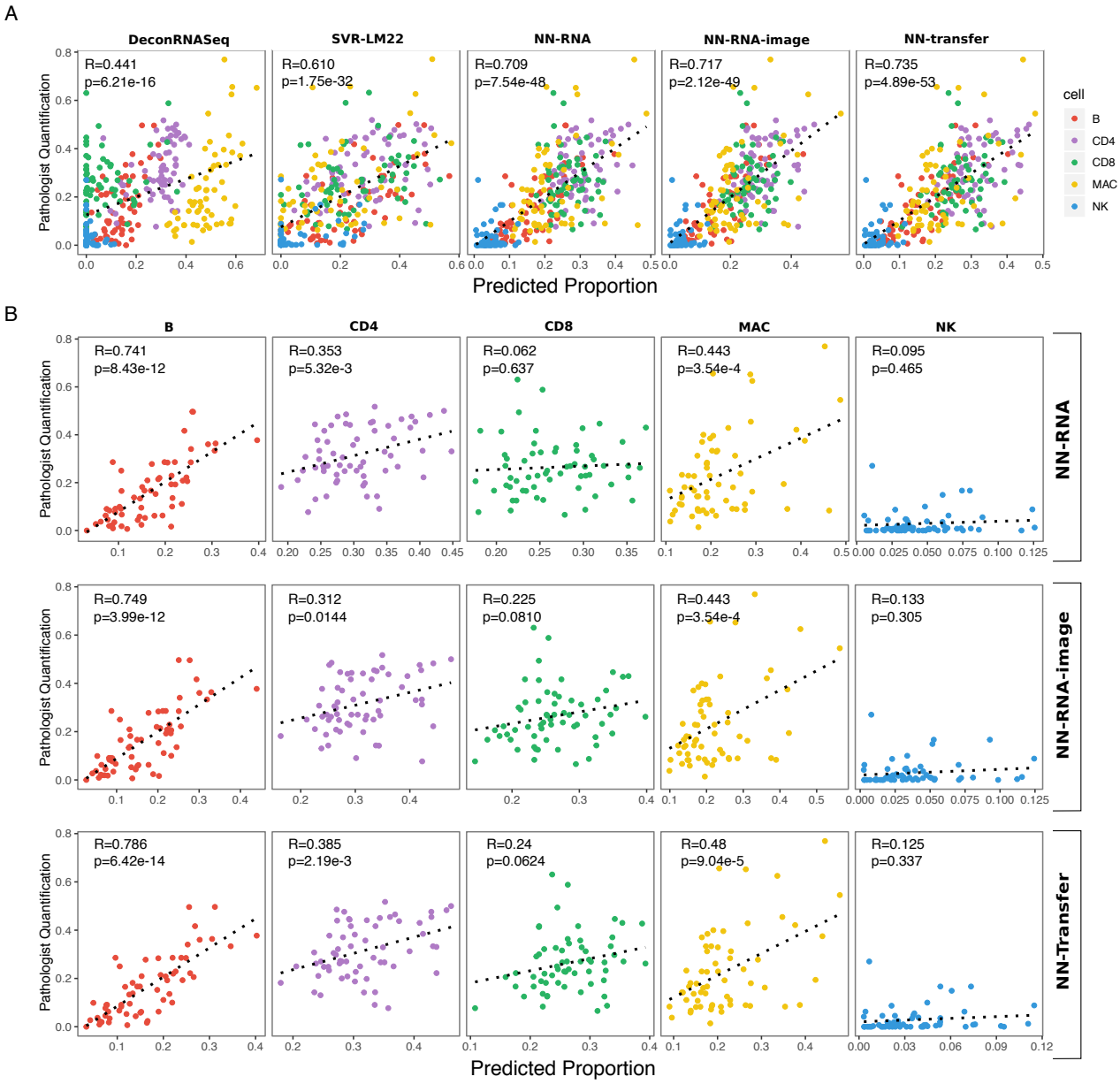


Fig. 3. Model performance benchmarking compared to expert pathologist assessment. (A) Predicted proportions of B, CD4 T, CD8 T, MAC, and NK cells of five models are shown in comparison to pathologist scoring of IHC lineage specific markers for 61 solid tumor samples. The sum of the proportions for all the cells for a particular sample will equal 1. The color of each point denotes the cell-type and the dotted line represents the linear regression line that best fits the data. The value of the Pearson correlation coefficient is shown in upper left corner of each plot. (B) The same data as (A) for the three neural network-based models (from top to bottom: NN-RNA, NN-RNA-image, NN-Transfer), but separated by immune cell-types.

3.1. *NEXT trained with RNA only (NN-RNA)*

Several groups have proposed methods for gene expression deconvolution using regression-based techniques. These include DeconRNASeq, which utilizes a non-negative linear regression

approach and CIBERSORT, which has demonstrated best-in-class performance for deconvolution using a support vector regression (SVR)-based approach.[18,19] We sought to determine if NEXT could perform comparably to these algorithms when trained on RNA-seq data only (NN-RNA). Due to commercial restrictions, we independently implemented a support vector regression deconvolution algorithm using the LM22 matrix.[18] Of the two regression-based techniques tested, we found that the SVR method performed better than DeconRNASeq, based on overall Pearson correlation.

To test the hypothesis that a neural network-based model (NN-RNA) could effectively learn immune cell proportions using RNA data only, we trained NEXT on the RNA-seq data using expert pathologist scoring of infiltration and evaluated performance using leave-one-out cross validation. The NN-RNA architecture was used to predict relative proportions for B, CD4 T, CD8 T, MACs, and NK cells. To establish a baseline against SVR, the RNA-seq data was filtered using the LM22 gene list and the TPM values were log transformed.

We found that NN-RNA performed better than SVR based on overall Pearson correlation (R=0.709; p=7.54e-48) (Fig. 3A). We attribute this improvement to two factors: (1) whereas SVR is a linear deconvolution method, NEXT can learn non-linear interactions between gene expression features; and (2) NEXT is trained and tested using RNA-seq data. While the authors of CIBERSORT indicate the SVR method with the LM22 matrix is amenable to RNA-seq data as well, we reason there is an advantage to using RNA-seq data for training when performing deconvolution on RNA-seq data. Overall, we find that NN-RNA effectively learns immune cell-type proportions and demonstrates better accuracies than current methods. Similar to SVR, NN-RNA performed best on B cells and worst on NK cells. This is likely due to B cells having a more distinct RNA profile, whereas NK cells likely share transcriptional similarities with CD8 T cells and comprise a smaller proportion of the immune infiltrate.[20]

3.2. *NEXT trained with RNA and image features (NN-RNA-image)*

Information about infiltrating immune cells in histopathology slides is normally only accessible by overlaying additional multiplexed immunofluorescence or immunohistochemistry stains.[21] We reason that embedded in microscopic H&E slides is latent information about the tumor-immune microenvironment, including the population structure and the underlying phenotypic states of the tumor and immune cells. Thus, we sought to test if integrating imaging features could further augment and improve the prediction of immune cell-type proportions.

To test this hypothesis, we obtained visual texture and intensity features from corresponding H&E images for each tumor sample. We utilized H&E image derived features due to the wide availability of H&E stained images used for cancer diagnosis and staging. To establish a baseline against NN-RNA, the RNA-seq data was filtered again using the LM22 gene list and the TPM values were log transformed. NN-RNA-image was trained to predict relative proportions for B, CD4 T, CD8 T, macrophage, and natural killer cells and was evaluated using leave-one-out cross validation.

NN-RNA-image boosted the prediction of immune cell-type proportions as evaluated by overall Pearson correlation (R=0.717; p=2.12e-49) (Fig. 3A). Of note, improvements were preferentially observed for CD8 T cells (R= .225; p=0.081) (Fig. 3B). These results suggest that

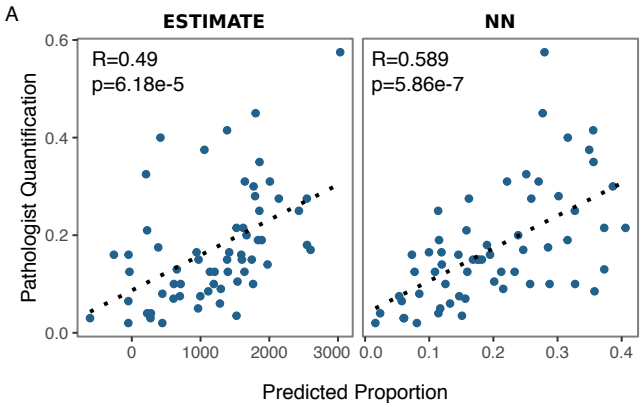


Fig. 4. Benchmarking of the total immune infiltrate fraction. Scatter plots illustrate pathologist counts compared to the immune score from ESTIMATE (left) and the predicted total fraction of immune infiltrate from NN-RNA-image (right).

integration of imaging features can function to improve immune infiltrate cell-type prediction.

3.3. *NEXT for predicting total tumor immune infiltration fraction*

Our choice of a generalizable neural network-based architecture for NN-RNA-image was deliberate as we hypothesized this could easily be adopted for other related but distinct tasks. We sought to evaluate the flexibility of NN-RNA-image in predicting the total fraction of immune infiltrate instead of the proportion of key immune cell-types. The total immune fraction framework seeks to predict the abundance of immune cells in the overall tumor microenvironment, in contrast to relative proportions of immune subsets in the total leukocyte population (Fig. 2). Here, a pathologist was instructed to assess immune cells (leukocytes) based on cell morphologies from patient H&E slides. We implemented a version of NN-RNA-image to predict two outputs, percent immune and non-immune fractions. We trained NN-RNA-image using RNA-seq data filtered using the LM22 gene set and imaging features. We evaluated performance using leave-one-out cross validation.

To benchmark our results, we analyzed samples with ESTIMATE, which is a tool for predicting tumor purity, and the presence of infiltrating stromal/immune cells in tumor tissues using gene expression data (Fig. 4).[22] We found that a neural network-based model (NN-RNA-image) could be effectively adopted to learn the total immune infiltrate proportion. We found that our NN-RNA-image trained model performed better than ESTIMATE based on overall Pearson correlation. Taken together, NEXT provides a flexible framework for integrating RNA-seq and imaging features, and for predicting estimates of tumor immune infiltrate.

3.4. *NEXT augmentation with total immune infiltration fraction (NN-Transfer)*

Motivated by our previous results estimating the total fraction of immune infiltrate using both RNA-seq and imaging features, we sought to test the fourth hypothesis that integrating estimates of the total fraction of immune infiltrate could further augment and improve the prediction of infiltration cell-type proportions. We reasoned that including the total immune

and non-immune fraction may provide additional meaningful contextual features. Concomitant predictions of the total fraction of immune infiltrate were concatenated to the RNA-seq and imaging feature representations in the second layer of the network. Consistent with previous models, the RNA-seq data was filtered using the LM22 gene list and the TPM values were log transformed. We trained the NN-Transfer model using RNA-seq and imaging features and evaluated performance using leave-one-out cross validation.

We found increased accuracy in immune infiltrate prediction as evaluated by overall Pearson correlation (R=0.735; p=4.89e-53) (Fig. 3A, NN-Transfer). This increased accuracy was driven largely by increased correlations for specific immune cell-types, including B, CD4 T, and MACs (Fig. 3B). In sum, we demonstrate the flexibility and utility of our framework by transferring additional contextual features, suggesting that other relevant histological, molecular or clinical features can be readily integrated and used for more accurate immune infiltrate prediction.

4. Discussion

This study represents an important advancement in elaborating the tumor microenvironment by predicting the tumor immunological composition of individual patients. We present a multimodal approach to estimating immune infiltration based on RNA-seq gene expression data and histopathology imaging features. We demonstrate the NEXT framework is efficient and flexible, allowing investigators to integrate pre-existing, routine clinical H&E stained slides with RNA-seq data (Fig. 1 and 2). We also demonstrate increased accuracy in predicting the abundance of key immune cell subtypes in solid tumors when compared to expert pathologist assessment of IHC (Fig. 3 and 4).

To our knowledge, this is the first report using multiple laboratory-based modalities to predict immune infiltration proportions in tumor samples and using gold standard expert pathologist reviewed IHC samples. Our particular focus on developing a generalizable and flexible framework for clinical RNA-seq and imaging data holds the potential for substantial clinical impact, including broadening widespread pathological reporting of the immune infiltrate in tumor biopsies and ultimately guiding patient treatment decisions.

We anticipate further research to fully evaluate these types of models in real-world clinical settings, and across a larger distribution and spectrum of cancer types. We note that our framework is amenable to larger datasets because it allows for larger or more layers to increase learning capacity. Larger datasets would also allow for learning unsupervised input features. Currently, our model incorporates supervised guided features of the human transcriptome and imaging data, but larger datasets can enable us to learn unsupervised H&E image features, such as through an auto-encoder, which may lead to performance boosts. Furthermore, our current model treats each cell type independently, but in some cases, the relative and absolute abundance of certain cell types may be correlated. Future work can also exploit the correlative structure of immune infiltration.

Additionally, as new routine and widespread laboratory-based techniques become adopted, our framework provides a principled approach for integrating relevant molecular and clinical features to further improve model performance. As we demonstrated in the NN-Transfer

model, the addition of other contextual information to the model can lead to better overall prediction accuracy. Other assays, such as DNA sequencing, radiological imaging, methylation profiling, immunofluorescence or other histological staining techniques, flow or mass cytometry, can be used to generate distinct features that can be integrated with the RNA and image components of the model in a similar fashion to NN-Transfer.

Finally, we note that NEXT can also be used to train a model for predicting any arbitrary mixture of cells with known proportions. For instance, instead of immune cell subtypes or total immune fraction, the approach can be adjusted to estimate the relative proportion of tumor and endothelial cells, which would provide information about how much vascularization is present in a tumor. The utility of these models is also not limited to cancer samples. Inferring the relative and absolute proportions of different cell types in complex mixtures has value in many other disease areas, like lupus and rheumatoid arthritis, and is also a useful tool in basic science research.

Acknowledgments

The authors acknowledge and thank Dr. Alexandria Bobe for critical review, suggestions, and discussion. In addition, the authors thank Dr. Jason Perera, Dr. Alan Chang, Dr. Nike Beaubier, Dr. Tim Taxter, Dr. Stephen Yip and Ms. Erin McCarthy for immuno-oncology discussions, assistance with pathology review, immunohistochemistry services, tumor sample acquisitions, RNA-sequencing, digital imaging, and assistance with data retrieval.

References

1. Rizvi, N. A., Hellmann, M. D., Snyder, A., Kvistborg, P., Makarov, V., Havel, J. J., Lee, W., Yuan, J., Wong, P., Ho, T. S., Miller, M. L., Rekhtman, N., Moreira, A. L., Ibrahim, F., Bruggeman, C., Gasmi, B., Zappasodi, R., Maeda, Y., Sander, C., Garon, E. B., Merghoub, T., Wolchok, J. D., Schumacher, T. N., and Chan, T. A. *Science (New York, N.Y.)* **348**(6230), 124–8 apr (2015).
2. Rooney, M. S., Shukla, S. A., Wu, C. J., Getz, G., and Hacohen, N. *Cell* **160**(1-2), 48–61 jan (2015).
3. Li, B., Severson, E., Pignon, J.-C., Zhao, H., Li, T., Novak, J., Jiang, P., Shen, H., Aster, J. C., Rodig, S., Signoretti, S., Liu, J. S., and Liu, X. S. *Genome biology* **17**(1), 174 dec (2016).
4. Herbst, R. S., Soria, J.-C., Kowanetz, M., Fine, G. D., Hamid, O., Gordon, M. S., Sosman, J. A., McDermott, D. F., Powderly, J. D., Gettinger, S. N., Kohrt, H. E. K., Horn, L., Lawrence, D. P., Rost, S., Leabman, M., Xiao, Y., Mokatrin, A., Koeppen, H., Hegde, P. S., Mellman, I., Chen, D. S., and Hodi, F. S. *Nature* **515**(7528), 563–567 nov (2014).
5. Jenkins, R. W., Barbie, D. A., and Flaherty, K. T. *British Journal of Cancer* (118), 9–16 (2018).
6. Anitei, M.-G., Zeitoun, G., Mlecnik, B., Marliot, F., Haicheur, N., Todosi, A.-M., Kirilovsky, A., Lagorce, C., Bindea, G., Ferariu, D., Danciu, M., Bruneval, P., Scripcariu, V., Chevallier, J.-M., Zinzindohoue, F., Berger, A., Galon, J., and Pages, F. *Clinical Cancer Research* **20**(7), 1891–1899 apr (2014).
7. Alejandro Jiménez-Sánchez, A., Memon, D., Pourpe, S., Merghoub, T., Snyder, A., Miller Correspondence snyderca, M. L., Jimé nez Sá nchez, A., Veeraraghavan, H., Li, Y., Alberto Vargas, H., Gill, M. B., Park, K. J., Zivanovic, O., Konner, J., Ricca, J., Zamarin, D., Walther, T., Aghajanian, C., Wolchok, J. D., Sala, E., and Miller, M. L. *Cell* **170**, 927–938 (2017).
8. Chen, D. S. and Mellman, I. *Nature* **541**(7637), 321–330 jan (2017).

9. Woo, S. R., Corrales, L., and Gajewski, T. F. *Annual Review of Immunology* **33**(1), 445–474 mar (2015).
10. Robinson, D. R., Wu, Y.-M., Lonigro, R. J., Vats, P., Cobain, E., Everett, J., Cao, X., Rabban, E., Kumar-Sinha, C., Raymond, V., Schuetze, S., Alva, A., Siddiqui, J., Chugh, R., Worden, F., Zalupski, M. M., Innis, J., Mody, R. J., Tomlins, S. A., Lucas, D., Baker, L. H., Ramnath, N., Schott, A. F., Hayes, D. F., Vijai, J., Offit, K., Stoffel, E. M., Roberts, J. S., Smith, D. C., Kunju, L. P., Talpaz, M., Cieślik, M., and Chinnaiyan, A. M. *Nature* **548**(7667), 297–303 (2017).
11. Khotanzad, A. and Hong, Y. H. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**(5), 489–497 May (1990).
12. Hamilton, N. A., Pantelic, R. S., Hanson, K., and Teasdale, R. D. *BMC Bioinformatics* **8**(1), 110 Mar (2007).
13. Harwood, D., Ojala, T., Pietikäinen, M., Kelman, S., and Davis, L. S. *Pattern Recognition Letters* **16**, 1–10 (1995).
14. Haralick, R. M., Shanmugam, K., and Dinstein, I. *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-3**(6), 610–621 Nov (1973).
15. Lowe, D. G. *Int. J. Comput. Vision* **60**(2), 91–110 November (2004).
16. Bankhead, P., Loughrey, M. B., Fernández, J. A., Dombrowski, Y., McArt, D. G., Dunne, P. D., McQuaid, S., Gray, R. T., Murray, L. J., Coleman, H. G., James, J. A., Salto-Tellez, M., and Hamilton, P. W. *Scientific Reports* **7**(1), 16878 (2017).
17. Coelho, L. P. *CoRR* **abs/1211.4907** (2012).
18. Newman, A. M., Liu, C. L., Green, M. R., Gentles, A. J., Feng, W., Xu, Y., Hoang, C. D., Diehn, M., and Alizadeh, A. A. *Nature methods* **12**(5), 453–7 may (2015).
19. Gong, T. and Szustakowski, J. D. *Bioinformatics* **29**(8), 1083–1085 apr (2013).
20. Narni-Mancinelli, E., Vivier, E., and Kerdiles, Y. M. *International Immunology* **23**(7), 427–431 jul (2011).
21. Tsujikawa, T., Kumar, S., Borkar, R. N., Gray, J. W., Flint, P. W., and Coussens Correspondence, L. M. *CellReports* **19**, 203–217 (2017).
22. Yoshihara, K., Shahmoradgoli, M., Martínez, E., Vegesna, R., Kim, H., Torres-Garcia, W., Treviño, V., Shen, H., Laird, P. W., Levine, D. A., Carter, S. L., Getz, G., Stemke-Hale, K., Mills, G. B., and Verhaak, R. G. *Nature Communications* **4**(1), 2612 dec (2013).

Influence of tissue context on gene prioritization for predicted transcriptome-wide association studies

Binglan Li
Genomics and Computational Biology Program, University of Pennsylvania
Philadelphia, PA 19104, USA
Email: binglan.li@pennmedicine.upenn.edu
Yogasudha Veturi, Yuki Bradford, Shefali S. Verma, Anurag Verma, Anastasia M. Lucas
Department of Genetics, University of Pennsylvania
Philadelphia, PA 19104, USA
Email: Yogasudha.Veturi@pennmedicine.upenn.edu, Yuki.Bradford@pennmedicine.upenn.edu, Shefali.SetiaVerma@pennmedicine.upenn.edu, anurag.verma@pennmedicine.upenn.edu, Anastasia.Lucas@pennmedicine.upenn.edu
David W. Haas
Departments of Medicine, Pharmacology, Pathology, Microbiology & Immunology, Vanderbilt University School of Medicine, Nashville, TN, and Department of Internal Medicine, Meharry Medical College, Nashville, TN, USA
Email: david.haas@vanderbilt.edu
Marylyn D. Ritchie*
Department of Genetics, Institute for Biomedical Informatics, University of Pennsylvania
Philadelphia, PA 19104, USA
Email: marylyn@pennmedicine.upenn.edu

Transcriptome-wide association studies (TWAS) have recently gained great attention due to their ability to prioritize complex trait-associated genes and promote potential therapeutics development for complex human diseases. TWAS integrates genotypic data with expression quantitative trait loci (eQTLs) to predict genetically regulated gene expression components and associates predictions with a trait of interest. As such, TWAS can prioritize genes whose differential expressions contribute to the trait of interest and provide mechanistic explanation of complex trait(s). Tissue-specific eQTL information grants TWAS the ability to perform association analysis on tissues whose gene expression profiles are otherwise hard to obtain, such as liver and heart. However, as eQTLs are tissue context-dependent, whether and how the tissue-specificity of eQTLs influences TWAS gene prioritization has not been fully investigated. In this study, we addressed this question by adopting two distinct TWAS methods, PrediXcan and UTMOST, which assume single tissue and integrative tissue effects of eQTLs, respectively. Thirty-eight baseline laboratory traits in 4,360 antiretroviral

* To whom correspondence should be addressed to.

The project described was supported by Award Number U01AI068636 from the National Institute of Allergy and Infectious Diseases (NIAID) and supported by National Institute of Mental Health (NIMH), National Institute of Dental and Craniofacial Research (NIDCR). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute of Allergy and Infectious Diseases or the National Institutes of Health. This work was supported by the AIDS Clinical Trials Group funded by the National Institute of Allergy and Infectious Diseases (AI068636, AI038858, AI068634, AI038855). Additional grant support included AI077505, AI069439, TR000445, AI054999, AI116794 and AI110527.

Clinical Research Sites that participated in ACTG protocol A5202, and collected DNA under protocol A5128, were supported by the following grants from NIAID: AI069477, AI027675, AI073961, AI069474, AI069432, AI069513, AI069423, AI050410, AI069452, AI069450, AI054907, AI069428, AI069439, AI069467, AI045008, AI069495, AI069415, AI069556, AI069484, AI069424, AI069532, AI069419, AI069471, AI025859, AI069418, AI050409, AI069423, AI069501, AI069502, AI069511, AI069481, AI069465, AI069494, AI069472, AI069470, AI046376, AI072626, AI027661, AI034853, AI069447, AI032782, AI027658, AI-27666, AI058740, and AI046370, and by the following grants from the National Center for Research Resources (NCRR): RR00051, RR00046, RR025747, RR025777, RR024160, RR024996, RR024156, RR024160, and RR024160. Study drugs were provided by Bristol-Myers Squibb Co., Gilead Sciences, and GlaxoSmithKline, Inc.

treatment-naïve individuals from the AIDS Clinical Trials Group (ACTG) studies comprised the input dataset for TWAS. We performed TWAS in a tissue-specific manner and obtained a total of 430 significant gene-trait associations (q-value < 0.05) across multiple tissues. Single tissue-based analysis by PrediXcan contributed 116 of the 430 associations including 64 unique gene-trait pairs in 28 tissues. Integrative tissue-based analysis by UTMOST found the other 314 significant associations that include 50 unique gene-trait pairs across all 44 tissues. Both analyses were able to replicate some associations identified in past variant-based genome-wide association studies (GWAS), such as high-density lipoprotein (HDL) and *CETP* (PrediXcan, q-value = 3.2e-16). Both analyses also identified novel associations. Moreover, single tissue-based and integrative tissue-based analysis shared 11 of 103 unique gene-trait pairs, for example, *PSRC1*-low-density lipoprotein (PrediXcan's lowest q-value = 8.5e-06; UTMOST's lowest q-value = 1.8e-05). This study suggests that single tissue-based analysis may have performed better at discovering gene-trait associations when combining results from all tissues. Integrative tissue-based analysis was better at prioritizing genes in multiple tissues and in trait-related tissue. Additional exploration is needed to confirm this conclusion. Finally, although single tissue-based and integrative tissue-based analysis shared significant novel discoveries, tissue context-dependency of eQTLs impacted TWAS gene prioritization. This study provides preliminary data to support continued work on tissue context-dependency of eQTL studies and TWAS.

Keywords: TWAS; integrative; context; PrediXcan; UTMOST.

1. Introduction

Improving antiretroviral therapy (ART) efficacy and safety is an ongoing goal for addressing the HIV pandemic. According to the Joint United Nations Programme on HIV and AIDS (UNAIDS) (http://aidsinfo.unaids.org/), approximately 36.7 million people worldwide were living with human immunodeficiency virus (HIV) in 2016. Over the past three decades there has been immense progress on HIV care and treatment, and in 2017 there were about 20.9 million HIV-positive people who had access to ART. The connection of genomics with pharmacology has led to the discovery of numerous single nucleotide polymorphisms (SNPs) in drug absorption, distribution, metabolism, and elimination (ADME) genes and off-target genes. Many SNPs have been related to effects and/or pharmacokinetics of antiretroviral drugs[1-6]. However, most trait-related SNPs lack connections to actual functional genes, which suggests the need for alternative analysis approaches.

The emerging field of transcriptome-wide association studies (TWAS) offer a new way to directly identify gene-trait associations via integration of genotypic data and expression quantitative trait loci (eQTLs). eQTLs are an important class of genetic functional elements, which affect transcriptional regulation on target genes. Integration of eQTL information with genotypic data allows TWAS to estimate the extent to which a gene's expression level is regulated by genetic variants and how this correlates with traits of interest[8]. The Genotype Tissue Expression Project (GTEx[7]) provides the data and the opportunity to identify eQTLs and estimate effect sizes for multiple human tissues (44 tissues in GTEx v6p). With GTEx, TWAS can explore gene-trait associations on tissues whose gene expression profiles are otherwise hard to obtain, such as liver and heart. However, current TWAS focuses primarily on eQTLs identified in a tissue-by-tissue manner, while many studies have either acknowledged or supported the power of an integrative tissue context in identifying single-tissue and multi-tissue eQTLs[9,10].

In this study, we aimed to address whether and how single tissue and integrative tissue context of eQTLs influence TWAS gene prioritization by comparing two distinct TWAS methods,

PrediXcan[11] and Unified Test for MOlecular SignaTures (UTMOST[12]). PrediXcan uses elastic-net regression model and identifies eQTLs in a tissue-by-tissue manner. UTMOST adopts group-lasso and search through all tissues at once to spot eQTLs of a certain gene. This strategy allows UTMOST to identify single-tissue specific eQTLs similar to PrediXcan but increase the chance of detecting multi-tissue eQTLs. Here, 38 baseline (i.e. pre-ART) laboratory values and genotypic data of 4,360 ACTG clinical trials participants from multiple previous studies[13-19] comprised the input for TWAS. Genotyping had been previously generated in multiple phases with Illumina assays: 650Y (phase I), 1M Duo (phase II and III), or Human Core Exome (phase IV). We performed the two TWAS methods separately in a tissue-specific manner (i.e. 44 tissues) (**Figure 1**). If tissue context-dependency of eQTLs did not affect TWAS gene prioritization, we expected to observe shared gene-trait associations between single tissue-based analysis (PrediXcan) and integrative tissue-based analysis (UTMOST). The results partially supported this hypothesis, but also suggested varied gene prioritization abilities of single tissue-based and integrative tissue-based approaches respectively. The former found more unique gene-trait pairs, while the latter tended to prioritize genes expressed in multiple tissues. This study provides supportive evidence for tissue context-dependency of eQTLs and its impact on TWAS gene prioritization.

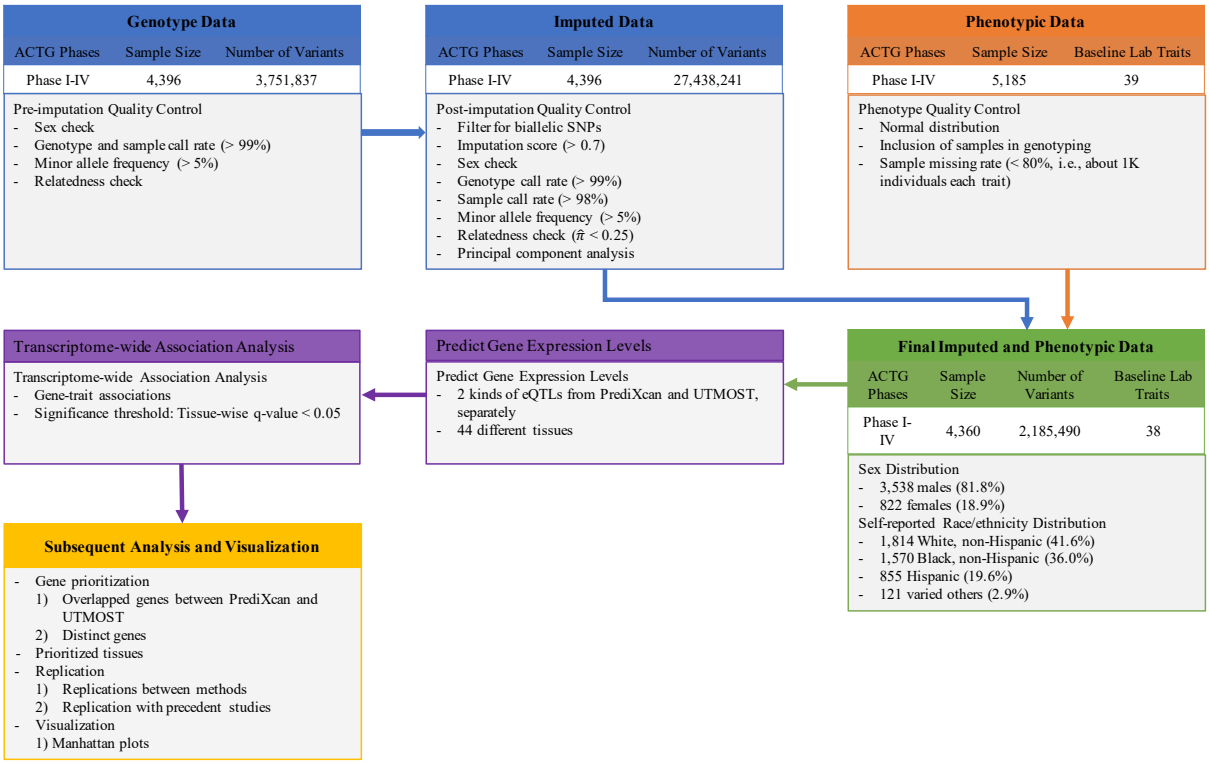


Figure 1. This study investigates the influence of tissue context-dependency of eQTLs on TWAS gene prioritization by comparing two distinct TWAS methods, PrediXcan and UTMOST. PrediXcan assumes single tissue context of eQTLs, while UTMOST assumes eQTLs to possibly have effects in multiple tissues.

2. Methods

2.1. *Data and Study Participants*

In this study, we used four different genotyping phases of ACTG studies in a combined dataset that

included samples and data from participants in prospective, randomized ART-naïve treatment trials[13-19]. Clinical trial designs and results, and results of a genome-wide pleiotropic study results for baseline laboratory values have been described elsewhere[13-21].

2.2. *Quality Control*

2.2.1. *Genotypic data*

A total of 4,393 individuals were genotyped in four phases. Phase I was genotyped using Illumina 650Y array; Phase II and III were genotyped using Illumina 1M duo array; Phase IV was genotyped using Illumina HumanCoreExome BeadChip.

The computational preparation of genotypic data included pre-imputation quality control (QC), imputation, and post-imputation quality control. Pre- and post-imputation quality control followed the same guidelines[22] and used PLINK1.90[23] and R programming language. Imputation was performed on ACTG phase I-IV combined genotype data. Genotyped variants surviving the pre-imputation quality control comprised the input datasets for imputation, which used IMPUTE2[24] with 1000 Genomes[25] Phase 1 v3 as the reference panel. ACTG phase I-IV combined imputed data had 4,941 individuals and 27,438,241 variants. The following procedures/parameters were used in the post-imputation quality control by PLINK1.90: sample inclusion in phase I-IV phenotype collection, biallelic SNP check, imputation score (> 0.7), sex check, genotype call rate (> 99%), sample call rate (> 98%), and minor allele frequency (MAF > 5%), and relatedness check ($\hat{\pi} > 0.25$). Subsequent principal component analysis (EIGENSOFT[26]) projected remaining individuals onto the 1000 Genomes Project sample space to examine for population stratification. The first three principal components were used as covariates to adjust for population structure in the subsequent analysis. The final QC'ed ACTG phase I-IV combined imputed data contained 2,185,490 genotyped and imputed biallelic SNPs for 4,360 individuals (**Figure 1**).

2.2.2. *Phenotypic data*

The ACTG clinical trials included in this analysis collected baseline (i.e., pre-ART) laboratory traits from 5,185 ART-naïve individuals. We only retained individuals who have been genotyped and traits that were normally distributed and met a criterion of phenotype missing rate < 80%. The final combined phenotype dataset of ACTG genotyping phase I-IV retained 38 traits and the same number of individuals as the QC'ed imputed dataset (**Figure 1**).

2.3. *Predict Unmeasured Gene Expression Levels*

We adopted two TWAS methods, PrediXcan and UTMOST, to predict unmeasured gene expression levels in a tissue-specific manner. PrediXcan and UTMOST have estimated SNP effect sizes on gene expression levels in 44 tissues, which are available at http://predictdb.org/ and https://github.com/Joker-Jerome/UTMOST, respectively. The PrediXcan and UTMOST scripts were pulled from their GitHub project repositories on April 23rd and Jun 6th, 2018, respectively.

PrediXcan and UTMOST followed the same multivariate models. Let N denote the sample size and M denote the number of eQTLs in a certain gene. A gene's expression level can be predicted using the multivariate model as follows:

$$E = X\beta \quad (1)$$

where E is the $N \times 1$ vector of predicted gene expression levels of the gene, X is the $N \times M$ matrix of genotypes, and β is the $M \times 1$ vector of eQTLs' estimated regulatory effects on the gene.

Predicted gene expression levels were likely to differ between the two methods as each has a different hypothesis of eQTL regulatory mechanisms in terms of tissue context-dependency. To discover trait-related tissues without assumptions, we predicted gene expression levels in 44 tissues.

2.4. *Transcriptome-wide Association Analysis*

We tested for gene-trait associations by performing transcriptome-wide association tests on predicted gene expression levels and ACTG baseline lab traits using PLATO[27,28]. All baseline lab traits included in this study were continuous and thus were modeled using linear regression. Age, sex, and the first three principal components calculated by EIGENSOFT were included as covariates in linear models to adjust for sampling biases and underlying population structure. PrediXcan and UTMOST have different degrees of diversity in the number of eGenes and gene-trait associations among tissues. To avoid biases due to an uneven number of associations among tissues, p-values were adjusted using FDR with using Benjamini–Hochberg procedure[29] in a tissue-specific manner. For this study, we consider gene-trait associations significant if they had single tissue-wise q-value < 0.05.

3. Results

We compared the influence of tissue context-dependency of eQTLs on TWAS gene prioritization by comparing single tissue-based analysis (PrediXcan) and integrative tissue-based analysis (UTMOST). We performed TWAS on ACTG phase I-IV combined datasets. The data aggregation of ACTG phase I-IV provided a larger sample size to ensure the power of identifying gene-trait association. QC procedures left the ACTG phase I-IV combined imputed data with 4,360 individuals and 2,185,490 SNPs. There were 38 baseline lab traits in the final phenotypic datasets.

Single tissue-based and integrative tissue-based analysis identified a total of 430 significant gene-trait associations (103 unique gene-trait pairs regardless of tissue, q-value < 0.05) and share 11 unique gene-trait pairs. Single tissue-based analysis identified 116 of the 430 significant associations (64 unique gene-trait pairs), encompassing 41 genes, 17 traits, and 28 tissues. Integrative tissue-based analysis identified the remaining 314 significant associations (50 unique gene trait pairs), encompassing 38 genes, 20 traits, and all 44 tissues.

3.1. *Tissue Context-dependency Influenced TWAS Gene Prioritization*

Gene prioritization results from single tissue-based analysis (PrediXcan) and integrative tissue-based analysis (UTMOST) were compared to evaluate the influence of tissue context-dependency of eQTLs on TWAS. Single and integrative tissue-based analyses shared 11 of 103 unique gene-trait pairs regardless of tissue (**Table 1**). Several of these replicated the findings of previous studies (**Table 2**). The lowest p-value by integrative tissue-based analysis was for *MROH2A*-total bilirubin levels[20] (UTMOST, q-value = 6.0e-27), which had a moderate p-value from single tissue-based analysis (q-value = 0.005). Another replication was between *PSRC1* and two lipid-related traits, cholesterol and LDL, which have been reported in other studies[30-33]. Although it was *SORT1,* which neighbors *PSRC1*, that has been functionally related to LDL via mice knockdown experiments[34]. *ALDH5A1* and *GPLD1* have been associated with the liver function test, alkaline phosphatase (ALP)[35]. In the cases of *PSRC1*, *ALDH5A1*, and *GPLD1*, integrative tissue-based analysis (UTMOST) prioritized the genes in their biological function-related organ, liver, which was not always the case for single tissue-based analysis (PrediXcan). Possible novel associations were observed between absolute neutrophil count and *C1orf204*[36], *ATF6*, and *VANGL2*[37].

Table 1. Significant gene-trait associations (q-value < 0.05) shared by single and integrative tissue-based analysis. The two different analyses shared 11 out of 103 unique significant gene-trait pairs.

Traits	Genes	Methods	#Tissues	Major Tissue Types*
Absolute neutrophil count	*ATF6*	PrediXcan	1	Brain
	ATF6	UTMOST	2	Brain, Transformed Fibroblasts
	C1orf204	PrediXcan	1	Brain
	C1orf204	UTMOST	5	Brain, Ovary, Pituitary
	VANGL2	PrediXcan	1	Brain
	VANGL2	UTMOST	1	Brain
Alkaline phosphatase	*ALDH5A1*	PrediXcan	9	Artery, Colon, Liver, Lung, Nerve, Pancreas, Skin, Thyroid, Transformed Lymphocytes
	ALDH5A1	UTMOST	39	Adipose, Adrenal Gland, Artery, Brain, Breast, Colon, Esophagus, Heart, Liver, Lung, Nerve, Ovary, Pancreas, Pituitary, Prostate, Skeletal Muscle, Skin, Small Intestine, Spleen, Stomach, Testis, Thyroid, Transformed Lymphocytes, Uterus, Vagina
	GPLD1	PrediXcan	2	Artery, Thyroid
	GPLD1	UTMOST	24	Adipose, Artery, Brain, Esophagus, Heart, Liver, Lung, Nerve, Pituitary, Prostate, Skeletal Muscle, Skin, Small Intestine, Stomach, Testis, Thyroid, Transformed Lymphocytes, Vagina, Whole Blood
Cholesterol	*PSRC1*	PrediXcan	9	Brain, Esophagus, Lung, Pancreas, Pituitary, Skeletal Muscle, Skin, Whole Blood
	PSRC1	UTMOST	25	Adipose, Brain, Breast, Colon, Esophagus, Heart, Liver, Lung, Nerve, Ovary, Pancreas, Pituitary, Prostate, Skeletal Muscle, Skin, Testis, Uterus, Whole Blood
Fasting cholesterol	*PSRC1*	PrediXcan	9	Brain, Esophagus, Lung, Pancreas, Pituitary, Skeletal Muscle, Skin, Whole Blood
	PSRC1	UTMOST	22	Adipose, Brain, Breast, Colon, Esophagus, Heart, Liver, Lung, Nerve, Ovary, Pituitary, Prostate, Skeletal Muscle, Skin, Testis, Uterus, Whole Blood
Fasting LDL	*PSRC1*	PrediXcan	11	Brain, Esophagus, Lung, Pancreas, Pituitary, Skeletal Muscle, Skin, Testis, Thyroid, Whole Blood
	PSRC1	UTMOST	27	Adipose, Brain, Breast, Colon, Esophagus, Heart, Liver, Lung, Nerve, Ovary, Pancreas, Pituitary, Prostate, Skeletal Muscle, Skin, Testis, Thyroid, Uterus, Whole Blood
Hemoglobin	*CAMSAP1*	PrediXcan	1	Nerve
	CAMSAP1	UTMOST	31	Adipose, Artery, Brain, Breast, Colon, Esophagus, Heart, Liver, Lung, Nerve, Ovary, Prostate, Skeletal Muscle, Skin, Small Intestine, Spleen, Thyroid, Transformed Fibroblasts, Transformed Lymphocytes, Whole Blood
LDL	*PSRC1*	PrediXcan	11	Brain, Esophagus, Lung, Pancreas, Pituitary, Skeletal Muscle, Skin, Testis, Thyroid, Whole Blood
	PSRC1	UTMOST	27	Adipose, Brain, Breast, Colon, Esophagus, Heart, Liver, Lung, Nerve, Ovary, Pancreas, Pituitary, Prostate, Skeletal Muscle, Skin, Testis, Thyroid, Uterus, Whole Blood
Total bilirubin	*MROH2A*	PrediXcan	1	Adipose
	MROH2A	UTMOST	1	Stomach

* For simplicity, only major tissue types were shown. Skin, heart, esophagus, colon, brain, artery, and adipose have subtypes.

Table 2. Validation of some of the TWAS prioritized genes.

GENES	METHODS	TISSUES	Q-VALUE+	ACTG TRAITS	GWAS CATALOG REPORTED TRAITS	PMID
ATF6	PrediXcan UTMOST	Brain Transformed Fibroblasts*, Brain	1.30E-02 1.63E-02	Absolute neutrophil count	White blood cell count	28158719
VANGL2	PrediXcan UTMOST	Brain Brain	1.30E-02 4.70E-02	Absolute neutrophil count	Multiple sclerosis	24076602
ADAMTS4	UTMOST	Artery	1.50E-04	Absolute neutrophil count*, White blood cell count	Monocyte percentage of white cells	27863252
ALDH5A1	PrediXcan UTMOST	Colon*, Artery, **Liver**, Lung, Nerve, Pancreas, Skin, Thyroid, Transformed Lymphocytes Artery*, Adipose, Adrenal Gland, Brain, Breast, Colon, Esophagus, Heart, **Liver**, Lung, Nerve, Ovary, Pancreas, Pituitary, Prostate, Skeletal Muscle, Skin, Small Intestine, Spleen, Stomach, Testis, Thyroid, Transformed Lymphocytes, Uterus, Vagina	1.57E-05 6.58E-03	Alkaline phosphatase Alkaline phosphatase	Liver enzyme levels (alkaline phosphatase)	22001757
ITLN1	PrediXcan	Stomach	1.04E-05	Alkaline phosphatase, Absolute basophil count, Triglyceride, Viral load	Crohn's disease	18587394
CELSR2	PrediXcan	Brain*, Skeletal Muscle	6.67E-06	Cholesterol, Fasting cholesterol, Fasting LDL, LDL	Total cholesterol, LDL	20686565, 17903299
PSRC1	PrediXcan UTMOST	Lung*, Brain, Esophagus, Pancreas, Pituitary, Skeletal Muscle, Skin, Whole Blood Heart*, Adipose, Brain, Breast, Colon, Esophagus, **Liver**, Lung, Nerve, Ovary, Pancreas, Pituitary, Prostate, Skeletal Muscle, Skin, Testis, Thyroid, Uterus, Whole Blood	8.47E-06 1.75E-05	LDL*, Cholesterol, Fasting cholesterol, Fasting LDL	Total cholesterol, LDL	20686565, 17903299, 19936222, 17903299, 25101658
CETP	PrediXcan	Colon	3.24E-17	HDL*, Fasting HDL	HDL cholesterol	25884002, 20686565
MROH2A	PrediXcan UTMOST	Adipose Stomach	5.23E-03 5.97E-27	Total bilirubin	Bilirubin levels	25884002, 21646302
UGT1A1	PrediXcan	Skin	7.13E-07	Total bilirubin	Bilirubin levels	25884002, 21646302
UGT1A7	UTMOST	Skin*, Adrenal Gland, Colon, Esophagus, **Liver**, Stomach	5.15E-40	Total bilirubin	Bilirubin levels	25884002, 21646302
APOA1	PrediXcan	Brain	2.93E-02	Triglyceride	Total cholesterol, Triglyceride, LDL, HDL	20686565, 17903299
APOC3	PrediXcan	Heart	1.61E-02	Triglyceride	Total cholesterol, Triglyceride, LDL, HDL	20686565, 17903299

Bolded tissues are known trait-related tissues.
* denotes the most significant tissue and/or trait that were associated with genes.
+ q-value in the most significant tissue denoted by asterisk.

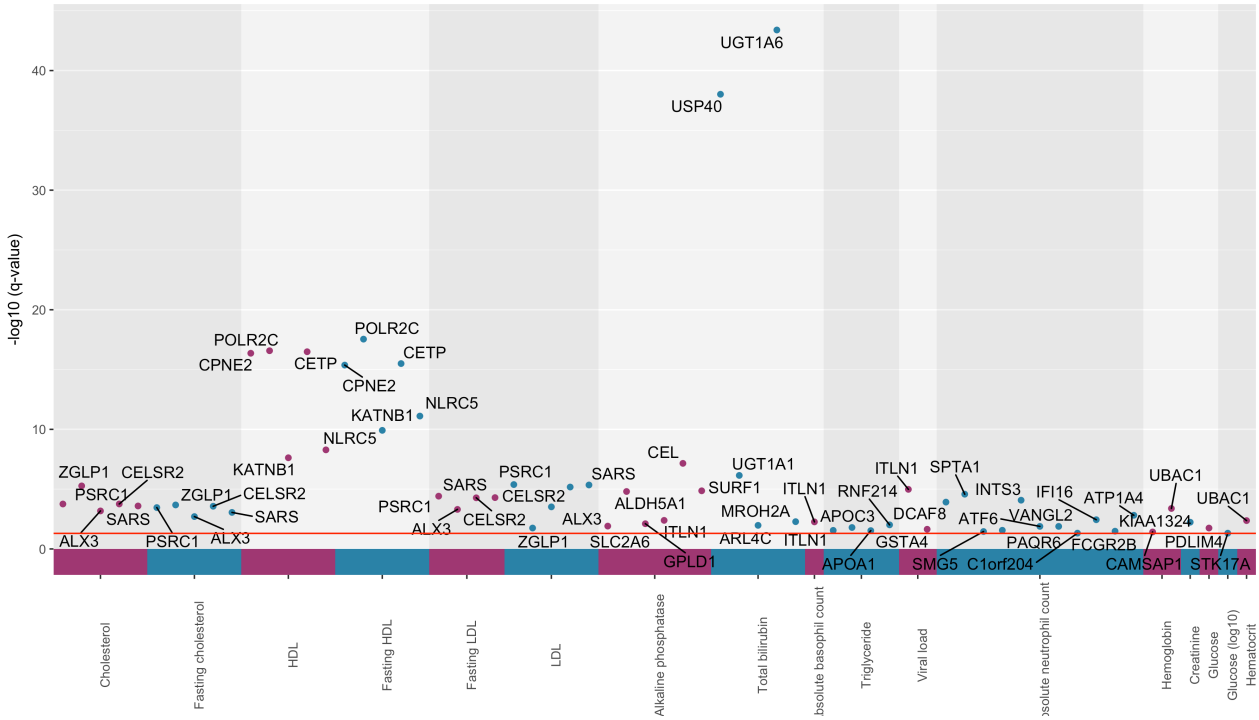


Figure 2. Manhattan plot of gene-trait associations identified by PrediXcan. X-axis showed only significant traits. Y-axis was the q-value transformed by -log10. For simplicity, the plot only shows the lowest p-value of a gene-trait pair, which may appear in multiple tissues.

3.2. *Single Tissue-based Analysis Found a Greater Number of Unique Gene-trait Associations*

Single tissue-based analysis using PrediXcan identified 64 unique gene-trait association across different tissues (**Figure 2**). Some associations have been reported previously (**Table 2**). PrediXcan associated total bilirubin levels with *UGT1A1*[20] (skin, q-value = 7.1e-07) and *MROH2A*[20] (adipose, q-value = 0.005), and LDL and cholesterol to *CELSR2* [30,38,39] (most significant with LDL in brain, q-value = 6.7e-06). HDL was associated with *CETP*[20,32] (most significant in colon with q-value = 3.2e-17) and *NLRC5*[38] (adrenal gland, q-value = 7.8e-12). Triglyceride was associated with *APOA1*[30,39] (brain, q-value = 0.029) and *APOC3*[30,39] (heart, q-value = 0.016).

Single tissue-based analysis identified novel gene-trait associations, which warrants further investigation. One interesting example was the association of *ITLN1* with multiple traits, including HIV-1 viral load, triglyceride, and total neutrophil count. As *ITLN1* was reported in a previous Crohn's disease study[40], our result suggested an potential relationship between Crohn's disease and HIV infection[41].

3.3. *Integrative Tissue-based Analysis Found Multi-tissue Gene-trait Associations*

Regardless of tissue, integrative tissue-based analysis using UTMOST identified 50 unique gene-trait pairs (**Figure 3**). Although it prioritized fewer genes, the integrative tissue-based analysis was more likely to prioritize multiple tissues where genes are expressed. For instance, *PSRC1* is highly expressed in almost all tissues[7]. *PSRC1*-LDL and cholesterol associations were prioritized in at least ten more tissues by integrative tissue-based analysis Most importantly, they were found consistently in the liver which is critically involved in lipid regulation. There was some evidence for distinct

associations identified via integrative tissue-based approach (**Table 2**), such as *ADAMTS4*[42] with white blood cell count (artery, q-value = 0.023), and *AMFR*[43] with fasting HDL (most significant in heart, q-value = 3.2e-05).

Other prioritized genes suggested novel associations and potential pleiotropy. Most prioritized genes have been associated with other traits by GWAS according to GWAS Catalog[44]. Similar to the single tissue-based approach, integrative tissue-based analysis prioritized total bilirubin-associated genes from the *UGT1A*[45] gene locus (*UGT1A7* and *UGT1A10*) across multiple tissues.

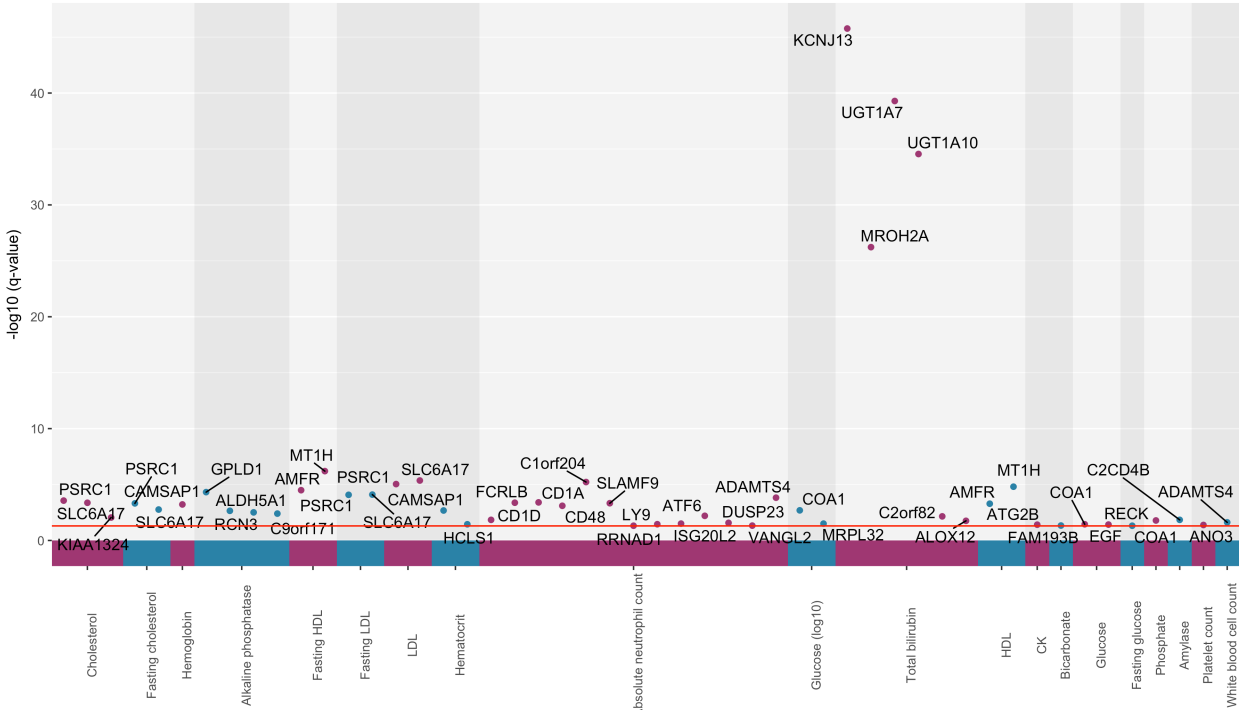


Figure 3. Manhattan plot of gene-trait associations identified by UTMOST. X-axis showed only significant traits. Y-axis was the q-value transformed by -log10. For simplicity, the plot only showed the most significant p-value of a gene-trait pair, which may appear in multiple tissues.

4. Discussions

This study investigated whether and how TWAS gene prioritization was influenced by tissue context-dependency of eQTLs by comparing two approaches, single tissue-based TWAS (implemented in PrediXcan) and integrative tissue-based TWAS (implemented in UTMOST). PrediXcan evaluated eQTLs' effects in the context of a single tissue, which did not consider potential multi-tissue effects of eQTLs. UTMOST estimated eQTLs' effect in an integrative tissue setting and increase the chance of identifying multi-tissue eQTLs. We found that both types of analyses could replicate associations discovered by previous studies and identify novel ones. While there were a fair number of overlaps, the two types of analyses prioritized different sets of genes. Single tissue-based analysis identified more unique gene-trait associations. Integrative tissue-based analysis tended to prioritize the same associations in multiple tissues and most importantly association were found in tissues critically related to traits of interest. Results suggest that tissue context-dependency of eQTLs influenced TWAS gene prioritization results.

The comparison raised questions of power and type I error rate of tested TWAS approaches. Integrative tissue context has shown an improved power in identifying eQTLs. As such, integrative

tissue-based analysis might have universally greater power in identifying trait-associated genes than single tissue-based analysis. However, in this study, single tissue-based analysis found more validated associations (**Table 2**). It is hard to tell if integrative tissue-based analysis has universally greater power as expected, whereas single tissue-based analysis happened to identify more false positives. It is also possible that one type of analysis outperformed the other at certain scenarios. A simulation study is necessary to discern these possibilities.

Similar to GWAS, prioritized genes might merely be tag genes for causal ones. Both kinds of analyses prioritized genes at the chromosome 1p13.3 locus where a lipid-related gene, *SORT1*, is located. Single tissue-based analysis associated multiple lipid-related traits with genes that neighbor *SORT1*, such as *SARS*, *CELSR2*, *PSRC1*, and *ALX3*, which all are in the 1p13.3 locus and the same topologically associating domain (TAD[46,47]). Besides *PSRC1*, integrative tissue-based analysis repetitively identified *SLC6A17*. Even though it is not adjacent to *SORT1*, this gene is in the 1p13.3 locus and might serve as a tag gene for causal one(s). Hence, for TWAS, prioritized genes might be merely tag genes and fine-mapping of causal genes may need a larger search boundary than GWAS, such as TADs.

Future investigation or validation experiments may be needed to explain the prioritized genes and/or tissues. For example, *UGT1A1* glucuronidates bilirubin in the liver[48], but single tissue-based analysis only identified a *UGT1A1*-total bilirubin association in skin. Further analysis found that there was no single *UGT1A1* eQTL identified in liver by either PrediXcan or UTMOST trained on GTEx v6p or v7 data. It is likely that identification of *UGT1A1* eQTLs is limited by tissue sample size ($N_{liver} = 175$) or genetic variants may regulate *UGT1A1* via mechanisms other than transcriptional regulation. Another observation of this study was that genes adjacent to *UGT1A1* sporadically showed up as significant in either single tissue-based or integrative tissue-based analysis, including *USP40*, *UGT1A6*, *UGT1A7*, *UGT1A10*, *KCNJ13*, and also *MROH2A*[20]. These genes span 1Mbp in chromosome 2 and locate within the same TAD[46,47]. The repetitive pattern may suggest a specific regulatory activity that targets the whole genetic region of *KCNJ13-USP40-UGT1A-MROH2A*.

TWAS can prioritize trait-related genes, which may be important for HIV-positive patients regarding genetically informed therapeutic development and drug safety. This study showed that TWAS were able to not only replicate known associations, but also identify novel gene-trait associations. It also suggested the importance of biological context in eQTL studies, and the ensemble of TWAS methods with different transcriptional regulation assumptions gave a more comprehensive picture of gene-trait relationships. In the future, we would like to perform cross-tissue TWAS analysis[12,49], which aggregate gene-trait association information across all tissues and even across different consortia to further prioritize the trait-related genes and better describe the genetic architecture of complex diseases.

5. Acknowledgments

The authors are grateful to the many persons living with HIV who volunteered for A5095, A5142, ACTG 384, A5202, and A5257. In addition, they acknowledge the contributions of study teams and site staff for these protocols. We thank Paul J. McLaren, PhD (Public Health Agency of Canada, Winnipeg, Canada) for prior involvement and collaborations that used these genome-wide genotype data. Study drugs were provided by DuPont Pharmaceutical Company, Bristol-Myers Squibb, Inc., Agouron Pharmaceuticals, Inc., GlaxoWellcome, Inc., Merck and Co., Inc. Boehringer-Ingelheim Pharmaceuticals, Inc., Gilead Sciences, Inc., GlaxoSmithKline, Inc., Abbott Laboratories, Inc., Tibotec Therapeutics. The clinical trials were ACTG 384 (ClinicalTrials.gov: NCT00000919),

A5095 (NCT00013520), A5142 (NCT00050895), A5202 (NCT00118898), and A5257 (NCT00811954). We also thank Yiming Hu from Yale University and Alvaro Barbeira and Dr. Hae Kyung Im from the University of Chicago for their support.

References

1. Mallal, S. *et al.* HLA-B*5701 screening for hypersensitivity to abacavir. *N. Engl. J. Med.* **358,** 568–579 (2008).
2. Rotger, M. *et al.* Gilbert syndrome and the development of antiretroviral therapy-associated hyperbilirubinemia. *J. Infect. Dis.* **192,** 1381–1386 (2005).
3. Holzinger, E. R. *et al.* Genome-wide association study of plasma efavirenz pharmacokinetics in AIDS Clinical Trials Group protocols implicates several CYP2B6 variants. *Pharmacogenetics and Genomics* **22,** 858–867 (2012).
4. Haas, D. W. *et al.* Pharmacogenetics of efavirenz and central nervous system side effects: an Adult AIDS Clinical Trials Group study. *AIDS* **18,** 2391–2400 (2004).
5. Lubomirov, R. *et al.* ADME pharmacogenetics: investigation of the pharmacokinetics of the antiretroviral agent lopinavir coformulated with ritonavir. *Pharmacogenetics and Genomics* **20,** 217 (2010).
6. Yuan, J. *et al.* Toxicogenomics of nevirapine-associated cutaneous and hepatic adverse events among populations of African, Asian, and European descent. *AIDS* **25,** 1271–1280 (2011).
7. GTEx Consortium *et al.* Genetic effects on gene expression across human tissues. *Nature Publishing Group* **550,** 204–213 (2017).
8. Li, B. *et al.* Evaluation of PrediXcan for prioritizing GWAS associations and predicting gene expression. in 448–459 (WORLD SCIENTIFIC, 2017). doi:10.1142/9789813235533_0041
9. Liu, X. *et al.* Functional Architectures of Local and Distal Regulation of Gene Expression in Multiple Human Tissues. *American journal of human genetics* **100,** 605–616 (2017).
10. Sul, J. H., Han, B., Ye, C., Choi, T. & Eskin, E. Effectively identifying eQTLs from multiple tissues by combining mixed model and meta-analytic approaches. *PLoS Genet* **9,** e1003491 (2013).
11. Gamazon, E. R. *et al.* A gene-based association method for mapping traits using reference transcriptome data. *Nat Genet* **47,** 1091–1098 (2015).
12. Hu, Y. *et al.* A statistical framework for cross-tissue transcriptome-wide association analysis. *bioRxiv* 286013 (2018). doi:10.1101/286013
13. Robbins, G. K. *et al.* Comparison of sequential three-drug regimens as initial therapy for HIV-1 infection. *N. Engl. J. Med.* **349,** 2293–2303 (2003).
14. Gulick, R. M. *et al.* Triple-nucleoside regimens versus efavirenz-containing regimens for the initial treatment of HIV-1 infection. *N. Engl. J. Med.* **350,** 1850–1861 (2004).
15. Gulick, R. M. *et al.* Three- vs four-drug antiretroviral regimens for the initial treatment of HIV-1 infection: a randomized controlled trial. *JAMA* **296,** 769–781 (2006).
16. Riddler, S. A. *et al.* Class-sparing regimens for initial treatment of HIV-1 infection. *N. Engl. J. Med.* **358,** 2095–2106 (2008).
17. Sax, P. E. *et al.* Abacavir-lamivudine versus tenofovir-emtricitabine for initial HIV-1 therapy. *N. Engl. J. Med.* **361,** 2230–2240 (2009).
18. Daar, E. S. *et al.* Atazanavir Plus Ritonavir or Efavirenz as Part of a 3-Drug Regimen for Initial Treatment of HIV-1: A Randomized Trial. *Ann Intern Med* **154,** 445–456 (2011).
19. Lennox, J. L. *et al.* A Phase III Comparative Study of the Efficacy and Tolerability of Three Non-Nucleoside Reverse Transcriptase Inhibitor-Sparing Antiretroviral Regimens for Treatment-Naïve HIV-1-Infected Volunteers: A Randomized, Controlled Trial. *Ann Intern Med* **161,** 461–471 (2014).
20. Moore, C. B. *et al.* Phenome-wide Association Study Relating Pretreatment Laboratory Parameters With Human Genetic Variants in AIDS Clinical Trials Group Protocols. *Open Forum Infectious Diseases* **2,** ofu113–ofu113 (2015).
21. Verma, A. *et al.* Multiphenotype association study of patients randomized to initiate antiretroviral regimens in AIDS Clinical Trials Group protocol A5202. *Pharmacogenetics and Genomics* **27,** 101–111 (2017).
22. Turner, S. *et al.* Quality control procedures for genome-wide association studies. *Curr Protoc Hum Genet* **Chapter 1,** Unit1.19–1.19.18 (2011).
23. Purcell, S. *et al.* PLINK: a tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics* **81,** 559–575 (2007).
24. Howie, B. N., Donnelly, P. & Marchini, J. A Flexible and Accurate Genotype Imputation Method for the Next Generation of Genome-Wide Association Studies. *PLoS Genet* **5,** e1000529 (2009).

25.1000 Genomes Project Consortium *et al.* A map of human genome variation from population-scale sequencing. *Nature Publishing Group* **467,** 1061–1073 (2010).
26.Price, A. L. *et al.* Principal components analysis corrects for stratification in genome-wide association studies. *Nat Genet* **38,** 904–909 (2006).
27.Hall, M. A. *et al.* PLATO software provides analytic framework for investigating complexity beyond genome-wide association studies. *Nature Communications* **8,** 1167 (2017).
28.Grady, B. J. *et al.* Finding unique filter sets in PLATO: a precursor to efficient interaction analysis in GWAS data. *Pac Symp Biocomput* 315–326 (2010). doi:10.1142/7628;page:string:Article/Chapter
29.Benjamini, Y. & Hochberg, Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)* **57,** 289–300 (1995).
30.Teslovich, T. M. *et al.* Biological, clinical and population relevance of 95 loci for blood lipids. *Nature* **466,** 707–713 (2010).
31.Chasman, D. I. *et al.* Forty-three loci associated with plasma lipoprotein size, concentration, and cholesterol content in genome-wide analysis. *PLoS Genet* **5,** e1000730 (2009).
32.Willer, C. J. *et al.* Discovery and refinement of loci associated with lipid levels. *Nat Genet* **45,** 1274–1283 (2013).
33.Kathiresan, S. *et al.* A genome-wide association study for blood lipid phenotypes in the Framingham Heart Study. *BMC Med. Genet.* **8 Suppl 1,** S17 (2007).
34.Strong, A., Patel, K. & Rader, D. J. Sortilin and lipoprotein metabolism: making sense out of complexity. *Curr. Opin. Lipidol.* **25,** 350–357 (2014).
35.Chambers, J. C. *et al.* Genome-wide association study identifies loci influencing concentrations of liver enzymes in plasma. *Nat Genet* **43,** 1131–1138 (2011).
36.Mero, I.-L. *et al.* Oligoclonal band status in Scandinavian multiple sclerosis patients is associated with specific genetic risk alleles. *PLoS ONE* **8,** e58352 (2013).
37.International Multiple Sclerosis Genetics Consortium (IMSGC) *et al.* Analysis of immune-related loci identifies 48 new susceptibility variants for multiple sclerosis. *Nat Genet* **45,** 1353–1360 (2013).
38.Weissglas-Volkov, D. *et al.* Genomic study in Mexicans identifies a new locus for triglycerides and refines European lipid loci. *J. Med. Genet.* **50,** 298–308 (2013).
39.Willer, C. J. *et al.* Discovery and refinement of loci associated with lipid levels. *Nat Genet* **45,** 1274–1283 (2013).
40.Barrett, J. C. *et al.* Genome-wide association defines more than 30 distinct susceptibility loci for Crohn's disease. *Nat Genet* **40,** 955–962 (2008).
41.Lautenbach, E. & Lichtenstein, G. R. Human immunodeficiency virus infection and Crohn's disease: the role of the CD4 cell in inflammatory bowel disease. *J. Clin. Gastroenterol.* **25,** 456–459 (1997).
42.Astle, W. J. *et al.* The Allelic Landscape of Human Blood Cell Trait Variation and Links to Common Complex Disease. *Cell* **167,** 1415–1429.e19 (2016).
43.Benton, M. C. *et al.* Mapping eQTLs in the Norfolk Island genetic isolate identifies candidate genes for CVD risk traits. *American journal of human genetics* **93,** 1087–1099 (2013).
44.MacArthur, J. *et al.* The new NHGRI-EBI Catalog of published genome-wide association studies (GWAS Catalog). *Nucleic Acids Research* **45,** D896–D901 (2017).
45.Bielinski, S. J. *et al.* Mayo Genome Consortia: a genotype-phenotype resource for genome-wide association studies with an application to the analysis of circulating bilirubin levels. *Mayo Clin. Proc.* **86,** 606–614 (2011).
46.Dixon, J. R. *et al.* Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature Publishing Group* **485,** 376–380 (2012).
47.Wang, Y. *et al.* The 3D Genome Browser: a web-based browser for visualizing 3D genome organization and long-range chromatin interactions. *bioRxiv* 112268 (2017). doi:10.1101/112268
48.Tukey, R. H. & Strassburg, C. P. Human UDP-glucuronosyltransferases: metabolism, expression, and disease. *Annu. Rev. Pharmacol. Toxicol.* **40,** 581–616 (2000).
49.Barbeira, A. N. *et al.* Integrating Predicted Transcriptome From Multiple Tissues Improves Association Detection. *bioRxiv* 292649 (2018). doi:10.1101/292649
50.Jain, D. *et al.* Genome-wide association of white blood cell counts in Hispanic/Latino Americans: the Hispanic Community Health Study/Study of Latinos. Hum. Mol. Genet. 26, 1193–1204 (2017).

Precision drug repurposing via convergent eQTL-based molecules and pathway targeting independent disease-associated polymorphisms*

Francesca Vitali[†,1,2], Joanne Berghout[†,1-3], Jungwei Fan[†,1,2], Jianrong Li[1,2], Qike Li[1], Haiquan Li[*,1,2,4] and Yves A. Lussier[*1-7]

[1]Center for Biomedical Informatics and Biostatistics (CB2), [2]Department of Medicine COM-T,
[3]The Center for Applied Genetics and Genomics in Medicine, [4]Department of Biosystems Engineering
[5]BIO5 Institute, [6]UA Cancer Center, [7]UA Health Science (UAHS),
The University of Arizona, Tucson, AZ 85721, USA
Emails: francescavitali@email.arizona.edu, jberghout@email.arizona.edu, fanj@email.arizona.edu, qikeli@email.arizona.edu, jianrong@email.arizona.edu, haiquan@email.arizona.edu, yves@email.arizona.edu

Repurposing existing drugs for new therapeutic indications can improve success rates and streamline development. Use of large-scale biomedical data repositories, including eQTL regulatory relationships and genome-wide disease risk associations, offers opportunities to propose novel indications for drugs targeting common or convergent molecular candidates associated to two or more diseases. This proposed novel computational approach scales across 262 complex diseases, building a multi-partite hierarchical network integrating (i) GWAS-derived SNP-to-disease associations, (ii) eQTL-derived SNP-to-eGene associations incorporating both *cis*- and *trans*-relationships from 19 tissues, (iii) protein target-to-drug, and (iv) drug-to-disease indications with (iv) Gene Ontology-based information theoretic semantic (ITS) similarity calculated between protein target functions. Our hypothesis is that if two diseases are associated to a common or functionally similar eGene - and a drug targeting that eGene/protein in one disease exists - the second disease becomes a potential repurposing indication. To explore this, all possible pairs of independently segregating GWAS-derived SNPs were generated, and a statistical network of similarity within each SNP-SNP pair was calculated according to scale-free overrepresentation of convergent biological processes activity in regulated eGenes ($ITS_{eGENE\text{-}eGENE}$) and scale-free overrepresentation of common eGene targets between the two SNPs ($ITS_{SNP\text{-}SNP}$). Significance of $ITS_{SNP\text{-}SNP}$ was conservatively estimated using empirical scale-free permutation resampling keeping the node-degree constant for each molecule in each permutation. We identified 26 new drug repurposing indication candidates spanning 89 GWAS diseases, including a potential repurposing of the calcium-channel blocker Verapamil from coronary disease to gout. Predictions from our approach are compared to known drug indications using DrugBank as a gold standard (odds ratio=13.1, p-value=2.49×10^{-8}). Because of specific disease-SNPs associations to candidate drug targets, the proposed method provides evidence for future precision drug repositioning to a patient's specific polymorphisms.

Keywords: Drug repurposing; network analysis; drug repositioning; translational bioinformatics
Supplementary material: http://lussiergroup.org/publications/drug_repurposing_by_eQTL

* This work was supported in part by The University of Arizona Health Sciences CB2, the BIO5 Institute, The UA Cancer Center, and NIH (U01AI122275)
† Authors contributed equally to this work
* Corresponding authors contributed equally to this work

1. Introduction

Drug repurposing is an approach that investigates an approved drug for its potential efficacy as a treatment for other diseases[1]. This strategy can be cheaper, faster, and more efficient than *de novo* drug discovery since many preclinical and safety studies have already been completed[2, 3].

Some reported repurposing successes have relied on serendipitous clinical observation (i.e., Sildenafil/Viagra repurposed from pulmonary arterial hypertension to erectile dysfunction)[4] while many others use disease-specific basic biology hypotheses where a single molecular factor has been independently associated with pathology in two or more diseases (i.e., FYN in solid tumor proliferation and Alzheimer's)[3]. Employing scalable computational methods offers great potential for finding credible, novel, and hypothesis-free repurposing opportunities[2, 5] by rapidly linking genetic risk factors and/or molecules perturbed during disease processes with known drug targets or other identified consequences of therapy[2, 5-7]. Several computational network analysis methods have been developed for drug repurposing, generally beginning from a seed set of well-described proteins or druggable targets. These then incorporate data from protein-protein and/or protein-drug biochemistry to propose new functional candidate molecules and drug activity based on presumptive physical interactions[8, 9]. Other methods examine gene expression changes to predict signature similarity between two diseases or between a disease and a drug exposure as a way to propose candidates[10, 11]. However, these methods are limited due to (i) typically relying on single-scale methodologies and (ii) focusing on coding DNA or their gene products. High-level integration of different data sources and knowledge are required to efficiently perform multiscale analysis for a more thorough approach to hypothesis-free drug repurposing, as well as integration of signals from noncoding areas of the genome.

Genome-wide association studies (**GWAS**) represent a large potential source of information on genetic factors associated with disease risk or severity. However, about 50% of associations detected by GWAS have mapped to intergenic or noncoding sequences, suggesting altered regulatory capacity that remains difficult to interpret[12]. Fortunately, massive amounts of new data have been generated to address questions of noncoding function. These include the Genotype-Tissue Expression (GTEx) resource which mapped expression quantitative trait loci (eQTL) linking single nucleotide polymorphisms (**SNP**) to tissue-specific regulation of gene transcripts (**eGenes**)[13]. Colocalization of GWAS positional loci with these data[14, 15] and/or with additional computational integration of data in other knowledge bases (e.g., protein-protein interaction networks, Gene Ontology (**GO**)[16] annotations) shows that GWAS loci are enriched in putatively functional regions[13, 14]. In addition, non-scalable and rate-limited studies have led to the discovery and characterization of several new disease-gene and disease-biological pathway mechanistic candidates[17-21].

Motivation. We have previously designed a multiscale network approach where SNPs from GWAS are connected to gene products and their annotations via eQTL[22]. In that study, we demonstrated that pairs of independently segregating GWAS SNPs associated to the same disease were significantly more likely to be involved in similar biological processes, colocalized with binding sites for the same transcription factor(s), and involved in chromatin interactions with each other when compared to pairs of SNPs where each SNP mapped to a different disease[22]. This is consistent with the prevailing idea that heterogeneous risk factors for a given complex disease will

display some form of coalescent properties and/or converge into a few non-random, key pathways involved in driving pathology, at least in many cases[23, 24].

In this study, we ***hypothesized*** that the downstream convergence of eQTL signals between highly similar SNP-SNP pairs can be leveraged to identify druggable molecular targets relevant to two diseases. Therapeutic modulation of that factor or the pathways it is involved with present a potential opportunity for drug repurposing. We computed similarity scores between risk factors (here, SNP-SNP pairs) based on information theoretic semantic (**ITS**) similarity of their associated gene ontology biological process terms ($\boldsymbol{ITS_{GENE\text{-}GENE}}$) and overrepresentation of shared or similar eGenes ($\boldsymbol{ITS_{SNP\text{-}SNP}}$). These data were integrated with drug targeting data[25, 26]. We further demonstrate that a scale-free resampling analysis of the resulting multiscale network discovers and prioritizes a significant number of known drug-to-indication relationships from our gold standard, i.e., known treatments for the network diseases. We also report a repurposing example with literature evidence confirming the plausibility of our findings. The drug repurposing approach we developed is different from the standard approaches (for a review refer to[5]) since, to our knowledge, no method has been yet published that integrates GWAS studies with eQTL associations as pairs, with gene ontology similarities leveraged to repurpose drugs across diseases incorporating both identical and similar pathological effectors and mechanisms.

2. Methods

2.1. *Datasets*

GWAS SNP-to-disease associations were obtained from the NHGRI-EBI GWAS Catalog[27] (11/20/2017) comprising 53,009 associations between 2,373 diseases/traits and 41,973 lead SNPs.

SNP-to-eGene associations. A comprehensive secondary *cis*- and *trans*-eQTL analysis by Fagny et al[19] of the original raw data in the Genotype-Tissue Expression dataset[28] (GTEx vers. 6.0) was used for linking SNPs to eGenes (http://networkmedicine.org:3838/eqtl/ ; 19 tissues). Fagny et al[19] adjusted p-values for multiple testing using Benjamini-Hochberg correction for *cis*- and *trans*-eQTL separately, and suggest retaining associations with False Discovery Rate (FDR)< 0.2. Sample genotypes were imputed by GTEx[29], providing comprehensive overlap with the GWAS SNP set. The entire dataset included 5,896,354 associations between 1,114,453 SNPs and 21,971 eGenes.

Molecular drug-to-indication and target-to-drug and associations were downloaded from DrugBank API Portal (v1, 02/01/2018)[25] and DrugBank (01/11/2017)[26] respectively. The database consisted of 4,943 associations linking 1,133 drugs with 2,622 unstructured indications (i.e., diseases), as well as 11,978 associations linking 2,515 molecular targets with 5,623 drugs.

Gene Ontology (GO)[30] (06/28/2016) provided 29,690 GO IDs in Biological Processes (GO-BP) and 120,779 associations involving 16,604 genes and 11,052 GO-BP IDs.

2.2. *Building the drug repurposing network*

Briefly, we constructed an integrated multiscale biomolecular network connecting (i) diseases to (ii) SNPs to (iii) eGenes (eQTL transcripts) and cognate proteins intersected with both (iv-a) GO biological processes annotations (GO-BP) and (iv-b) drugs acting on the protein molecular targets

(**Fig. 1**). This network thus links each SNP to a set of eGenes and GO-BP terms. All possible SNP-SNP pairs were created, filtered to remove those marking the same linkage locus, and SNP-SNP similarity was computed based on information theoretic semantic similarity of each eGene pair's GO-BP terms ($ITS_{eGENE\text{-}eGENE}$) and overrepresentation of the SNP-pair's shared or similar eGenes ($ITS_{SNP\text{-}SNP}$). Statistically prioritized SNP pairs within a disease were used for method and target validation (**Fig. 1D**). SNP pairs that spanned two diseases yet still showed an overrepresentation of shared and/or highly similar molecular downstream eGenes were suggested as repurposing candidates (**Fig. 1C** and **4B**).

Preprocessing the data was necessary for the integration of each element in the drug repurposing network. First, disease terms used by the GWAS Catalog and DrugBank required standardization into a formal representation (**Methods 2.2.1**), as well as an automated approach for match identical or highly similar diseases between these datasets (**Methods 2.2.2**). Next, we developed a method to establish the convergent biomolecular processes revealed by within-disease GWAS risk SNP-SNP pairs and compute similarity of these processes across diseases. We propose a nested information theoretic distance that considers the functional similarities between downstream eGenes of SNP pairs for prioritization of SNP pairs (**Methods 2.2.3-5**). Once the statistically significant eGene and SNP pairs are identified (FDR<0.05), we construct the biomolecular layer (**Methods 2.2.6)** and integrate this with the drug information (**Methods 2.2.7**) to create the **Drug Repurposing Network**.

2.2.1. Formal representation of disease terms (NHGRI GWAS and DrugBank). Multiple GWAS disease traits collected from the NHGRI GWAS Catalog were grouped into semantic disease bundles, each assigned to a SNOMED-Clinical Terms (**CT**) concept representation[31]. The GWAS curator-assigned Experimental Factor Ontology (EFO)[27] was used to filter out non-disease phenotypes (e.g., pharmacogenomics responses, etc.) by retaining those under the branch EFO0000408: disease, reducing the 2,373 GWAS traits to 533 diseases. Text mining scripts and cross-mapping were used to link SNOMED-CT concepts to the EFO diseases, which were checked and curated into 262 bundles and coded to SNOMED-CT IDs. These bundles are referred to as "**GWAS diseases**" hereafter. We similarly coded 1,936 out of the 2,622 unstructured text disease terms of "**DrugBank indications**" to 2,054 distinct SNOMED IDs (**Fig.1C**). Note that one DrugBank indication can map to multiple SNOMED IDs.

2.2.2. Disease similarity computation. SNOMED-CT ontology was chosen because of its rich hierarchical relationships and high clinical coverage relevant to GWAS diseases and DrugBank indications. Disease-disease semantic similarity was determined by applying Lin's information-theoretic similarity (**ITS)** metric[32] with Sánchez et al.'s information content (**IC**) estimation[33] (**Eq.1**). By integrating these, ITS between diseases d_1 and d_2 ($\boldsymbol{ITS_{DISEASE\text{-}DISEASE}}$) can be calculated through **Eq. 2**, based on the hierarchical structure of the SNOMED-CT ontology. ITS similarity scores range from 0 to 1, where 1 corresponds to identity and 0 to complete dissimilarity. Two disease concepts with ITS>0.7 were considered similar. Using SNOMED, similarity is computed between every disease pair within the GWAS disease list as well as across the GWAS disease list and the DrugBank indication list (**Eq. 2**). Of note, drug repurposing is predicted between independent GWAS disease(s)-associated SNPs with non-trivial convergent eQTL mechanisms (**Sections 2.2.3-5**), in

which one of these GWAS diseases is similar or identical to a DrugBank indication ($\boldsymbol{ITS_{GWAS\_Disease-DrugBank\_Indication}(d_1,d_2)}$>0.7, applied **Eq.2**; **Methods 2.2.7**).

$$IC(c) = -log\left(\frac{\frac{|leaves(c)|}{|subsumers(c)|}+1}{max\_leaves+1}\right) \quad (1) \qquad ITS_{DISEASE-DISEASE}(d_1,d_2) = \frac{2 \times IC(lca(d_1,d_2))}{IC(d_1)+IC(d_2)} \quad (2)$$

where $|leaves(c)|$ is the number of leaf nodes under the concept c, $|subsumers(c)|$ is the number of ancestor nodes above the concept, max_leaves is the total number of leaves covered by the root node, d is a disease, and lca is the least common ancestor to d_1 and d_2.

2.2.3. Information theoretic similarity between two eGenes using GO Biological Processes. We also applied the information-theoretic approach that we previously published[34] to calculate functional similarity between any pair of eGenes (**Fig.1A**), i.e., $\boldsymbol{ITS_{eGENE-eGENE}}$. In GO, each gene product (g_x), used here as the canonical cognate protein of an eGene transcript, can be annotated to a set of GO terms (T), denoted as $T(g_x)$. The similarity between *eGene 1* (g_1) and *eGene 2* (g_2) is calculated by semantic similarity between $T(g_1)$ and $T(g_2)$. For each GO-BP term (t_i) associated to g_1, the similarity score $\boldsymbol{ITS_{GO-GO}(t_i, t_j)}$ is then calculated for every GO term (t_j) associated to g_2 ($t_i \in T(g_2)$) (**Fig.1A**) and use the maximum value among them (max); and vice-versa for g_2. The similarity between two genes g_1 and g_2 is thus calculated as the average of all these maximum scores (**Eq.3):**

$$ITS_{eGENE-eGENE}(g_1,g_2) = \frac{\sum_{t_i \in T(g_1)} \underset{t_j \in T(g_2)}{max}\left(ITS_{GO-GO}(t_i,t_j)\right) + \sum_{t_j \in T(g_2)} \underset{t_i \in T(g_1)}{max}\left(ITS_{GO-GO}(t_i,t_j)\right)}{|T(g_1)|+|T(g_2)|} \quad (3)$$

where $|T(g_1)|$ is the cardinality of the set $T(g_1)$. The $ITS_{eGENE-eGENE}$ output has a range between 0 and 1, where 0 indicates two genes having no similar GO annotations and 1 indicates two genes having identical GO annotations.

2.2.4. Information theoretic similarity between SNPs. The ITS of a SNP-SNP pair was calculated where both are (i) associated with at least one of the 262 GWAS diseases (**Methods; 2.1.1**) and (ii) regulate at least one eGene. Our previously published calculation[22] of similarity between a pair of SNPs ($\boldsymbol{ITS_{SNP-SNP}}$) is an extension of the $ITS_{eGENE-eGENE}$. Since every SNP can be associated with multiple eGenes and every eGene can be associated with multiple GO terms, the $ITS_{SNP-SNP}$ is a nested calculation that leverages the $ITS_{eGENE-eGENE}$ scores. It is based on the average similarity of the set of genes associated by eQTL with the two SNPs, as shown in **Eq.4** below:

$$ITS_{SNP-SNP}(s_1,s_2) = \frac{\sum_{g_i \in G(s_1)} \underset{g_j \in G(s_2)}{max}\left(ITS_{eGENE-eGENE}(g_i,g_j)\right) + \sum_{g_j \in G(s_2)} \underset{g_i \in G(s_1)}{max}\left(ITS_{eGENE-eGENE}(g_i,g_j)\right)}{|G(s_1)|+|G(s_2)|} \quad (4)$$

where SNP s_1 was associated with a set of genes $G(s_1)$, and $|G(s_1)|$ is the cardinality of the set $G(s_1)$, similarly for s_2. The $ITS_{eGENE-eGENE}$ is the similarity of two genes computed with **Eq.3**. Likewise, the $ITS_{SNP-SNP}$ has a score ranging from 0 to 1; a value of 1 indicates two SNPs of perfect similarity, and 0 refers to two SNPs of null functional similarity.

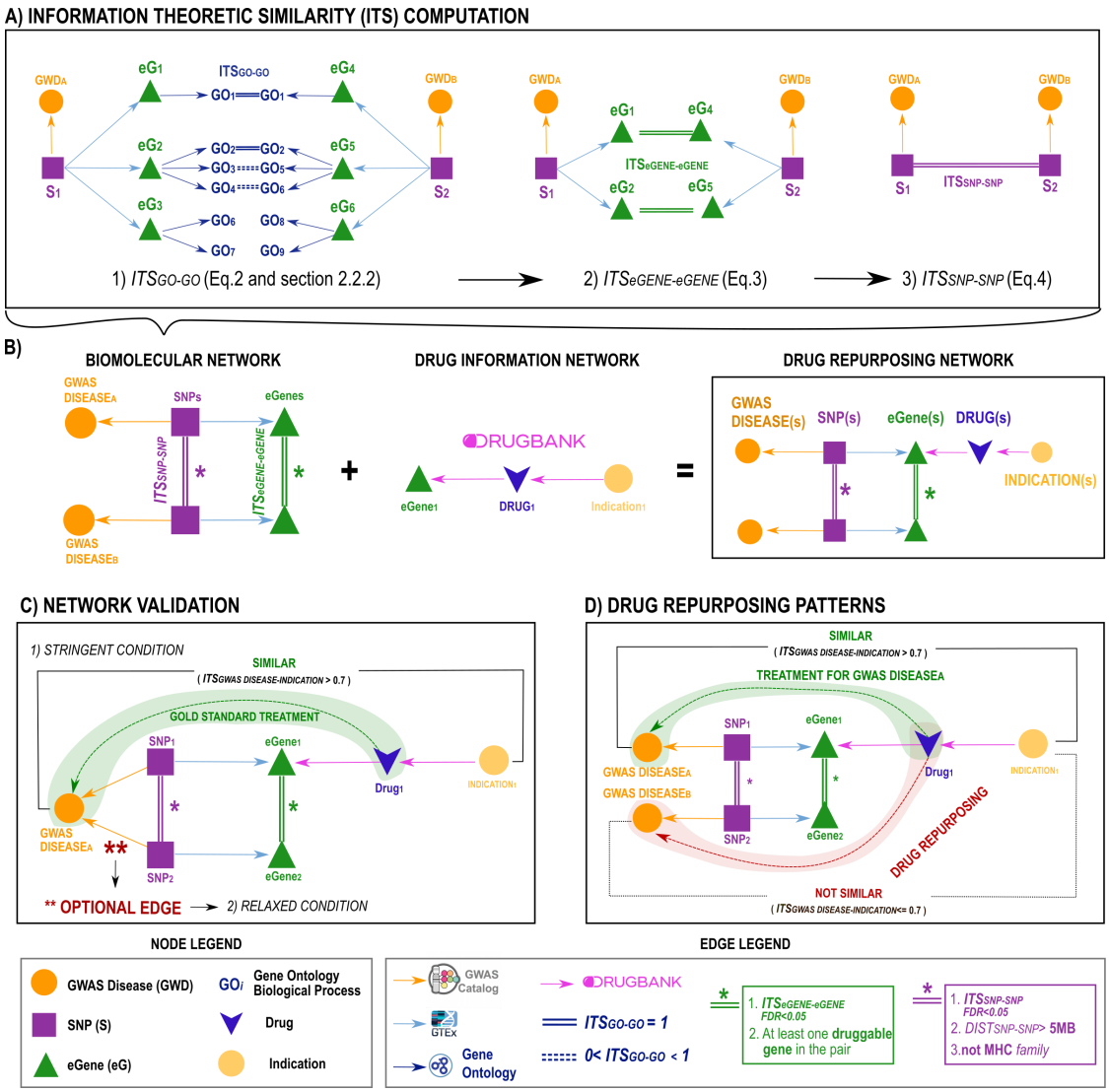


Fig. 1. Overview of the construction, computational prioritization, and validation of the drug repurposing network. A) *ITS computation*. We applied ITS to compute the similarity between GO-BPs, SNPs, and genes through a cascade process as described in **Methods 2.2.3-5.** This began construction of the biomolecular network layer. **B)** *Integration of multiscale biomolecular associations* using GWAS diseases, SNPs, and eGenes as nodes. The associations (edges) between nodes were obtained by extracting GWAS disease-to-SNP, and SNP-to-eGene (SNP-eG) relationships from the database resources described (**Methods 2.1**). The biomolecular network was then filtered to remove SNP-SNP pairs not meeting the introduced criteria (Edge Legend, **Methods 2.2.6.**). $ITS_{SNP\text{-}SNP}$ is computed as in **Eq.4** considering all the eGenes extracted from eQTL data and the network was further refined to include only significantly similar eGene-eGene pairs, i.e. $ITS_{SNP\text{-}SNP}$ and $ITS_{eGENE\text{-}eGENE}$ **(Eq.3)** False Discovery Rate (FDR) <0.05. Drug-eGene and Drug-indication associations extracted from Drugbank (drug information layer) are included to obtain the final drug-repurposing network. **C)** *Network validation*. The drug repurposing network is validated by querying if the network predicted a significantly high number of gold standard treatments for GWAS diseases. Two conditions of validation are proposed, one stringent and one more relaxed (**). **D)** *Drug repurposing patterns*. We extracted GWAS disease pairs and the related convergent mechanisms where at least a gold standard treatment was predicted for one of the two GWAS diseases. The approach predicts new candidate therapies by repositioning drugs across these GWAS disease pairs.

2.2.5. Scale-Free permutation for FDR estimation of ITS. 10,000 and 100,000 conservative scale-free permutations were performed to estimate statistical significance of the $ITS_{eGENE\text{-}eGENE}$ and $ITS_{SNP\text{-}SNP}$ scores (~500,000 core hours), respectively. In each permutation, the node degree of every node in the gene-GO annotation network was preserved (each specific gene retained the node degree of GO term associations and vice-versa). Multiplicity of prioritization was controlled by Benjamini-Hochberg with a cutoff of FDR$\leqslant$0.05 (*p.adjust* for both $ITS_{eGENE\text{-}eGENE}$ and $ITS_{SNP\text{-}SNP}$).

*2.2.6. Biomolecular network layer construction (**Fig.1**).* The drug repurposing network construction starts by defining its biomolecular layer. This level associates **GWAS diseases**, **SNPs**, and **molecular targets** (**Fig.1A**). Disease-to-SNP edges were obtained from GWAS lead SNPs, and SNP-to-regulated molecular target (eGene) edges were obtained from eQTL data as described in **Methods 2.1.2**. This produced a network of 9,750 associations between 8,955 SNPs and 235 unique diseases, where each of the retained SNPs was also associated with at least one eGene via eQTL. All SNP-SNP pairs were generated and filtered to remove SNP pairs (i) separated by less than 5Mb, (ii) in linkage disequilibrium with one another (r^2>0.01) according to HapMap and 1000 Genomes CEU data, and/or (iii) SNP pairs where both mapped within the Major Histocompatibility Complex (MHC; Chr6: 28,477,797-33,448,355, ±2 Mb; GRCh37). SNP-SNP pairs where only one SNP mapped to the MHC were retained. This was done to remove SNP pairs trivially marking the same locus. Similarity is computed ($ITS_{SNP\text{-}SNP}$) for each retained SNP pair according to **Eq.4** (**Methods 2.2.4**). Focusing only on the SNP pairs that were statistically significant ($ITS_{SNP\text{-}SNP;}$ FDR<0.05), $ITS_{eGENE\text{-}eGENE}$ is computed (**Eq.3)** to further filter. SNP pairs that satisfied both $ITS_{SNP\text{-}SNP}$ and $ITS_{eGENE\text{-}eGENE}$ at FDR<0.05 were considered as having convergent biological mechanisms and used to construct the final biomolecular network.

2.2.7. Construction of the drug repurposing network. The final network construction step involves the integration of drug knowledge (**Fig.1B**) with the biomolecular level by matching protein-coding eGenes with the molecular targets of drugs acquired from DrugBank (**Methods 2.1.3**). In this step, the disease indications are included for these drugs, as they serve to validate our predictions when recapturing known indications (validation, **Methods 2.3**) and to identify novel opportunities predicted by our method that can be used for drug repurposing (**Methods 2.4**).

2.3. *Validation of the drug repurposing network*

Before analyzing potential drug repurposing candidates, we validated our drug repurposing network by determining whether known drug indications for the included GWAS diseases could be inferred from the network above the chance expectation (**Fig.1C**). To this end, a Fisher's Exact Test (FET) is performed considering: (i) all druggable molecular targets (**DMTs**) and (ii) all druggable diseases (**DD**). In this validation, a DMT was defined as any eGene that has at least one drug in DrugBank targeting the cognate protein, and that the drug is indicated for one or more of the 262 GWAS diseases defining our set (**Methods 2.2**). A DD is defined as any GWAS diseases in the network associated with at least one target eGene found in DrugBank, and therefore corresponds to all the GWAS diseases that could theoretically be validated using these databases. In this way, we can determine how many of the theoretical combinations of DMTs and DDs (DMTs*DDs) are predicted

by analysis of significant eGenes associated with prioritized SNP pairs with convergent mechanisms. The enrichment of gold standard drug indications among the predictions is conducted assuming that the GWAS disease-eGenes analysis can, in principle, discover any drug targets in DrugBank. We constructed the contingency table to perform the FET by counting the number of DMT-DD interactions (i) present/not present in Drugbank vs (ii) included/not included in our final ITS-filtered network (**Fig.1C**).

The validation procedure includes similarity between GWAS diseases and indications (**Fig.1D;** $ITS_{GWAS\_Disease-GWAS\_Disease}$, **Eq.2**; **Methods 2.2.2**). The network validation procedure is then conducted by applying two additional conditions, one stringent and one more relaxed (**Fig.1D**), using DrugBank as a gold standard. First, convergent mechanisms between two SNPs associated with the same GWAS are prioritized ($ITS_{GWAS\_Disease-GWAS\_Disease}$>0.7), i.e., similar SNP pairs ($ITS_{SNP\text{-}SNP}$ FDR<0.05) with eGene pairs ($ITS_{eGENE\text{-}eGENE}$ FDR<0.05), and the number of eGene-GWAS disease associations that were identical or similar ($ITS_{GWAS\_Disease-DrugBank\_Indication}$>0.7) to the related molecule-indication associations found in DrugBank were counted (**Fig.1D**). In the relaxed condition, the same procedure is applied, but without the constraint that both SNPs in the prioritized pair must map to the same disease (**Fig.1D**).

2.4. *Drug repurposing pattern identification*

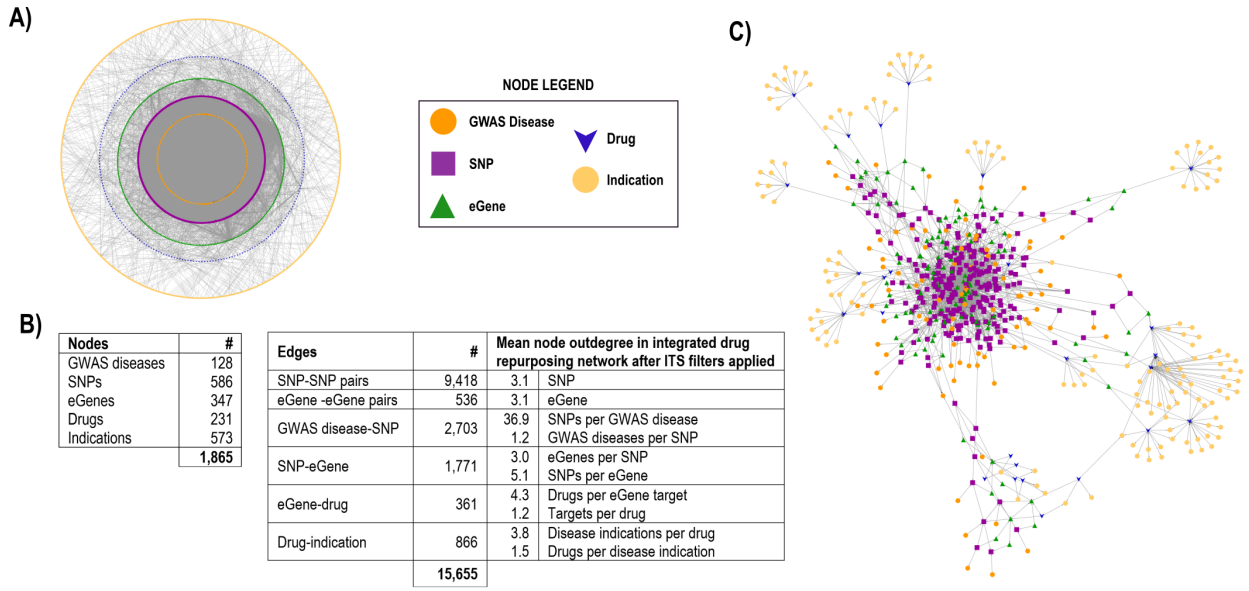
Drug repurposing candidates are identified by analyzing specific network patterns as illustrated in **Fig.1D**. We prioritized all subnetworks involving pairs of GWAS diseases related to convergent mechanism in which at least one eGene was targeted by a drug known to treat one of the two GWAS diseases or a similar ($ITS_{GWAS\_Disease-DrugBank\_Indication}$ ≤0.7) disease. Thus, if the drug is prescribed as a treatment for two diseases dissimilar in the pair ($ITS_{GWAS\_Disease-GWAS\_Disease}$>0.7), then it is predicted as a repurposing candidate across the two GWAS diseases.

3. Results and discussion

3.1. *Overall results and visualization*

The drug repurposing network (**Fig.2A**) comprises 1,865 nodes and 15,655 edges (**Fig.2B**) and was obtained after considering the similarity of 479,896 SNP-SNP pairs. 74,803 SNP pairs are prioritized with significant convergent biomolecular mechanisms ($ITS_{SNP\text{-}SNP}$ with FDR<0.05, **Eq.4**; **Methods 2.2.5**). The list of similar SNP-pairs is further constrained to those with an association to at least one disease for which an indication is known in DrugBank, resulting in 9,418 retained SNP pairs, their associated significant eGene pairs ($ITS_{eGENE\text{-}eGENE}$ with FDR<0.05, **Methods 2.2.3**), and drug information (**Methods 2.2.7**). All retained SNP pairs marked two independently segregating disease loci, based on the positional and linkage filters applied in **Methods 2.2.6**. SNP pair similarity was driven by both *cis*- and *trans*-eQTL associations, with 8,329 SNP pairs prioritized through regulation of similar eGenes found in *cis* to each SNP, and 1,089 SNP pairs prioritized based on at least one *trans*-regulated eGene by one of the SNPs (**Fig.2A**). **Fig.2B** shows details of the network nodes and edges. While they remain a minority, having 12% of prioritized SNP pairs reliant on *trans*-eQTL relationships highlights the importance of including these complex regulatory data, as

these would have been overlooked by focusing exclusively on those genes near the GWAS SNP. The subnetwork relevant for drug repurposing comprises only the SNP-pairs for which their prioritized eGenes code for the protein target of an existing drug (**Fig.2C**).



Nodes	#
GWAS diseases	128
SNPs	586
eGenes	347
Drugs	231
Indications	573
	1,865

Edges	#	Mean node outdegree in integrated drug repurposing network after ITS filters applied	
SNP-SNP pairs	9,418	3.1	SNP
eGene -eGene pairs	536	3.1	eGene
GWAS disease-SNP	2,703	36.9 1.2	SNPs per GWAS disease GWAS diseases per SNP
SNP-eGene	1,771	3.0 5.1	eGenes per SNP SNPs per eGene
eGene-drug	361	4.3 1.2	Drugs per eGene target Targets per drug
Drug-indication	866	3.8 1.5	Disease indications per drug Drugs per disease indication
	15,655		

Fig. 2. Drug Repurposing Network. A) Comprehensive biomolecular network comprising significant convergent *cis*- and *trans*-eQTL mechanisms between GWAS disease-associated SNPs ($ITS_{SNP\text{-}SNP}$ FDR<0.05; $ITS_{eGENE\text{-}eGENE}$ FDR<0.05), for which there exists indications in DrugBank (i.e., the molecular target of $Drug_i$ is the protein transcribed by at least one eGene associated by eQTL to SNP-SNP $Pair_x$; **Fig.1B**; **Methods 2.2**). **B)** Tables summarizing the number of nodes and edges of the network shown in panel A; for each edge type we also reported the mean node outdegree. **C)** Prioritized subset of the network in panel A relevant for drug repurposing because it satisfies one additional criteria: the disease indication of a $Drug_i$ is identical or similar to the GWAS disease associated to the SNP-SNP $Pair_x$ related to the eGene targeted by the $Drug_i$ (**Fig.1C; Methods 2.4;** $ITS_{GWAS\_Disease-DrugBank\_Indication}$>0.7). In **Supplementary Material -Figure S1,** we reported a high-resolution version of this network with labeled network node names.

3.2. *Network validation results*

We validated our network by calculating the enrichment of drug targets predicted by our method (**Methods 2.3**) over drug targets reported in a curated database gold standard (DrugBank). First, identical or similar disease indications matched to any of the 262 GWAS diseases are extracted, which resulted in 127 "druggable" diseases (DD) together with their 1,336 associated druggable molecular targets (DMT). This yielded 169,672 eGene-disease combinations that could potentially be predicted (DMT*DD). Assuming the stringent criterion where DrugBank's annotated drug indication must be identical or similar to the GWAS disease and both SNPs in the prioritized SNP-SNP pair must be associated to that same GWAS disease, our method predicted 56 relationships involving DMTs and GWAS diseases. DrugBank included 2,783 DMT-DD associations with 10 overlapping (Fisher's Exact Test-FET $p=2.5x10^{-8}$; odds ratio=13.1). When considering the more relaxed criterion of high similarity between gold standard diseases and predicted indications, we found 29 overlapping, from a total of 299 potential predictions (FET $p= 3.6x10^{-14}$; odds ratio= 6.5).

Fig.3A illustrates a drug target for Rheumatoid Arthritis (RA) that was predicted by eQTL similarity of two distinct GWAS SNPs[35] ($ITS_{SNP\text{-}SNP}$ FDR=0.0007) and confirmed in DrugBank as the known target of Etanercept indicated for Polyarticular Juvenile Idiopathic Arthritis (PJIA)[36, 37]. These two RA SNPs (rs72717009 and rs4239702) affect the expression of *FCGR2C* and *CD40* respectively. The gene products of *FCGR2C* and *CD40* are annotated to highly similar biological processes ($ITS_{eGENE\text{-}eGENE}$ FDR=0.01), suggesting a convergent mechanism revealed by these two independently segregating factors. Since RA and PJIA are highly similar diseases ($ITS_{RA\text{-}PJIA}$=0.78), our approach could correctly predict Etanercept as a treatment for RA[37].

3.3. *Drug repurposing results*

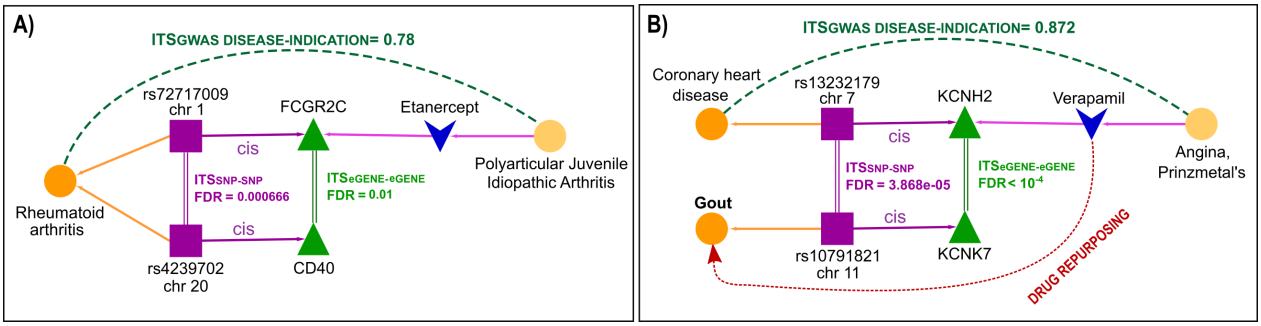


Fig. 3. Examples of prediction by eQTL signal convergence across distinct chromosomes. A) *Gold standard validation.* In the drug repurposing network, we could confirm Etanercept as standard treatment for Rheumatoid arthritis. **B)** *Drug repurposing.* Our approach was able to predict Verapamil as a new potential treatment for gout, for which a retrospective study reports lower incidence of gout.

Following the procedure in **Methods 2.4**, we extracted the GWAS diseases having convergent mechanisms ($ITS_{SNP\text{-}SNP}$ FDR<0.05 and $ITS_{GENE\text{-}GENE}$ FDR<0.05) with one of the GWAS diseases for which at least one gold standard indication was present in the network. In detail, we identified 181 distinct GWAS disease pairs involving 90 diseases. 19 of these diseases had a molecularly-targeted treatment indicated in DrugBank that matched the eGene-prioritized molecular targets (i.e., GWAS $disease_A$ shown in **Fig.1D**). 89 diseases had new drug candidates identified by our network, potentially allowing repurposing (i.e., GWAS $disease_B$ in **Fig.1D**). We extracted 1,288 patterns (**Supplementary Material -Table S1**) including 26 drug candidates relevant to at least one of the 89 GWAS diseases. The subnetwork obtained by considering the drug repurposing patterns is depicted in **Fig.2B** and comprises 628 nodes (90 GWAS diseases, 253 SNPs, 108 eGenes, 26 drugs and 151 indications) and 1,758 edges. Within the 391 SNP-SNP pairs (edges), 25 were prioritized based on at least one *trans*-eQTL association and 366 are driven exclusively by *cis*-eQTL associations. Tissue source of each eQTL association are provided in **Table S1.** As eQTL detection power varied between tissues in our input and multi-organ pathologies are common in complex diseases, we chose not to restrict our results to only those with shared or overlapping tissue sources. However, as candidates are considered more closely, these filters may allow prioritization and/or a cleaner set of hypotheses.

Fig.3B illustrates Verapamil as a candidate drug target for gout repositioned from coronary artery disease that was predicted by eQTL similarity of their respective distinct GWAS SNPs

($ITS_{SNP\text{-}SNP}$ FDR=0.000039). The proposed method predicted that KCNH2 is involved in similar biological processes as KCKN7 (FDR $ITS_{eGENE\text{-}eGENE}$ FDR<10^{-4}). Verapamil is a calcium channel blocker and inhibitor of the protein Potassium voltage-gated channel subfamily H member 2 (*KCNH2)*[38]. It is a class IV anti-arrhythmia agent currently used to treat hypertension, angina, and cluster headache. The cross-disease prioritized SNP pair indicates that variation at rs13232179 (coronary artery disease[39]) modulates expression of KCNH2 in tibial artery and that variation at rs10791821 (gout[40]) modulates expression of KCNK7 in tibial artery, transverse colon, esophagus muscularis, and thyroid. Functional similarity between KCNH2 and KCNK7 suggests that effective pathway modifying medications may play a role in both conditions. Supporting this prediction, studies have demonstrated that other calcium channel blockers are associated with a lower risk of incident gout[41].

4. Limitations and future studies

Currently, our method cannot detect if the effect of the expression from eQTL studies is concordant; and so, the proposed method may predict adverse events as well as drug repurposing opportunities. For example, Adalimumab (**Fig.2C**), currently prescribed for inflammatory bowel disease, is predicted as a possible treatment for Multiple Sclerosis (MS). However, anecdotal cases report worsening of MS patients treated with this drug[42]. Regulation of eGenes in distinct tissues may also have important biological consequences. Future studies will focus on (i) experimental validation of select candidates, (ii) to provide the data with filtering and analysis tools as an online public repository, and (iii) the integration of directional eQTL information in the presence of specific SNP variants to determine if these cases can be predicted.

5. Summary and conclusion

Drug repurposing offers novel venues to use currently available or investigational drugs. We developed a computational drug repurposing approach leveraging several data and knowledge resources, by integrating GWAS studies, eQTL data, drug information, and GO similarities in a multi-partite hierarchical network. Our approach is anchored on the identification of convergent *cis*- and *trans*- eQTL targets across distinct disease-associated polymorphisms. These repurposings are distinct from previous approaches in that we integrate convergent downstream *cis*- and *trans*-eQTL signals from any polymorphism, inclusive of intergenic regions. This automatically suggests drug repurposing through shared molecular target candidates identified across diseases, beyond the straightforward “host” or “nearest” gene overlap (e.g., protein-interaction networks). Our study demonstrates that GWAS and eQTL-derived networks can predict a significant number of gold standard indications and novel drug repurposing suggestions. Because of specific disease SNPs-associations to candidate drug targets, the proposed method provides evidence for future precision drug repositioning to a patient’s specific polymorphisms.

Acknowledgements

We thank Drs. M Fagny, JN Paulson, J Quackenbush and J Platig for providing early access to tissue specific eQTL associations.

References

1. C. R. Chong and D. J. Sullivan, Jr., *Nature*, 2007, **448**, 645-646.
2. Y. Cha, T. Erez, et al., *British journal of pharmacology*, 2018, **175**, 168-180.
3. S. M. Strittmatter, *Nat Med*, 2014, **20**, 590-591.
4. T. A. Ban, *Dialogues Clin Neurosci*, 2006, **8**, 335-344.
5. J. Li, S. Zheng, et al., *Briefings in bioinformatics*, 2016, **17**, 2-12.
6. J. Lamb, E. D. Crawford, et al., *science*, 2006, **313**, 1929-1935.
7. M. Sirota, J. T. Dudley, et al., *Science translational medicine*, 2011, **3**, 96ra77.
8. F. Cheng, R. J. Desai, et al., *Nat Commun*, 2018, **9**, 2691.
9. Y. Luo, X. Zhao, et al., *Nat Commun*, 2017, **8**, 573.
10. X. He, C. K. Fuller, et al., *Am J Hum Genet*, 2013, **92**, 667-680.
11. S. M. Corsello, J. A. Bittker, et al., *Nat Med*, 2017, **23**, 405-408.
12. P. M. Visscher, N. R. Wray, et al., *AJHG*, 2017, **101**, 5-22.
13. G. Consortium, *Science*, 2015, **348**, 648-660.
14. G. T. Consortium, D. A. Laboratory, et al., *Nature*, 2017, **550**, 204-213.
15. M. A. Schaub, A. P. Boyle, et al., *Genome research*, 2012, **22**, 1748-1759.
16. M. Ashburner, C. A. Ball, et al., *Nature genetics*, 2000, **25**, 25-29.
17. Y. Lee, H. Li, et al., *J Am Med Inform Assoc*, 2013, **20**, 619-629.
18. Z. Yue, I. Arora, et al., *BMC bioinformatics*, 2017, **18**, 532.
19. M. Fagny, J. N. Paulson, et al., *PNAS USA*, 2017, **114**, E7841-e7850.
20. J. Zhang, K. Jiang, et al., *PloS one*, 2015, **10**, e0116477.
21. P. Sanseau, P. Agarwal, et al., *Nature biotechnology*, 2012, **30**, 317.
22. H. Li, I. Achour, et al., *NPJ Genom Med*, 2016, **1**.
23. A. Califano, A. J. Butte, et al., *Nat Genet*, 2012, **44**, 841-847.
24. E. A. Boyle, Y. I. Li and J. K. Pritchard, *Cell*, 2017, **169**, 1177-1186.
25. J. Mullen, S. J. Cockell, et al., *PloS one*, 2016, **11**, e0155811.
26. V. Law, C. Knox, et al., *Nucleic Acids Res*, 2014, **42**, D1091-1097.
27. J. MacArthur, E. Bowler, et al., *Nucleic acids research*, 2017, **45**, D896-D901.
28. G. T. Consortium, *Science*, 2015, **348**, 648-660.
29. A. Battle, C. D. Brown, et al., *Nature*, 2017, **550**, 204-213.
30. C. Gene Ontology, *Nucleic Acids Res*, 2015, **43**, D1049-1056.
31. Y. A. Lussier, D. J. Rothwell and R. A. Cote, *Methods Inf. Med.*, 1998, **37**, 161-164.
32. D. Lin, *Icml*, 1998, **98**, 296-304.
33. D. Sánchez, M. Batet and D. Isern, *Knowledge-Based Systems*, 2011, **24**, 297-303.
34. Y. Tao, J. Li, et al., *Bioinformatics (Oxford, England)*, 2007, **23**, i529-i538.
35. Y. Okada, D. Wu, et al., *Nature*, 2014, **506**, 376-381.
36. D. J. Lovell, E. H. Giannini, et al., *NEJM*, 2000, **342**, 763-769.
37. L. W. Moreland, M. H. Schiff, et al., *Annals of internal medicine*, 1999, **130**, 478-486.
38. P. Tfelt-Hansen and J. Tfelt-Hansen, *Headache*, 2009, **49**, 117-125.
39. G. Lettre, C. D. Palmer, et al., *PLoS genetics*, 2011, **7**, e1001300.
40. H. Matsuo, K. Yamamoto, et al., *Annals of the rheumatic diseases*, 2016, **75**, 652-659.
41. H. K. Choi, L. C. Soriano, et al., *Bmj*, 2012, **344**, d8190.
42. T. Matsumoto, I. Nakamura, et al., *Clinical rheumatology*, 2013, **32**, 271-275.

An Optimal Policy for Patient Laboratory Tests in Intensive Care Units

Li-Fang Cheng*[1], Niranjani Prasad*[2] and Barbara E. Engelhardt[2,3]
[1]*Department of Electrical Engineering, Princeton University*
[2]*Department of Computer Science, Princeton University*
[3]*Center for Statistics and Machine Learning, Princeton University*

Laboratory testing is an integral tool in the management of patient care in hospitals, particularly in intensive care units (ICUs). There exists an inherent trade-off in the selection and timing of lab tests between considerations of the expected utility in clinical decision-making of a given test at a specific time, and the associated cost or risk it poses to the patient. In this work, we introduce a framework that learns policies for ordering lab tests which optimizes for this trade-off. Our approach uses batch off-policy reinforcement learning with a composite reward function based on clinical imperatives, applied to data that include examples of clinicians ordering labs for patients. To this end, we develop and extend principles of Pareto optimality to improve the selection of actions based on multiple reward function components while respecting typical procedural considerations and prioritization of clinical goals in the ICU. Our experiments show that we can estimate a policy that reduces the frequency of lab tests and optimizes timing to minimize information redundancy. We also find that the estimated policies typically suggest ordering lab tests well ahead of critical onsets—such as mechanical ventilation or dialysis—that depend on the lab results. We evaluate our approach by quantifying how these policies may initiate earlier onset of treatment.

Keywords: Reinforcement Learning, Dynamic Treatment Regimes, Pareto Learning

1. Introduction

Precise, targeted patient monitoring is central to improving treatment in an ICU, allowing clinicians to detect changes in patient state and to intervene promptly and only when necessary. While basic physiological parameters that can be monitored bedside (e.g., heart rate) are recorded continually, those that require invasive or expensive laboratory tests (e.g., white blood cell counts) are more intermittently sampled. These lab tests are estimated to influence up to 70% percent of diagnoses or treatment decisions, and are often cited as the motivation for more costly downstream care [1, 2]. Recent medical reviews raise several concerns about the over-ordering of lab tests in the ICU [3]. Redundant testing can occur when labs are ordered by multiple clinicians treating the same patient or when recurring orders are placed without reassessment of clinical necessity. Many of these orders occur at time intervals that are unlikely to include a clinically relevant change or when large panel testing is repeated to detect a change in a small subset of analyses [4]. This leads to inflation in costs of care and in the likelihood of false positives in diagnostics, and also causes unnecessary discomfort to the patient.

*These authors contributed equally to this work.

Moreover, excessive phlebotomies (blood tests) can contribute to risk of hospital-acquired anaemia; around 95% of patients in the ICU have below normal haemoglobin levels by day 3 of admission and are in need of blood transfusions. It has been shown that phlebotomy accounts for almost half the variation in the amount of blood transfused [5].

With the disproportionate rise in lab costs relative to medical activity in recent years, there is a pressing need for a sustainable approach to test ordering. A variety of approaches have been considered to this end, including restrictions on the minimum time interval between tests or the total number of tests ordered per week. More data-driven approaches include an information theoretic framework to analyze the amount of novel information in each ICU lab test by computing conditional entropy and quantifying the decrease in novel information of a test over the first three days of an admission [6].

In a similar vein, a binary classifier was trained using fuzzy modeling to determine whether or not a given lab test contributes to information gain in the clinical management of patients with gastrointestinal bleeding [7]. An "informative" lab test is one in which there is significant change in the value of the tested parameter, or where values were beyond certain clinically defined thresholds; the results suggest a 50% reduction in lab tests compared with observed behaviour. More recent work looked at predicting the results of ferratin testing for iron deficiency from information in other labs performed concurrently [8]. The predictability of the measurement is inversely proportional to the novel information in the test. These past approaches underscore the high levels of redundancy that arise from current practice. However, there are many key clinical factors that have not been previously accounted for, such as the low-cost predictive information available from vital signs, causal connection of clinical interventions with test results, and the relative costs associated with ordering tests.

In this work, we introduce a reinforcement learning (RL) based method to tackle the problem of developing a policy to perform actionable lab testing in ICU patients. Our approach is two-fold: first, we build an interpretable model to forecast future patient states based on past observations, including uncertainty quantification. We adapt multi-output Gaussian processes (MOGPs; [9, 10]) to learn the patient state transition dynamics from a patient cohort including sparse and irregularly sampled medical time series data, and to predict future states of a given patient trajectory. Second, we model patient trajectories as a Markov decision process (MDP). This framework has been applied to the recommendation of treatment strategies for critical care patients in a variety of different settings, from recommending drug dosages to efficiently weaning patients from mechanical ventilation [11–13]. We design the state and reward functions of the MDP to incorporate relevant clinical information, such as the expected information gain, administered interventions, and costs of actions (here, ordering a lab test). A major challenge is designing a reward function that can trade off multiple, often opposing, objectives. There has been initial work on extending the MDP framework to composite reward functions [14]. Specifically, fitted Q-iteration (FQI) has been used to learn policies for multi-objective MDPs with vector-valued rewards, for the sequence of interventions in two-stage clinical antipsychotic trials [15]. A variation of Pareto domination was then used to generate a partial ordering of policies and extract all policies that are optimal for some scalarization function, leaving the choice of parameters of the scalarization function to decision makers.

Here, we look to translate these principles to the problem of lab test ordering. Specifically, we focus on blood tests relevant in the diagnosis of sepsis or acute renal failure, two common conditions associated with high mortality risk in the ICU: white blood cell count (WBC), blood lactate level, serum creatinine, and blood urea nitrogen (BUN). We present our methods within a flexible framework that can in principle be adapted to a patient cohort with different diagnoses or treatment objectives, influenced by a distinct set of lab results. Our proposed framework integrates prior work on off-policy RL and Pareto learning with practical clinical constraints to yield policies that are close to intuition demonstrated in historical data. We apply our framework to a publicly available database of ICU admissions, evaluating the estimated policy against the policy followed by clinicians using both importance sampling based estimators for off-policy policy evaluation and by comparing against multiple clinically inspired objectives, including onset of clinical treatment that was motivated by the lab results.

2. Methods

2.1. *Cohort selection and preprocessing*

We extract our cohort of interest from the MIMIC III database [16], which includes de-identified critical care data from over 58,000 hospital admissions. From this database, we first select adult patients with at least one recorded measure for each of 20 vital signs and lab tests commonly ordered and reviewed by clinicians (for instance, results reported in a complete blood count or basic metabolic panel). We further filter patients by their length-of-stay, keeping only those in the ICU for between one and twenty days, to obtain a final set of 6,060 patients (Table 1).

Table 1. **Total recordings, mean & standard deviation (SD)** for covariates in selected cohort.

Covariate	Count	Mean	SD
Respiratory Rate (RR)	1,046,364	20.1	5.7
Heart Rate (HR)	964,804	87.5	18.2
Mean Blood Pressure (Mean BP)	969,062	77.9	15.3
Temperature, °F	209,499	98.5	1.4
Creatinine	67,565	1.5	1.2
Blood Urea Nitrogen (BUN)	66,746	31.0	21.1
White Blood Cell Count (WBC)	59,777	11.6	6.2
Lactate	39,667	2.4	1.8

Included in the 20 physiological traits we filter for are eight that are particularly predictive of the onset of severe sepsis, septic shock, or acute kidney failure. These traits are included in the SIRS (System Inflammatory Response Syndrome) and SOFA (Sequential Organ Failure Assessment) criteria [17]. The average daily measurements or lab test orders across the chosen cohort for these eight traits is highly variable (Figure 1). Of these eight traits, the first three are vitals measured using bedside monitoring systems for which approximately hourly measurements are recorded; the latter four are labs requiring phlebotomy and are typically measured just 2–3 times each day. We find the frequency of orders also varies across different labs, possibly due in part to differences in cost; for example, WBC (which is relatively inexpensive to test) is on average sampled slightly more often than lactate. In order to apply our proposed RL

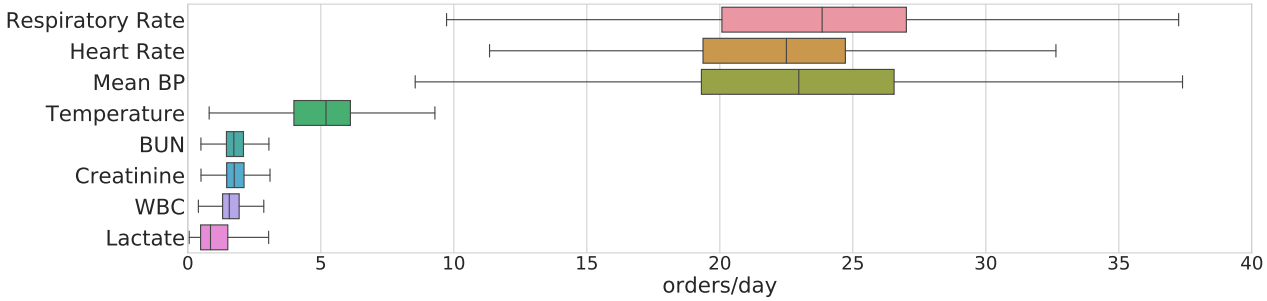


Fig. 1. **Mean number of recorded measurements per day, of chosen vitals and lab tests.** These eight traits are commonly used in computing clinical risk scores or diagnosing sepsis.

algorithm to this sparse, irregularly sampled dataset, we adapt the multi-output Gaussian process (MOGP) framework [10] to obtain hourly predictions of patient state with uncertainty quantified, on 17 of the 20 clinical traits. For three of the vitals, namely the components of the Glasgow Coma Scale, we impute with the last recorded measurement.

2.2. *MDP formulation*

Each patient admission is modelled as an MDP with:

(1) a state space $\mathcal{S}$, such that the patient physiological state at time t is given by $s_t \in \mathcal{S}$;
(2) an action space $\mathcal{A}$ from which the clinician's action a_t is chosen;
(3) an unknown transition function $\mathcal{P}_{sa}$ that determines the patient dynamics; and
(4) a reward function r_t that constitutes the observed clinical feedback for this action.

The objective of the RL agent is to learn an optimal policy $\pi^* : \mathcal{S} \rightarrow \mathcal{A}$ that maximizes the expected discounted accumulated reward over the course of an admission:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_t | \pi\right], \text{ where } T \text{ is admission length, } \gamma \text{ is the discount factor.}$$

We start by describing the state space of our MDP for ordering lab tests. We first resample the raw time series using a multi-objective Gaussian process with a sampling period of one hour. The patient state at time t is defined by:

$$\boldsymbol{s_t} = \left[m_t^{SOFA}\,,\, \boldsymbol{m}_t^{vitals}\,,\, \boldsymbol{m}_t^{labs}\,,\, \boldsymbol{y}_t^{labs}\,,\, \boldsymbol{\Delta}_t^{labs}\right]^\top \tag{1}$$

Here, $\boldsymbol{m}_t$ denotes the predictive means and standard deviations respectively of each of the vitals and lab tests. For the predictive SOFA score m_t^{SOFA}, we compute the value using its clinical definition, from the predictive means on five traits—mean BP, bilirubin, platelet, creatinine, FiO_2—along with GCS and related medication history (e.g., dopamine). Vitals include any time-varying physiological traits that we consider when determining whether to order a lab test. Here, we look at four key physiological traits—heart rate, respiratory rate, temperature, and mean blood pressure—and four lab tests—creatinine, BUN, WBC, and lactate. The values $\boldsymbol{y}_t$ are the last known measurements of each of the four labs, and $\boldsymbol{\Delta}_t$ denotes the elapsed time since each was last ordered. This formulation results in a 21-dimensional state space. Depending on the labs that we wish to learn recommendations for testing, the action space $\mathcal{A}$

is a set of binary vectors whose 0/1 elements indicate whether or not to place an order for a specific lab. These actions can be written as $\boldsymbol{a_t} \in \mathcal{A} = \{1, 0\}^L$, where L is the number of labs. In our experiments, we learn policies for each of the four labs independently, such that $L = 1$, but this framework could be easily extended to jointly learning recommendations for multiple labs.

In order for our RL agent to learn a meaningful policy, we need to design a reward function that provides positive feedback for the ordering of tests where necessary, while penalizing the over- or under-ordering of any given lab test. In particular, the agent should be encouraged to order labs when the physiological state of the patient is abnormal with high probability, based on estimates from the MOGP, or when a lab is predicted to be informative (in that the forecasted value is significantly different from the last known measurement) due to a sudden change in disease state. In addition, the agent should incur some penalty whenever a lab test is taken, decaying with elapsed time since the last measurement, to reflect the effective cost (both economic and in terms of discomfort to the patient) of the test. We formulate these ideas into a vector-valued reward function $\boldsymbol{r_t} \in \mathbb{R}^d$ of the state and action at time t, as follows:

$$\boldsymbol{r_t} = \left[{r_t}^{SOFA} \, , {r_t}^{treat} \, , {r_t}^{info} \, , -{r_t}^{cost} \right]^\top \tag{2}$$

Patient state: The first element, r^{SOFA}, uses the recently introduced SOFA score for sepsis [18] which assesses severity of organ dysfunction in a potentially septic patient. Our use of SOFA is motivated by the fact that, in practice, sepsis is more often recognized from the associated organ failure than from direct detection of the infection itself [19]. The raw SOFA score ranges from 0 to 24, with a maximum of four points assigned each to symptom of failure in the respiratory system, nervous system, liver, kidneys, and blood coagulation. A change in SOFA score ≥ 2 is considered a critical index for sepsis [18]. We use this rule of thumb to design the first reward term as follows:

$${r_t}^{SOFA} = \mathbb{1}_{\boldsymbol{a_t} \neq \boldsymbol{0}} \cdot \mathbb{1}_{f(\cdot) \geq 2} \, , \text{ where } f(\cdot) = m_t^{SOFA} - m_{t-1}^{SOFA}. \tag{3}$$

The raw score m_t^{SOFA} at each time step t is evaluated using current patient labs and vitals [19].

Treatment onset: The second term is an indicator variable for rewards capturing whether or not there is some treatment or intervention initiated at the next time step, $\boldsymbol{s_{t+1}}$:

$${r_t}^{treat} = \mathbb{1}_{\boldsymbol{a_t} \neq \boldsymbol{0}} \cdot \sum_{i \in M} \mathbb{1}_{\boldsymbol{s_{t+1}}(\text{treatment } i \text{ was given})}, \tag{4}$$

where M denotes the set of disease-specific interventions of interest. Again, the reward term is positive if a lab is ordered; this is based on the rationale that, if a lab test is ordered and immediately followed by an intervention, the test is likely to have provided actionable information. Possible interventions include antibiotics, vasopressors, dialysis or ventilation.

Lab redundancy: The term ${r_t}^{info}$ denotes the feedback from taking one or more lab tests with novel information. We quantify this by using the mean absolute difference between the last observed value and predictive mean from the MOGP as a proxy for the information available:

$${r_t}^{info} = \sum_{\ell=1}^{L} max\left(0, g(\cdot) - c_\ell\right) \cdot \mathbb{1}_{\boldsymbol{a_t}[\ell]=1} \, , \text{ where } g(\cdot) = \left| \frac{m_t^{(\ell)} - y_t^{(\ell)}}{\sigma_t^{(\ell)}} \right|, \tag{5}$$

where σ_t^ℓ is the normalization coefficient for lab ℓ, and the parameter c_ℓ determines the minimum prediction error necessary to trigger a reward; in our experiments, this is set to the median prediction error for labs ordered in the training data. The larger the deviation from current forecasts, the higher the potential information gain, and in turn the reward if the lab is taken.

Lab cost: The last term in the reward function, ${r_t}^{cost}$ adds a penalty whenever any test is ordered to reflect the effective "cost" of taking the lab at time t.

$$ {r_t}^{cost} = \sum_{\ell=1}^{L} \exp\left(-\frac{\Delta_t^{(\ell)}}{\Gamma_\ell}\right) \cdot \mathbb{1}_{\boldsymbol{a_t}[\ell]=1}, \tag{6} $$

where Γ_ℓ is a decay factor that controls the how fast the cost decays with the time Δ_t elapsed since the last measurement. In our experiments, we set $\Gamma_\ell = 6\ \forall \ell \in L$.

2.3. *Learning optimal policies*

Once we extract sequences of states, actions, and rewards from the ICU data, we can generate a dataset of one-step transition tuples of the form $\mathcal{D} = \{\langle s_t^n, a_t^n, s_{t+1}^n \rangle, r_t^n\}$, $n = 1...|\mathcal{D}|$. These tuples can then be used to learn an estimate of the Q-function, $\hat{Q} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ —where $d = 4$ is the dimensionality of the reward function—to map a given state-action pair to a vector of expected cumulative rewards. Each element in the Q-vector represents the estimated value of that state-action pair according to a different objective. We learn this Q-function using a variant of Fitted Q-iteration (FQI) with extremely randomized trees [13, 20]. FQI is a batch off-policy reinforcement learning algorithm that is well-suited to clinical applications where we have limited data and challenging state dynamics. The algorithm adapted here to handle vector-valued rewards is based on Pareto-optimal Fitted-Q [15].

In order to scale from the two-stage decision problem originally tackled to the much longer admission sequences here (≥ 24 time steps), we define a stricter pruning of actions: at each iteration we eliminate any *dominated* actions for a given state—those actions that are outperformed by alternatives for all elements of the Q-function—and retain only the set $\Pi(s) = \{a : \nexists a' \, (\forall \, d, \ \hat{Q}_d(s,a) < \hat{Q}_d(s,a'))\}$ for each s. Actions are further filtered for *consistency*: we might consider feature consistency to be defined as rewards being linear in each feature space [15]. Here, we relax this idea to filter out only those actions from policies that cannot be expressed by our nonlinear tree-based classifier. The function will still yield a non-deterministic policy (NDP) as, in most cases, there will not be a strictly optimal action that achieves the highest Q_d for all d. We suggest one possible approach for reducing the NDP to give a single best action for any given state based on practical considerations in the next section.

3. Results

Following the extraction of our 6,060 admissions and resampling in hourly intervals using the forecasting MOGP, we partitioned the cohort into training and test sets of 3,636 and 2,424 admissions respectively. This gave approximately 500,000 one-step transition tuples of the form $\langle s_t, a_t, s_{t+1}, r_t \rangle$ in the training set, and over 350,000 in the test set. We then ran batched FQI with these samples for 200 iterations with discount factor $\gamma = 0.9$. Each iteration took

Algorithm 1 Multi-Objective Fitted Q-iteration with strict pruning (MO-FQI)

Input:
One-step transitions $\mathcal{F} = \{\langle s_t^n, a_t^n, s_{t+1}^n \rangle, r_{t+1}^n\}_{n=1:|\mathcal{F}|}$;
Regression parameters θ; action space $\mathcal{A}$; subset size N
Initialize $Q^{(0)}(s_t, a_t) = \mathbf{0} \in \mathbb{R}^d \quad \forall s_t \in \mathcal{F}$, $a_t \in \mathcal{A}$
for *iteration* $k = 1 \rightarrow K$ **do**
  Sample $subset_N \sim \mathcal{F}$; initialize $S \leftarrow []$
  for $i \in subset_N$ **do**
    Generate set $\Pi(s_i)$ using $Q^{(k-1)}$
    Initialize classification parameters ϕ
    $\phi \leftarrow classify(s_i, a_i)$
    for $\pi_i \in \Pi$: **do**
      $a' \leftarrow \pi_i(s_{i+1}) \cap predict(s_{i+1}, \phi)$
      $Q^{(k)}(s_i, a_i) \leftarrow r_{i+1} + \gamma Q^{(k-1)}(s_{i+1}, a')$
    end
    $S \leftarrow append(S, \langle (s_i, a_i),\ Q^{(k)}(s_i, a_i) \rangle)$
  end
  $\theta \leftarrow regress(S)$
end
Result: θ

100,000 transitions, sampled from the training set, with probability inversely proportional to the frequency of the action in the tuple. The vector-valued outputs of estimated Q-function were then used to obtain a non-deterministic policy for each lab considered (Section 2.3). We chose to collapse this set to a practical deterministic policy as follows:

$$\Pi(s) = \begin{cases} 1, & \hat{Q}_d(s, a=0) < \hat{Q}_d(s, a=1) + \varepsilon_d, \quad \forall\, d \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

In particular, a lab should be taken ($\Pi(s) = 1$) *only if* the action is *optimal*, or estimated to outperform the alternative for all objectives in the Q-function. This strong condition for ordering a lab is motivated by the fact that one of our primary objectives here is to minimize unnecessary ordering; the variable ε_d allows us to relax this for certain objectives if desired. For example, if cost is a softer constraint, setting $\varepsilon_{cost} > 0$ is an intuitive way to specify this preference in the policy. In our experiments, we tuned ε_{cost} such that the total number of recommended orders of each lab approximates the number of actual orders in the training set.

With a deterministic set of optimal actions, we could train our final policy function $\pi : \mathcal{S} \rightarrow A$; again, we used extremely randomized trees. The estimated Gini feature importances of the policies learnt show that in the case of lactate the most important features are the mean and measured lactate, the time since last lactate measurement (Δ) and the SOFA score (Figure 2). These relative importance scores are expected: a change in SOFA score may indicate the onset of sepsis, and in turn warrant a lactate test to confirm a source of infection, fitting typical clinical protocol. For the other three policies (WBC, creatinine, BUN) again the time since last measurement of the respective lab tends be prominent in the policy, along with the Δ terms

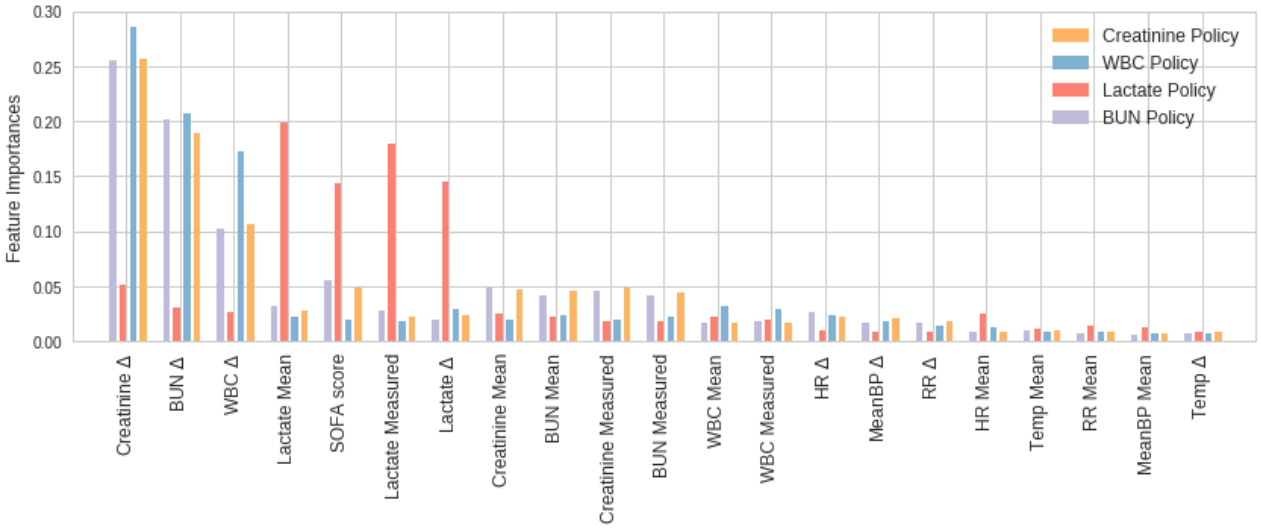


Fig. 2. **Feature importances** over the 21-dimensional state space, for each of our four policies.

for the other two labs. This suggests an overlap in information in these tests: For example, abnormally high white blood cell count is a key criteria for sepsis; severe sepsis often cascades into renal failure, which is typically diagnosed by elevated BUN and creatinine levels [21].

Once we have trained our policy functions, an additional component is added to our final recommendations: we introduce a *budget* that suggests taking a lab at the end of every 24 hour period for which our policy recommends no orders. This allows us to handle regions of very sparse recommendations by the policy function, and reflects clinical protocols that require minimum daily monitoring of key labs. In the policy for lactate orders in a typical patient admission, looking at the timing of the actual clinician orders, recommendations from our policy, and suggested orders from the budget framework, the actions are concentrated where lactate values are increasingly abnormal, or at sharp rises in SOFA score (Figure 3).

3.1. *Off-Policy Evaluation*

We evaluated the quality of our final policy recommendations in a number of ways. First, we implemented the per-step weighted importance sampling (PS-WIS) estimator to calculate the value of the policy π_e to be evaluated:

$$\hat{V}_{\text{PS-WIS}}(\pi_e) = \sum_{i=1}^{n} \sum_{t=0}^{T-1} \gamma_{\text{WIS}}^{t} \left[\frac{\rho_t^{(i)}}{\sum_{i=1}^{n} \rho_t^{(i)}} \right] r_t^{(i)}, \quad \text{where } \rho_t = \prod_{j=0}^{t-1} \frac{\pi_e(s_j|a_j)}{\pi_b(s_j|a_j)},$$

given data collected from behaviour policy π_b [22]. The behaviour policy was found by training a regressor on real state-action pairs observed in the dataset. The discount factor was set to $\gamma_{\text{WIS}} = 1.0$, so all time steps contribute equally to the value of a trajectory.

We then compared estimates for our policy (MO-FQI) against the behaviour policy and a set of randomized policies as baselines. These randomized policies were designed to generate random decisions to order a lab, with probabilities $p = \{0.01, p_{emp}, 0.5\}$, where p_{emp} is the empirical probability of an order in the behaviour policy. For each p, we evaluated ten randomly generated policies and averaged performance over these. We observed that MO-FQI outperforms

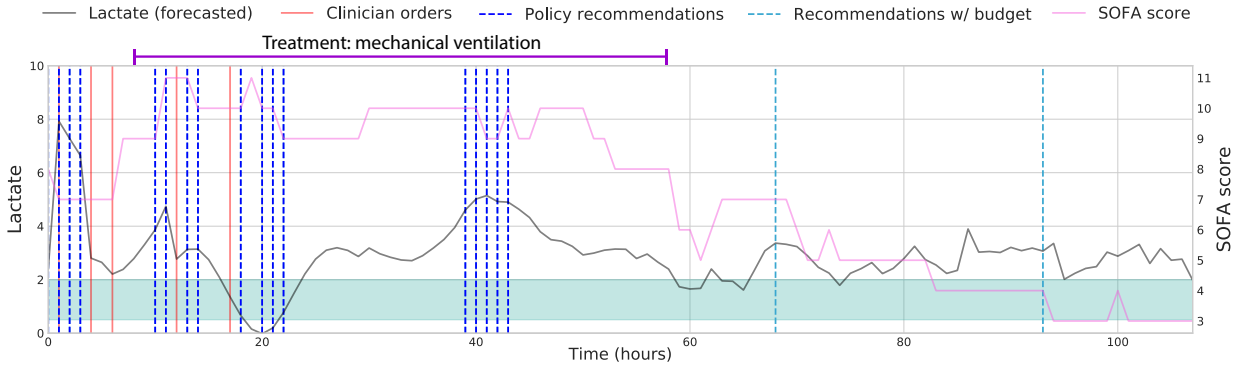


Fig. 3. **Demonstration of one test trajectory of recommending lactate orders.** The shaded green region denotes the range of normal lactate values (0.5–2 mmol/L).

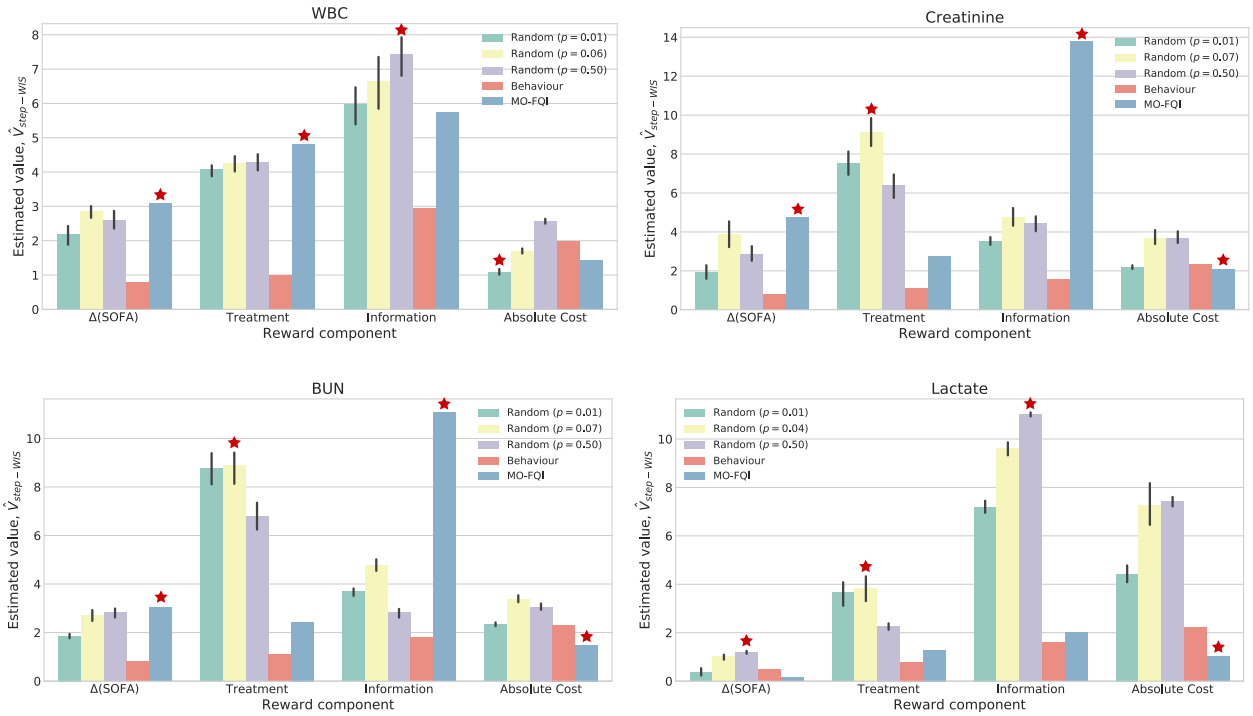


Fig. 4. **Evaluating $\hat{V}_d(\pi_e)$** for each reward component d, across policies for four labs. For randomized policies, error bars show standard deviations across 10 trials. The ($\star$) indicates the best performing policy for each reward component; for *absolute* cost, this corresponds to the lowest estimated value.

the behaviour policy across all reward components, for all four labs (Figure 4). Our policy also consistently approximately matches or outperforms other policies in terms of cost—note that lower cost is better—even with the inclusion of the slack variable ε_{cost} and the budget framework. Across the remaining objectives, MO-FQI outperforms the random policy in at least two of three components for all but lactate. This may be due in part to the relatively sparse orders for lactate resulting in higher variance value estimates.

In addition to evaluating using the per-step WIS estimator, we looked for more intuitive measures of how the final policy influences clinical practice. We computed three metrics here: (i) estimated reduction in total number of orders, (ii) mean information gain of orders taken, and (iii) time intervals between labs and subsequent treatment onsets.

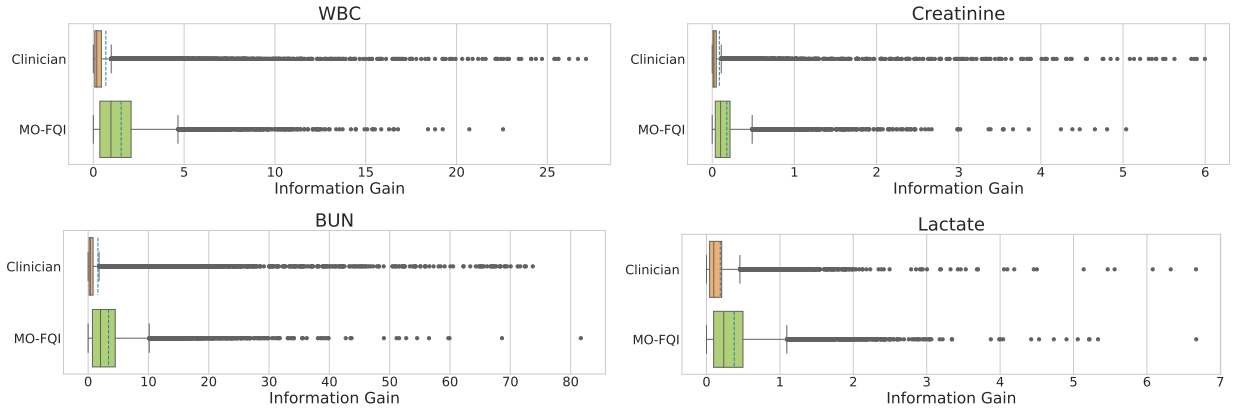


Fig. 5. **Evaluating Information Gain** of clinician actions against MO-FQI across all labs: the mean information in labs ordered by clinicians is consistently outperformed by MO-FQI: 0.69 vs 1.53 for WBC; 0.09 vs 0.18 for creatinine; 1.63 vs 3.39 for BUN; 0.19 vs 0.38 for lactate.

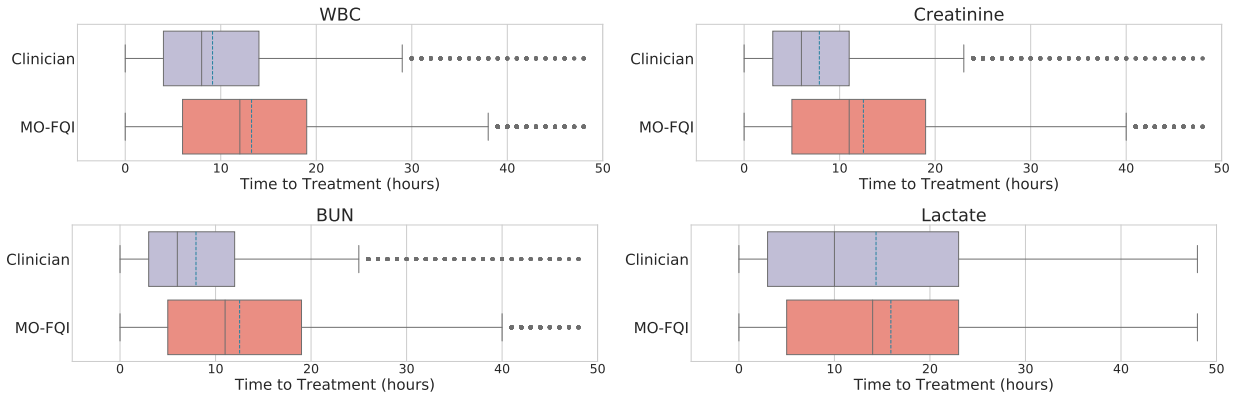


Fig. 6. **Evaluating Time to Treatment Onset** of lab orders by the clinician against MO-FQI across all labs: the mean time intervals are as follows (Clinician vs MO-FQI): 9.1 vs 13.2 for WBC; 7.9 vs 12.5 for creatinine; 8.0 vs 12.5 for BUN; 14.4 vs 15.9 for lactate.

In evaluating the total number of recommended orders, we first filter a sequence of recommended orders to the just the first (onset) of recommendations if there are no clinician orders between them. We argue that this is a fair comparison as subsequent recommendations are made without counterfactual state estimation, i.e., without assuming that the first recommendation was followed the clinician. Empirically, we find that the total number of recommendations is considerably reduced. For instance, in the case of recommending WBC orders, our final policy reports 12,358 orders in the test set, achieving a reduction of 44% from the number of true orders (22,172). In the case of lactate, for which clinicians' orders are the least frequent (14,558), we still achieved a reduction of 27%.

We also compared the approximate information gain of the actions taken by the estimated policy, in comparison with the policy used in the collected data. To do this, we defined the information gain at a given time by looking at the difference between the *approximated* true value of the target lab, which we impute using the MOGP model given all the observed values, and the forecasted value, computed using only the values observed before the current time. The distribution of aggregate information gain for orders recommended by our policy and actual

clinician's orders in the test set shows higher mean information gain with MO-FQI (Figure 5).

Lastly, we considered the time to onset of critical interventions, which we define to include initiation of vasopressors, antibiotics, mechanical ventilation or dialysis. We first obtained a sequence of treatment onset times for each test patient; for each of these time points, we traced back to the earliest observed or recommended order taking place within the past 48 hours, and computed the time between these: $\Delta_t = t_{treatment} - t_{order}$. The distribution of time-to-treatment for labs taken by the clinician in the true trajectory against that for recommendations from our policy, for all four labs, shows that the recommended orders tend to happen earlier than the actual time of an order by the clinician—on average over an hour in advance for lactate, and more that four hours in advance for WBC, creatinine, and BUN (Figure 6).

4. Conclusion

In this work, we propose a reinforcement learning framework for decision support in the ICU that learns a compositional optimal treatment policy for the ordering of lab tests from sub-optimal histories. We do this by designing a multi-objective reward function that reflects clinical considerations when ordering labs, and adapting methods for multi-objective batch RL to learning extended sequences of Pareto-optimal actions. Our final policies are evaluated using importance-sampling based estimators for off-policy evaluation, metrics for improvements in cost, and reducing redundancy of orders. Our results suggest that there is considerable room for improvement on current ordering practices, and the framework introduced here can help recommend best practices and be used to evaluate deviations from these across care providers, driving us towards more efficient health care. Furthermore, the low risk of these types of interventions in patient health care reduces the barrier of testing and deploying clinician-in-the-loop machine learning-assisted patient care in ICU settings.

References

1. T. Badrick, Evidence-based laboratory medicine *The Clinical Biochemist Reviews* **34** (The Australian Association of Clinical Biochemists, 2013).
2. M. Zhi, E. L. Ding, J. Theisen-Toupal, J. Whelan and R. Arnaout, The landscape of inappropriate laboratory testing: a 15-year meta-analysis *PloS one* **8** (Public Library of Science, 2013).
3. T. Loftsgard and R. Kashyap, Clinicians role in reducing lab order frequency in icu settings *J Perioper Crit Intensive Care Nurs* **2**2016.
4. R. L. Konger, P. Ndekwe, G. Jones, R. P. Schmidt, M. Trey, E. J. Baty, D. Wilhite, I. A. Munshi, B. M. Sutter, M. Rao *et al.*, Reduction in unnecessary clinical laboratory testing through utilization management at a us government veterans affairs hospital *American journal of clinical pathology* **145** (Oxford University Press, 2016).
5. ICUMedical, Reducing the risk of iatrogenic anemia and catheter-related bloodstream infections using closed blood sampling (ICU Medical Inc., 2015).
6. J. Lee and D. M. Maslove, Using information theory to identify redundancy in common laboratory tests in the intensive care unit *BMC medical informatics and decision making* **15** (BioMed Central, 2015).

REFERENCES

7. F. Cismondi, L. A. Celi, A. S. Fialho, S. M. Vieira, S. R. Reti, J. M. Sousa and S. N. Finkelstein, Reducing unnecessary lab testing in the icu with artificial intelligence *International journal of medical informatics* **82** (Elsevier, 2013).
8. Y. Luo, P. Szolovits, A. S. Dighe and J. M. Baron, Using machine learning to predict laboratory test results *American Journal of Clinical Pathology* **145**2016.
9. M. Ghassemi, M. A. F. Pimentel, T. Naumann, T. Brennan, D. A. Clifton, P. Szolovits and M. Feng, A multivariate timeseries modeling approach to severity of illness assessment and forecasting in ICU with sparse, heterogeneous clinical data, in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.
10. L.-F. Cheng, G. Darnell, C. Chivers, M. Draugelis, K. Li and B. Engelhardt, Sparse multi-output gaussian processes for medical time series prediction (2017).
11. S. Nemati, M. M. Ghassemi and G. D. Clifford, Optimal medication dosing from suboptimal clinical examples: A deep reinforcement learning approach, in *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, 2016.
12. A. Raghu, M. Komorowski, L. A. Celi, P. Szolovits and M. Ghassemi, Continuous state-space models for optimal sepsis treatment: a deep reinforcement learning approach, in *Proceedings of the Machine Learning for Health Care, MLHC 2017, Boston, Massachusetts, USA, 18-19 August 2017*, 2017.
13. N. Prasad, L. Cheng, C. Chivers, M. Draugelis and B. Engelhardt, A reinforcement learning approach to weaning of mechanical ventilation in intensive care units, in *Uncertainty in Artificial Intelligence 2017*, 1 2017.
14. S. Natarajan and P. Tadepalli, Dynamic preferences in multi-criteria reinforcement learning, in *Proceedings of the 22nd international conference on Machine learning*, 2005.
15. D. J. Lizotte and E. B. Laber, Multi-objective markov decision processes for data-driven decision support *Journal of Machine Learning Research* **17**2016.
16. A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi and R. G. Mark, Mimic-iii, a freely accessible critical care database *Scientific data* **3** (Nature Publishing Group, 2016).
17. P. E. Marik and A. M. Taeb, *Journal of thoracic disease* **9**, p. 943 (2017).
18. M. Singer, C. S. Deutschman, C. W. Seymour, M. Shankar-Hari, D. Annane, M. Bauer, R. Bellomo, G. R. Bernard, J.-D. Chiche, C. M. Coopersmith *et al.*, The third international consensus definitions for sepsis and septic shock (sepsis-3) *Jama* **315** (American Medical Association, 2016).
19. J.-L. Vincent, G. S. Martin and M. M. Levy, qsofa does not replace sirs in the definition of sepsis *Critical Care* **20** (BioMed Central, 2016).
20. D. Ernst, P. Geurts and L. Wehenkel, Tree-based batch mode reinforcement learning *Journal of Machine Learning Research* **6**2005.
21. M. R. Clarkson, B. M. Brenner and C. Magee, *Pocket Companion to Brenner and Rector's The Kidney E-Book* (Elsevier Health Sciences, 2010).
22. D. Precup, R. S. Sutton and S. P. Singh, Eligibility traces for off-policy policy evaluation, in *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00 (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000).

SINGLE CELL ANALYSIS, WHAT IS IN THE FUTURE?

Lana X. Garmire[†]
Department of Computational Medicine and Bioinformatics, University of Michigan
1600 Huron Parkway, Ann Arbor, 48105, USA
Email: lgarmire@med.umich.edu

Guo-Cheng Yuan
Department of Biostatistics and Computational Biology, Dana-Farber Cancer Institute and Harvard Chan School of Public Health, Boston, MA 02215, USA
Email: gcyuan@jimmy.harvard.edu

Rong Fan
Biomedical Engineering Department, Yale University
55 Prospect Street, MEC 213, New Haven, CT 06520, USA
Email: rong.fan@yale.edu

Gene W Yeo
Cellular and Molecular Medicine, University of California at San Diego
2880 Torrey Pines Scenic Dr. La Jolla, CA92037, USA
Email: geneyeo@ucsd.edu

John Quackenbush[††]
Department of Biostatistics, Harvard University
Dana-Farber Cancer Institute Smith 822A, Boston, MA 02215, USA
Email: johnq@jimmy.harvard.edu

Abstract

Single-cell genomics technology is an exciting emerging area that holds the promise to revolutionize our understanding of diseases and associated biological processes. It allows us to explore processes active in bulk tissue samples, survey tissue complexity, characterize heterogeneous cell populations and explore the role of cellular heterogeneity and interactions in disease. To deal with these new experimental data, new computational methods, software, and data portals to analyze, integrate and interpret the complexity of the system are clearly needed. The many areas where new analytical methods are needed include: (1) computational methods to identify bona fide patterns of gene

[†]LG's work is supported by grants K01ES025434 awarded by NIEHS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov), R01 LM012373 awarded by NLM, and R01 HD084633 award by NICHD.
[††]JQ's work is supported by a grant from the US National Cancer Institute, R35CA220523.

expression, mutations, or DNA methylation among single cells; (2) imaging of gene expression or *in situ* transcriptomic analysis to allow study of the spatial-temporal relationships of single cells in complex tissues; (3) new tools and methods to integrate multi-omics single cell data that can handle the sparsity associated with those data, and (4) new software packages and data portals to enable cloud/HPC deployment to both developers and non-informatics end-users. Here we briefly review the state-of-the-art single cell analysis methods, ranging from clustering to visualization, and discuss the future directions of single cell bioinformatics that overcomes the computational and technical challenges as well as promotes the wide-spread adoption in biomedical research labs.

Keywords: single cell; bioinformatics; software; computation; analysis; sequencing; clustering; visualization; pipeline

1. Background

Single cell genomics represents a major breakthrough in biological science. The technology has challenged both our understanding of how cells function alone and in communities, and the methods we have developed to analyze data from bulk tissue samples[1–3]. The most widely used single-sell technology is single cell RNA-sequencing (scRNA-seq). Platforms, such as Drop-seq, Fluidigm C1 system, and 10x Genomics Chromium System, have made it possible to study a large number of single cells in various biological systems in individual labs as well a world-wide consortium, the Human Cell Atlas, which has as its goal the creation of a reference human cell data resource. Beyond understanding fundamentals of gene expression patterns in each cell, this technology has been utilized in many areas of applications, such as characterizing developmental processes, discovering new cell types, revealing the heterogeneity within tumors, depicting tumor microevolution, as well as identifying novel biomarkers for disease progression and drug resistance[4].

As an exciting frontier of genomics technology, scRNA-seq data analysis is also computationally difficult, due in part to both the technology and basic biology of single cells[5]. For example, as each cell has very limited amount of RNA molecules and the capturing technology is not even close to 100% efficient, specific RNAs may be omitted and appear as “drop-outs”, meaning that the assay fails to capture them and thus their expression value is falsely reported as zero. PCR is sometimes used to amplify RNA as part of the product, “jackpotting” can occur; leading to inflated read counts for other genes. When using droplet based methods, occasionally multiple cells may be incorporated in the droplet, leading to doublets which can confuse data interpretation. Additionally, batch effect is known in single-cell experiments, like other omics assays. All of these factors have impact on estimating true expression values and each requires the use of rigorous modeling methods to estimate the effect and correct for it.

To address various issues such as the ones stated above, we have seen numerous computational methods reported recently. There are also new bioinformatics pipelines, packages and data portals available for public use, depending on users' background and preference[6,7]. A scRNA-seq analysis pipeline usually includes the following preprocessing steps: batch-effect removal, outlier removal, normalization, imputation and gene filtering. Downstream analyses include methods for clustering, differential expression analysis, pathway/ontology enrichment analysis, protein network interaction mapping, and pseudo-time construction. Read counts, the representation of gene expression (GE), are conventionally used as the inputs for bioinformatics analysis. However, some researchers also proposed to use other information, such as small nucleotide variation (SNV) as less bias-prone features to conduct downstream functional analysis[8].

2. Summary of single cell analysis session at PSB 2019

In the single cell analysis session at PSB 2019, four submitted full-length manuscripts were accepted. They cover a range of topics from visualization, pseudo-time inference, and evaluation of clustering methods to probabilistic approach to include gene expression data for metabolic modeling.

The work from Ouyang's group reports on a new method called LISA: Landmark Isomap for Single cell Analysis. It is an unsupervised method that constructs cell trajectory and the pseudo-time relationships. The authors present a thorough comparison to two widely used methods, TSCAN and Monocle2, using both simulated and real data. Their analysis concludes that LISA captures the biology of the system being analyzed more efficiently than Monocle2 or TSCAN, yet is more computationally efficient. Thus, it can be applied to ever-larger scRNA-seq data sets and might potentially useful in the analysis of other single cell omics data.

Huang et al. use a topological analysis method called Mapper to visualize single cell RNAseq subpopulation data. Topological analysis of scRNA-seq is very interesting and allows the delineation of complex relationships that extend beyond the simple clustering that is more commonly used. The authors compared their method to tSNE and showed that Mapper better preserves continuous structure in the data.

In Wolpert and Macready's No Free Lunch Theorem paper, they argued against general purpose algorithms tested on small data sets and built without taking advantage of prior knowledge of the system being analyzed [9]. The work of Greene et al. is a case study in this regard, applied to scRNA-seq analysis. The authors analyzed the effects of parameter tuning in a variational autoencoder

(VAE) on the clustering of simulated scRNA-seq results. They warned that without proper parameter sets, deep learning results can lead to significant error.

Gold et al. presents new application of prior work on sparsely-connected autoencoders (SSCA) and variation autoencoders (SSCVA), in single cell RNA-seq analysis. This paper replaces those statistical methods that were popular in this field with machine learning methods and adds some interpretability by mapping genes to gene sets. The results of SSCVA appear to be better than SSCA, but the gene-set level extraction is not better than raw gene expression.

3. Single cell analysis, what is in the future?

At present, scRNA-seq is the most widely used method of single cell analysis. As we previously noted that there are many choices for each of the various steps along the data analysis pipeline for single cell data. However, there is no clear consensus as to what represents best practices. This, in large part, represents the fact that scRNA-seq is so new that even discoveries of apparently new cell types in a bulk tissue sample need substantial validation using other methods and independent data sets before one can consider them to be reliable. As a result, there are no reliable benchmark data sets that can be used to objectively evaluate the many methods and pipelines that are now available.

Nevertheless, scRNA-seq data sets provide the opportunity to explore tens of thousands of individual cells—data sets that dwarf the number of samples in most other gene expression studies. Such expansive data provides many new opportunities for methods development and the use of creative approaches that can handle massive yet sparse data. Ultimately, these new methods must be critically assessed, and validation will require both careful evaluation of the methods and the design and conduct of experimental studies.

What is most exciting about single cell field is that the technology continues to rapidly evolve, setting the stage for further methodology development. One particularly interesting application is spatially-informed single cell analysis, in which the spatial relationship between various cell types is preserved. Current scRNA-seq protocols first dissociate individual cells and remove debris, followed by single cell encapsulation and sequencing. Analysis of such expression data will require new computational methods to detect spatial patterns and model the relationships between cell types and their associations with various phenotypes.

Another exciting possibility is the development of multi-omic analysis in which genomic, transcriptomic, epigenomic, or other data types are collected on each cell[10,11]. Development of these methods for single cell data presents great challenges as noisy or missing data can lead to incorrect conclusions about the interaction between the sources of those data. Computational methods developed for bulk cell multi-omics integration[12], may be the first-line option for single-cell multi-omics integration, after significant efforts on cleaning, imputation, and normalization that preserve relationships between data types.

Finally, efforts that improve user's experience will be very valuable. One area is increasingly recognized as essential is the development of new methods for visual representation of complex data. With the potential to generate data on millions of cells from hundreds of cell types in a single experiment, there is a clear need for methods that can show the relationships that exist between those cell types that reflect their lineages, relationships, and interacting processes between cell types which are related to the phenotypes. GUI based data portals for interactive scRNA-seq analysis will also help the researchers to navigate through massive amount of information.

Regardless of which area one chooses to focus, it is clear that there are many opportunities for methods development and application in single cell analysis. More importantly, single cell analysis promises to help us understand the complexities of human health and disease—but only if we have appropriate analytical methods.

References

1. Eberwine, J., Sul, J.-Y., Bartfai, T. & Kim, J. The promise of single-cell sequencing. *Nat. Methods* **11,** 25–27 (2014).
2. Poirion, O. B., Zhu, X., Ching, T. & Garmire, L. Single-Cell Transcriptomics Bioinformatics and Computational Challenges. *Front. Genet.* **7,** 163 (2016).
3. Stegle, O., Teichmann, S. A. & Marioni, J. C. Computational and analytical challenges in single-cell transcriptomics. *Nat. Rev. Genet.* **16,** 133–145 (2015).
4. Yuan, G.-C. *et al.* Challenges and emerging directions in single-cell analysis. *Genome Biol.* **18,** 84 (2017).
5. Hwang, B., Lee, J. H. & Bang, D. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Exp. Mol. Med.* **50,** 96 (2018).
6. Zhu, X. *et al.* Granatum: a graphical single-cell RNA-Seq analysis pipeline for genomics scientists. *Genome Med.* **9,** 108 (2017).
7. Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat. Biotechnol.*

36, 411–420 (2018).

8. Poirion, O. B., Zhu, X., Ching, T. & Garmire, L. X. Using Single Nucleotide Variations in Single-Cell RNA-Seq to Identify Subpopulations and Genotype-phenotype Linkage. *bioRxiv* 095810 (2018). doi:10.1101/095810
9. Wolpert, D. H. & Macready, W. G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1,** 67–82 (1997).
10. Packer, J. & Trapnell, C. Single-Cell Multi-omics: An Engine for New Quantitative Models of Gene Regulation. *Trends Genet.* **34,** 653–665 (2018).
11. Ortega, M. A. *et al.* Using single-cell multiple omics approaches to resolve tumor heterogeneity. *Clin. Transl. Med.* **6,** 46 (2017).
12. Huang, S., Chaudhary, K. & Garmire, L. X. More Is Better: Recent Progress in Multi-Omics Data Integration Methods. *Front. Genet.* **8,** 84 (2017).

LISA: Accurate reconstruction of cell trajectory and pseudo-time for massive single cell RNA-seq data

Yang Chen[1], Yuping Zhang[2,4,6] and Zhengqing Ouyang[1,3,4,5,*]
[1]*The Jackson Laboratory for Genomic Medicine, Farmington, CT 06032, USA*
[2]*Department of Statistics, University of Connecticut, Storrs, CT 06269, USA*
[3]*Department of Biomedical Engineering, University of Connecticut, Storrs, 06269, CT, USA*
[4]*Institute for Systems Genomics, University of Connecticut, Storrs, CT 06269, USA*
[5]*Department of Genetics and Genome Sciences, University of Connecticut, Farmington, 06030, CT, USA*
[6]*Center for Quantitative Medicine, University of Connecticut, Farmington, CT 06030, USA*

Cell trajectory reconstruction based on single cell RNA sequencing is important for obtaining the landscape of different cell types and discovering cell fate transitions. Despite intense effort, analyzing massive single cell RNA-seq datasets is still challenging. We propose a new method named Landmark Isomap for Single-cell Analysis (LISA). LISA is an unsupervised approach to build cell trajectory and compute pseudo-time in the isometric embedding based on geodesic distances. The advantages of LISA include: (1) It utilizes k-nearest-neighbor graph and hierarchical clustering to identify cell clusters, peaks and valleys in low-dimension representation of the data; (2) based on Landmark Isomap, it constructs the main geometric structure of cell lineages; (3) it projects cells to the edges of the main cell trajectory to generate the global pseudo-time. Assessments on simulated and real datasets demonstrate the advantages of LISA on cell trajectory and pseudo-time reconstruction compared to Monocle2 and TSCAN. LISA is accurate, fast, and requires less memory usage, allowing its applications to massive single cell datasets generated from current experimental platforms.

Keywords: single cell RNA-seq; cell trajectory; pseudo-time; manifold learning.

1. Introduction

Single cell RNA sequencing (scRNA-seq) is emerging to revolutionize the study of development and disease processes. It has been widely used to investigate the dynamic gene expression landscape, cell type identification, cell state transition, and pseudo-time estimation at single cell level [1-7].

An important computational issue of scRNA-seq analysis is on the reconstruction of cell trajectory and pseudo-time for individual cells. Among existing methods, Monocle2 [8], TSCAN [9], and Slingshot [10] are shown to have relatively better performance [4]. Monocle2 utilizes the principal component analysis (PCA) and discriminative dimensionality reduction tree (DDRTree) [11]. It is often able to build a tree structure. But an arbitrarily large cell cluster number (usually > 100) is used for minimum spanning tree (MST) construction. Slingshot extends the principle curve method to fit the lineages built on MST. Similar to DDRTree, it makes the tree structure smoother. But the users need to determine the dimension reduction and clustering methods and generate cell lineages before using Slingshot. TSCAN uses Gaussian mixture models and the Bayesian information criterion for automatically determining cell cluster number, and then build cell lineages by MST on cluster centers in the PCA space. TSCAN and Slingshot can only infer cell orders in

* To whom correspondence should be addressed. Email: zhengqing.ouyang@jax.org

each cell lineage and are not able to estimate the global pseudo-time of all cells. Most of the existing methods were only applied to small scRNA-seq datasets. It is not clear whether they are feasible for massive scRNA-seq datasets.

Large scale scRNA-seq technologies [5], such as 10x Genomics [12], make it possible to profile more than tens or hundreds of thousands of cells. Such massive scRNA-seq datasets promote the development of new cell trajectory reconstruction methods [1-4]. Existing literature has used empirical approaches to study cell lineages supervised by known time labels and cell marker genes [1-4]. It is not known how well one can reconstruct complex cell trajectory and pseudo-time by unsupervised approaches.

We have developed the Landmark Isomap for Single-cell Analysis (LISA), an unsupervised method aiming to reconstruct cell trajectory and pseudo-time for massive scRNA-seq datasets. Briefly, LISA first automatically determines cell clusters, peaks and valleys based on k-nearest-neighbor graph (kNN-graph) [13] and hierarchical clustering. Then it maps cells into the isometric embedding based on geodesic distances [14] using the peaks and valleys as landmarks. It then build the MST on the cluster centers as the main cell trajectory in the isometric embedding. Finally, it computes the pseudo-time by projecting cells onto the MST.

The rest of the paper is organized as follows: in Methods, we introduce the algorithm of LISA. In Results, we assess LISA on a simulated dataset, and two large scRNA-seq datasets. One dataset is on human embryo development containing 1,364 cells [15]. The other is on zebrafish embryogenesis including 38,731 cells [2]. We compared LISA with Monocle2 and TSCAN on cell trajectory reconstruction. We also compared LISA with Monocle2 on global pseudo-time estimation. The paper is concluded with a discussion.

2. Methods

The workflow of LISA is shown in Fig. 1. We can start with either unnormalized or normalized gene expression values for K genes and N cells. If the input data are raw read counts, log2-transformation will be performed. Lowly expressed genes will be filtered. Optionally, the genes with low variances will be removed. The details of the LISA method will be introduced as follows.

2.1. *Visualize cells by PCA and t-Distributed Stochastic Neighbor Embedding (t-SNE)*

PCA and t-SNE are two common dimensionality reduction methods for visualization. We use PCA to select top ranked PCs that keep the major variations in the data. We then derive the t-SNE [16] coordinates based on the selected PCs.

2.2. *Identify cell clusters, peaks, and valleys*

We identify cell clusters, peaks, and valleys based on kNN-graph and hierarchical clustering. We construct the kNN-graph based on the Euclidean distance with a default k as 50. To improve the speed, we use the kd-tree [13] to construct the kNN-graph, resulting a running time of O(NlogN), where N is the number of cells.

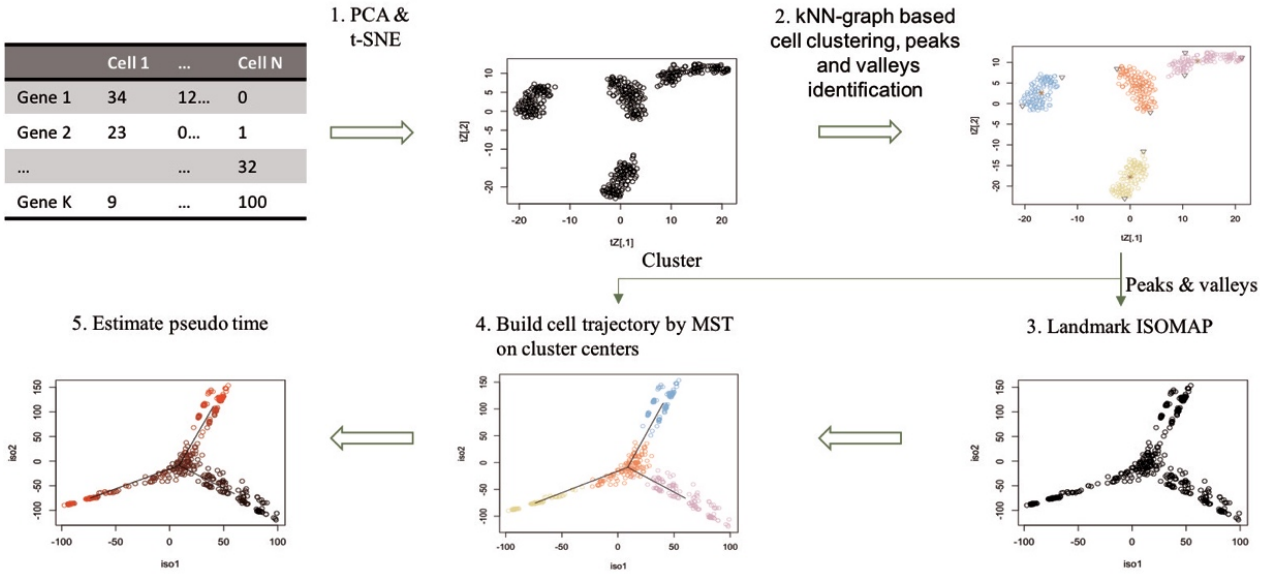


Figure 1. Workflow of LISA. (1) Do PCA for the gene expression matrix (K genes * N cells) and select top ranked PCs. Then the N cells with the selected PCs are mapped into the t-SNE embedding. (2) Estimate cell density in the t-SNE embedding and build the k-NN graph to find peaks and valleys. Then perform hierarchical clustering until each cluster contains one peak point (star shape). Valley points are shown as inverted triangles. (3) Using peaks and valleys as landmark points and map the N cells with the selected PCs into the isometric embedding based on geodesic distances. (4) Build the main cell trajectory using MST on the cluster centers in the isometric embedding. (5) Estimate global pseudo-time by projecting cells onto the main cell trajectory.

After building the kNN-graph, we then search for cell peaks and valleys. We first estimate cell density based on a nonparametric density estimation approach [17]. For each cell, if its density value is higher than all the k nearest neighbors, it is regarded as a peak. Conversely, if its density value is lower than all the k nearest neighbors, it is determined as a valley. Then we propose an iterative hierarchical clustering method as follows:

1. Do hierarchical clustering in the t-SNE embedding. Cut the resulting dendrogram so that the number of clusters is equal to the number of peaks.
2. Among the resulting clusters, if one cluster contains more than one peak, perform hierarchical clustering again on this cluster with the cluster number equal to the peak number in it.
3. Do step 2 until each cluster contains at most one peak.
4. For a cluster without a peak, merge it with another cluster containing a nearest peak. The nearest peak is defined as the one that is closest to the cluster with the minimum distance to the cells in the cluster.

2.3. *Landmark Isomap*

We employ the nonlinear dimension reduction method Landmark Isomap for deriving cell landscapes which preserve the geometric features of the input data. Isometric feature mapping (Isomap) [18] is based on neighborhood graph construction and multidimensional scaling of

geodesic distances, with time complexity of $O(N^3)$. To improve the computing efficiency, we adapt the Landmark Isomap [14] to make it suitable for massive scRNA-seq datasets. When using n landmark points ($n \ll N$), it has a time complexity of $O(mnNlogN) + O(m^2N)$, where m is the number of the nearest neighbors for constructing the neighborhood graph. Here, we use the peaks and valleys as landmark points.

2.4. *Estimating pseudo-time*

We build the main cell trajectory by MST on the cluster centers in the isometric embedding. We then map the cells on the main cell trajectory to estimate the pseudo-time for each cell. The detailed steps are as the following:

1. Set a root node in the MST.
2. For each cell c_k, project it onto the nearest edge in the MST. Assume the nearest edge is $e_{i,j} = <v_i, v_j>$, v_i is closer to the root than v_j does. The projection vector $\overrightarrow{c_k c_k'}$ on the vector $\overrightarrow{e_{i,j}}$ can be expressed as $\overrightarrow{c_k c_k'} = \frac{\overrightarrow{c_k v_i} * \overrightarrow{c_k v_j}}{\|c_k v_i\| \|c_k v_j\|} \frac{\|c_k v_i\|}{\|c_k v_j\|} \overrightarrow{c_k v_j}$. The shortest distance of cell c_k to $e_{i,j}$ can be expressed as $d_{k,e_{i,j}} = \left\| \overrightarrow{c_k v_i} - \overrightarrow{c_k c_k'} \right\|$.
3. For each projection point c_k', calculate its distance to the root as the pseudo-time. The pseudo-time $t_c = Distance(root, v_i) + \|c_k' v_i\|$. Here, $Distance(root, v_i)$ is the length of the path from v_i to the root in the MST.

The time complexity of the pseudo-time estimation is O(N).

3. Results

To demonstrate the capability of LISA to build cell trajectory and estimate pseudo-time accurately, we evaluated it on one simulated dataset and two real datasets. The sizes of datasets range from several hundreds to tens of thousands. All of them contain true time labels. LISA identified cell trajectory and estimate pseudo-time for all datasets. We used the Spearman correlation coefficients between the true time labels and the estimated pseudo-time to assess the performance of LISA. Furthermore, we compared our results with two other state-of-the-art tools, Monocle2 and TSCAN.

3.1. *Datasets*

SLS3279 is a simulated dataset which contains 475 cells and 48 genes [19]. The time label ranges from 1 to 5 with continuous values. It contains two terminal lineages along with time.

The EMTAB dataset contains 1,529 cells from 88 human preimplantation embryos from E3 to E7 [15]. The processed Reads Per Kilobase of transcript per Million mapped reads (RPKM) values is downloaded from EMBL-EBI (https://www.ebi.ac.uk/). Here, we obtained 1,364 cells after filtering lowly represented cells using Seurat-1.4.1 [20]. We then used the 736 high variance genes from Petropoulos *et al*. [15]. The RPKM values were log2-transformed.

We also used 38,731 cells from zebrafish embryos across 12 developmental stages between 3.3-12 hours [2]. The raw dataset was processed by URD (https://github.com/farrellja/URD). The

processed data were normalized to Transcripts Per Million (TPM) values. The TPM values were then log2-transformed. There were 1,883 highly variable genes in the dataset.

We compare the performance of LISA, Monocle2, and TSCAN on cell trajectory reconstruction. We also compared the performance of LISA and Monocle2 on global pseudo-time estimation. In the latter scenario, TSCAN was not compared as it cannot generate global pseudo-time for all cells. We also compared all three methods for running time and memory usage.

3.2. *Simulation results*

First, we used the simulated dataset to verify the capability of LISA. In the simulated dataset, it contains two cell lineages. We did PCA for SLS3279, and all PCs were retained. The PCA result was input for t-SNE. Fig. 2A shows the cell clusters, peaks, and valleys that were derived from the t-SNE embedding by kNN-graph and hierarchical clustering described in the Methods section. The cell densities were shown in Fig. 2B. Correspondingly, it contains four peaks. Then we performed Landmark Isomap and built the MST of the cluster centers (Fig. 2C). We obtained the cell trajectory with two terminal lineages by setting cluster 1 as the root cluster (Fig. 2D).

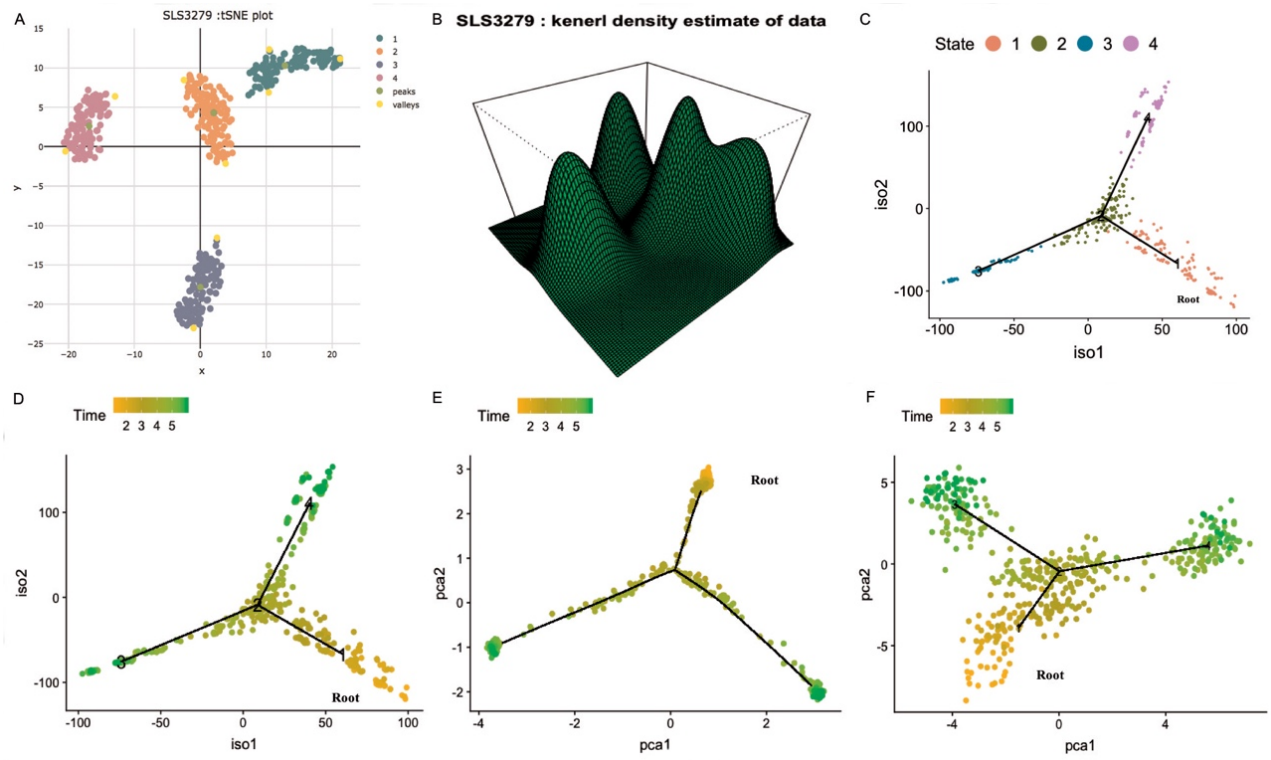


Figure 2. SLS3279 results. (A) The cell clusters, peaks, and valleys in the t-SNE embedding. (B) The cell density landscape. (C) The cell trajectory in the isometric embedding. (D)The cell trajectory reconstructed by LISA. (E) The cell trajectory reconstructed by Monocle2. (F) The cell trajectory reconstructed by TSCAN. In (D)-(F), the true time labels are shown.

For comparison, we applied Monocle2 and TSCAN to the simulated datasets. In the Monocle2 result, the cells were more concentrated at the ends of the branches (Fig. 2E). And the Spearman correlation coefficients between the estimated pseudo-time and true time labels were higher in LISA (0.97) than in Monocle2 (0.92). In the TSCAN result, the cells were more dispersed (Fig. 2F) and the global pseudo-time was not obtained. These results showed the potential of LISA in reconstructing cell trajectory and pseudo-time.

3.3. *Application to the EMTAB dataset*

We applied LISA to the EMTAB dataset which contains 1,364 cells [15]. It includes human preimplantation embryos cells developed into epiblast (EPI), primitive endoderm (PE) and trophectoderm (TE) cells from E3 to E7. The cell clusters, peaks and valleys were shown in Fig. 3A. The cell density plot was shown in Fig. 3B implying the complexity of cell clustering. We obtained 10 cell clusters. We then built the main cell trajectory in the isometric embedding (Fig. 3C). By setting cluster 9 as the root of cell trajectory, it clearly shows three terminal lineages in the cell differentiation path leading to cluster 5, 4, and 3, respectively. To understand the nature of the cell lineages, we used the 71 maker genes from EPI, PE and TE [15] to examine the genes expression patterns in different cell clusters (Fig. S1 in Appendix). It can be seen that cluster 5 is enriched for EPI marker genes, cluster 4 is enriched for PE marker genes, and cluster 3 is enriched for TE marker genes.

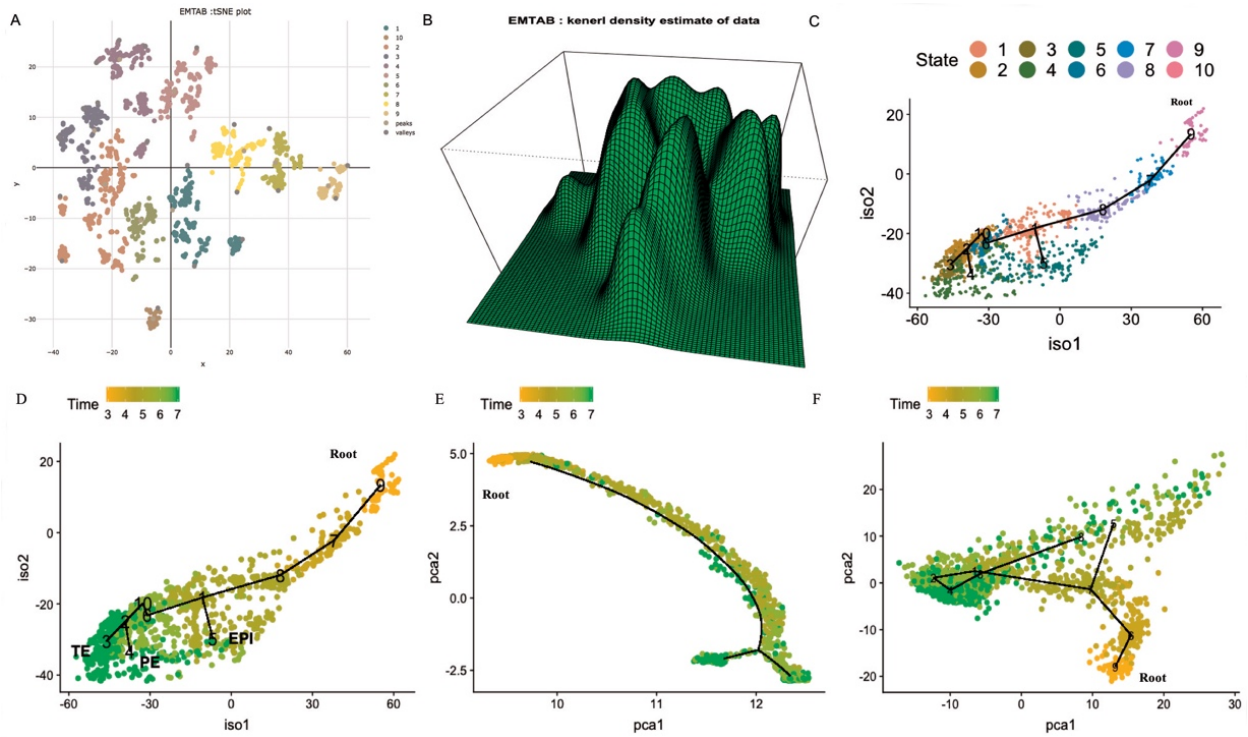


Figure 3. EMTAB results. (A) The cell clusters, peaks, and valleys in the t-SNE embedding. (B) The cell density landscape. (C) The cell trajectory in the isometric embedding. (D)The cell trajectory reconstructed by LISA. (E) The

cell trajectory reconstructed by Monocle2. (F) The cell trajectory reconstructed by TSCAN. In (D)-(F), the true time labels are shown.

As comparison, applying Monocle2 to the EMTAB dataset resulted in only two terminal lineages (Fig. 3E). Moreover, the Spearman correlation coefficients between the estimated pseudo-time and true time points were much higher in LISA (0.90) than in Monocle2 (0.77). The cell trajectory from TSCAN were shown in Fig. 3F, which also contains only two lineages.

3.4. *Application to the Zebrafish dataset*

We further applied LISA to a large zebrafish embryo differentiation dataset which contains 38,731 cells [2]. There are mainly three cell lineages including axial mesoderm, other mesendoderm, and ectoderm. In addition, it contains primordial germ and enveloping layer cells.

The cell clusters, peaks, and valleys of the zebrafish dataset are shown in Fig. 4A. The cell density plot is shown in Fig. 4B. We identified 27 cell clusters, peaks and valleys. We used the cell type marker genes [2] to investigate whether the main cell trajectories (Fig. 4C) are corresponding to known paths. As shown in Fig. S2 in Appendix, the endoderm marker genes were enriched in cluster 11 and 12. The primordial germ cell markers were enriched in cluster 1, 2 and 3. The enveloping layer cells (EVL) marker genes were enriched in cluster 4. The intermediate/lateral mesoderm marker genes were enriched in cluster 18, 24 and 25. The axial mesoderm marker genes were enriched in cluster 12 and 13. The paraxial mesoderm marker genes were enriched in cluster 19, 24 and 26. The pre-placodal ectoderm marker genes were enriched in cluster 21, 22, 26 and 27. The non-neural ectoderm marker genes were enriched in cluster 22, 23, 25, and 27. The hindbrain, fore/mid brain, neural crest and spinal cord marker genes were enriched in cluster 26 and 27. Based on the gene expression patterns, the cell lineage along cluster 11, 18, 12, and 13 was mainly corresponding to endoderm and axial mesoderm. The lineage along cluster 18, 20, 23, and 24 was mainly corresponding to intermediate/lateral mesoderm and paraxial mesoderm. The lineage along cluster 20, 21, 22, 25, 26, and 27 was corresponding to ectoderm which includes pre-placodal ectoderm, non-neural ectoderm, hindbrain, fore/mid brain, neural crest, and spinal cord. The lineage along cluster 1 was mainly corresponding to primordial germ cells. The lineage along cluster 4 was corresponding to EVL. Overall, the main cell trajectories reconstructed by LISA were consistent with those in Farrell *et al*. [2]. We set cluster 1 as the root of cell trajectory and estimated the pseudo-time of all cells. The Spearman correlation coefficients between the true time labels and the pseudo time reconstructed by LISA is 0.91.

As comparison, Monocle2 only generated one cell lineage (Fig. 4E). Furthermore, the pseudo-time reconstructed by Monocle2 is reverse to the true time labels resulting in a negative Spearman correlation coefficient. The TSCAN derived cell lineages were compressed and hard to be distinguished (Fig. 4F). Also, the cell lineages were not corresponding to Farrell *et al*. [2].

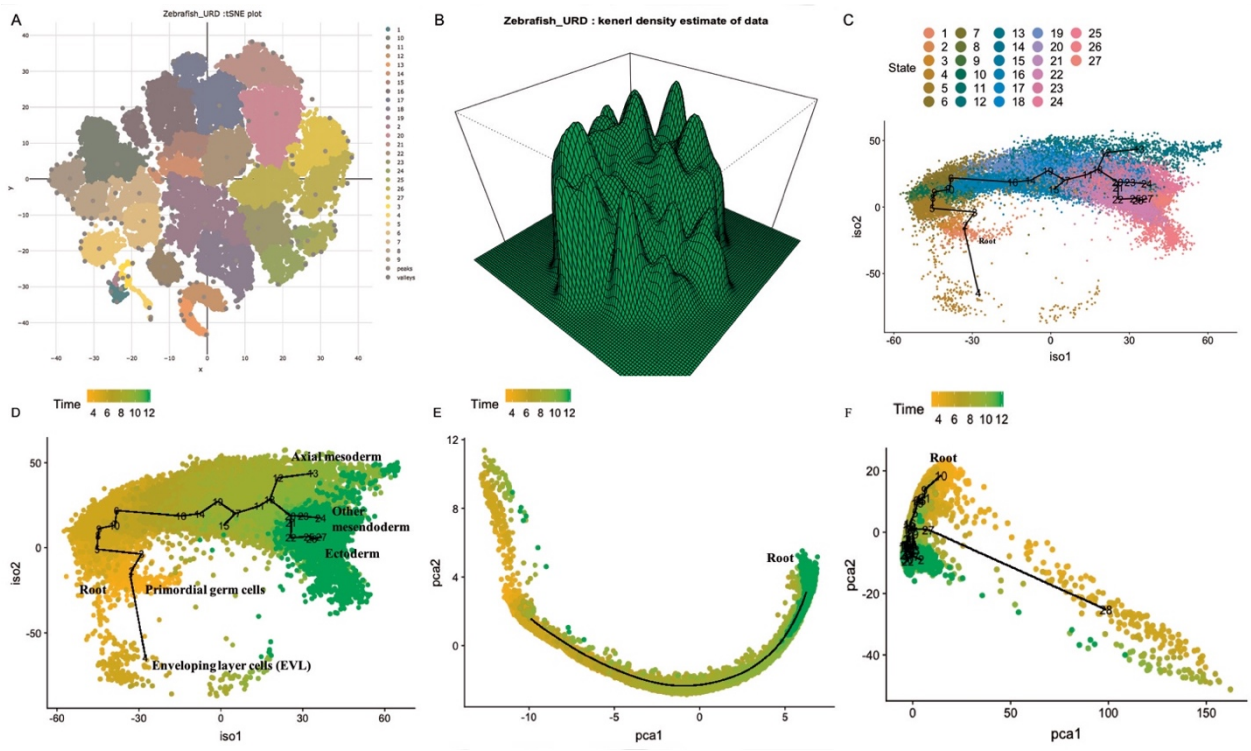


Figure 4. Zebrafish results. (A) The cell clusters, peaks, and valleys in the t-SNE embedding. (B) The cell density landscape. (C) The main cell trajectory in the isometric embedding. (D)The cell trajectory reconstructed by LISA. (E) The cell trajectory reconstructed by Monocle2. (F) The cell trajectory reconstructed by TSCAN. In (D)-(F), the true time labels are shown.

3.5. *Performance comparisons*

To estimate the pseudo-time of all cells, we set the root cluster based on the initial time point. In our comparisons, the clusters which contain the most numbers of cells at the initial time point were selected as the roots for both LISA and Monocle2. However, in the Zebrafish dataset, Monocle2 only found one lineage. In this case, the root cell was determined by Monocle2 automatically. The pseudo-time reconstructed by LISA was more consistent with the true time points than Monocle2 did (Fig. 5).

Overall, LISA showed better performance on reconstructing cell trajectory than Monocle2 and TSCAN did. Moreover, LISA used lower amount of computation time and required dramatically less memory than Monocle2 did (Fig. 6A-D). LISA used lower amount of computation time and memory than TSCAN did on the EMTAB dataset (Fig. 6A and C), and more computation time and similar memory usage compared to TSCAN on the Zebrafish dataset (Fig. 6B and D). In addition, in our tests, as cell number increases to exceed 50,000, Monocle2 was not able to estimate the pseudo-time, and TSCAN was not able to run its clustering procedure.

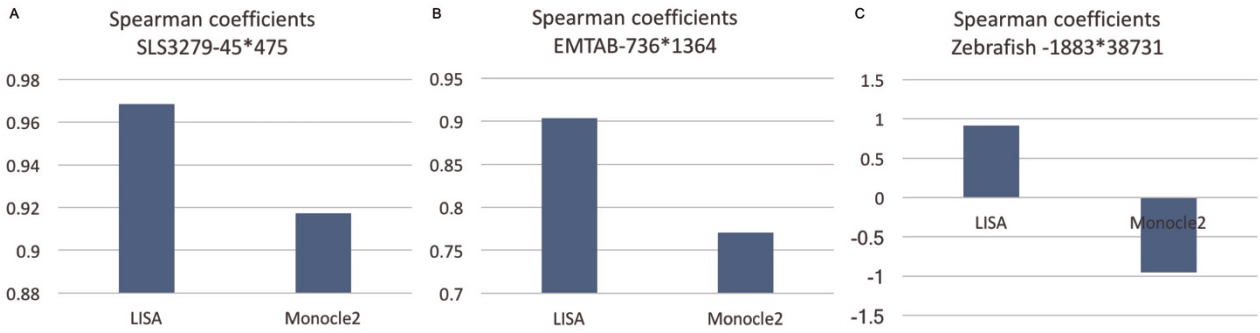


Figure 5. Comparing the Spearman correlation coefficients between the pseudo-time and the true time labels for different datasets using Monocle 2 and LISA. (A) SLS3279. (B) EMTAB. (C) Zebrafish.

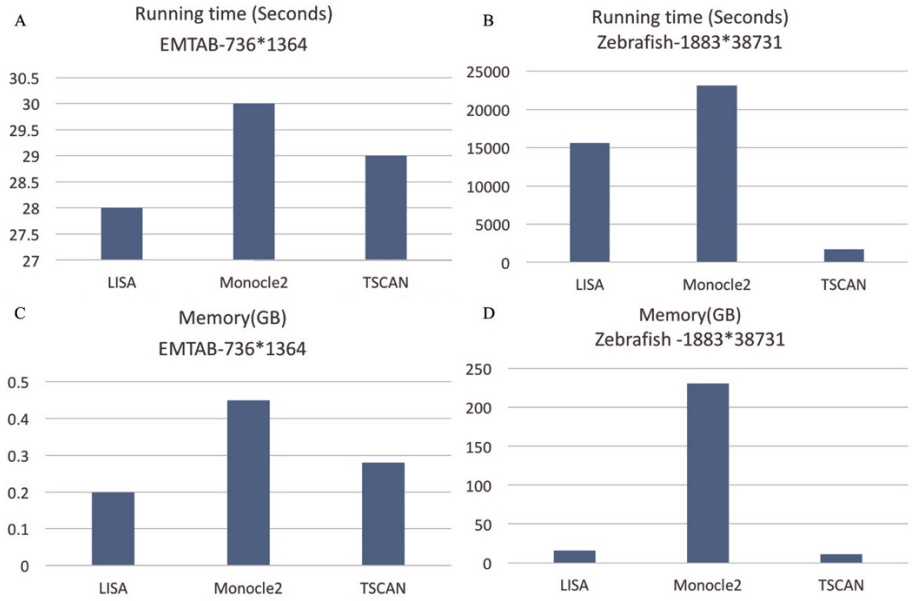


Figure 6. Computation time and memory usage of EMTAB and Zebrafish using LISA, Monocle2 and TSCAN. (A) The computation time on the EMTAB dataset. (B) The computation time on the Zebrafish dataset. (C) The memory usage of the EMTAB dataset. (D) The memory usage of the Zebrafish dataset.

4. Discussion

LISA is a new tool to reconstruct cell trajectory and pseudo-time of cells from scRNA-seq data. It uses kNN-graph and hierarchal clustering for identifying cell clusters, peaks, and valleys in the t-SNE embedding in an unsupervised way. It then uses the fast Landmark Isomap to derive the global geometrical structure of the data to estimate the main cell trajectory. Finally, it projects individual cells on the main cell trajectory and computes the global pseudo-time.

The assessments of cell trajectory and global pseduo-time reconstruction of LISA demonstrate its improved performance over existing methods such as Monocle2 and TSCAN. Meanwhile, LISA runs faster and requires less memory usage than Monocle2 does. In LISA, the root cluster can be set by the users for customized cell trajectory and pseudo-time analysis. Existing biological knowledge

of specific gene sets, e.g., known marker genes of cell types or states, can be used to reveal the biological meanings of the reconstructed cell lineages. In summary, LISA is an accurate, efficient, and flexible tool that can be broadly applied to massive scRNA-seq datasets.

5. Acknowledgments

We thank Disheng Mao and the Ouyang Lab members for discussions. This work is partially supported by the NIH/NIGMS R35 ESI MIRA Award and The Jackson Laboratory for Genomic Medicine faculty start-up fund (to ZO), and the Faculty Research Excellence Program Award at UConn (to YZ).

6. Appendix

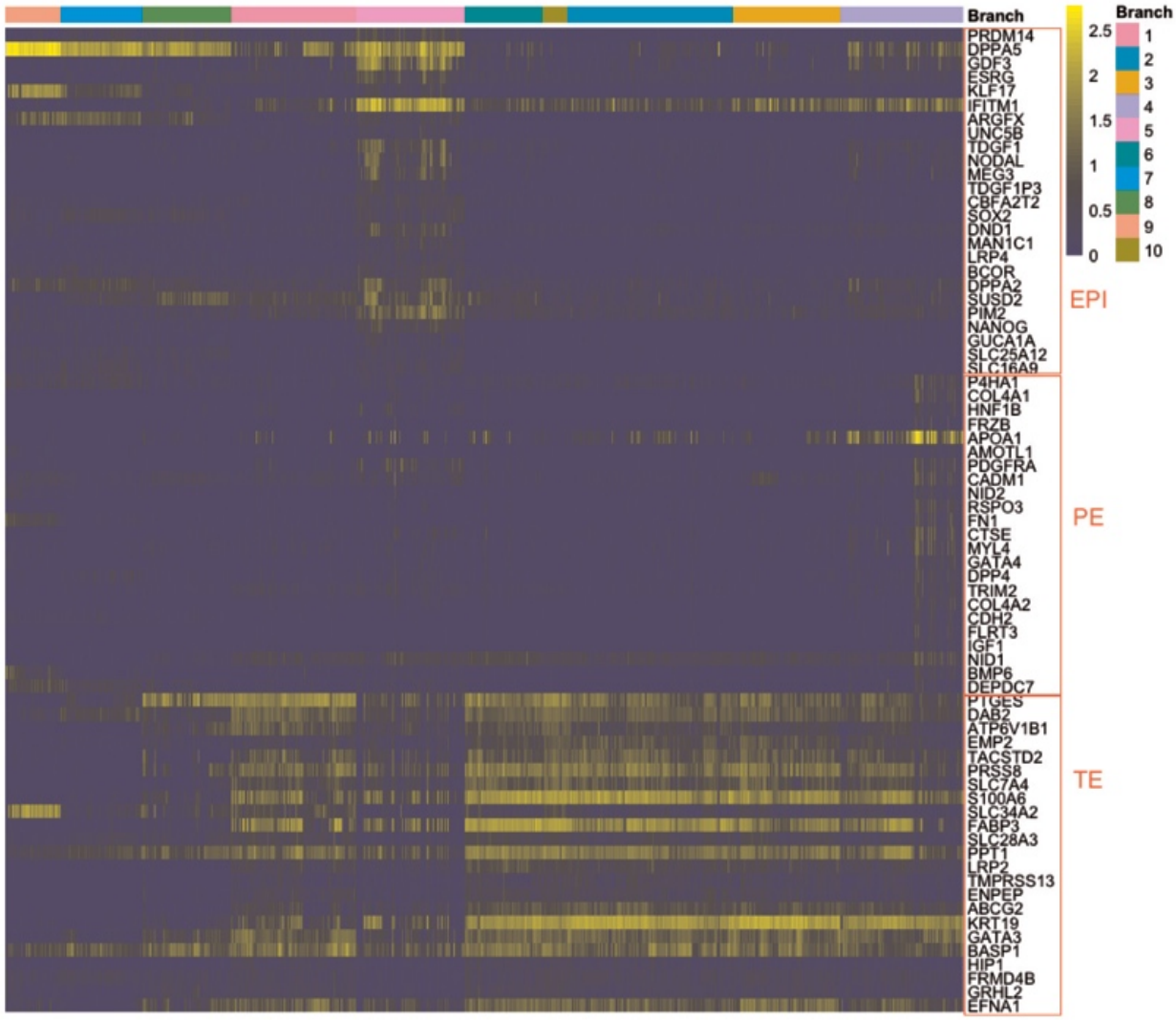


Figure S1. The gene expression heatmap of marker genes from three cell types (EPI, PE, TE). The branch names correspond to the cell clustering results.

Figure S2. The expression patterns of the marker genes of the 12 cell types in 27 clusters. (A) Enveloping layer cell and Periderm (B) Primodial germ (C) Axial mesoderm (D) Intermediate or lateral mesoderm (E) Paraxial Mesoderm (F) Endoderm (G) Neural cells (H) Spinal cord cells (I) Brain cells (J) Hindbrain cells (K) Non-neural ectoderm cells (L) Pre-placodal ectoderm.

References

1. Briggs, J. A., et al. *Science*. **360**,6392 (2018)
2. Farrell, J. A., et al. (2018). *Science*. **360**,6392 (2018)
3. Da Rocha, E. L., et al. *Nat. Commun*. **9**, 1 (2018)
4. Saelens, W., et al. *bioRxiv*. **276907** (2018)
5. Svensson, V., et al. *Nat Protoc*. 1**3,**4 (2018)
6. Jiang, L., et al. *Genome Biol*. 17, 1 (2016)
7. Butler, A., et al. *Nat Biotechnol*. **36**, 5 (2018)
8. Qiu, X., et al. *Nat Methods*. **14**, 10 (2017)
9. Ji, Z., et al. *Nucleic Acids Res*. **44**, 13 (2016)
10. Street, K., et al. *BMC genomic*.**19**, 1 (2018)
11. Mao, Q., et al. *Proceedings of the 21th ACM SIGKDD*. **15** (2015)
12. Zheng, G. X. et al. *Nat Biotechnol*. **34**, 3 (2016)
13. Arya S., et al. Journal of the ACM, 45 (1998)
14. Silva, V. D., et al. *In Advances in neural information processing systems*. (2003)
15. Petropoulos, S., et al. *Cell*. **165**, 4 (2016)
16. Maaten, L. V. D., & Hinton, G. *Journal of machine learning research*. **9**, 26 (2008)
17. Azzalini, A. a. M., et al. *arXiv*.**1301** (2013).
18. Tenenbaum, J. B, et al. *Science*. **290**, 5500 (2000)
19. Zwiessele, M., et al. *bioRxiv*. **057778** (2016)
20. Butler, A., et al. *Nat Biotechnol*. **36**, 5 (2018)

Topological Methods for Visualization and Analysis of High Dimensional Single-Cell RNA Sequencing Data

Tongxin Wang

Department of Computer Science, Indiana University Bloomington
Bloomington, Indiana, 47408, USA
Email: tw11@iu.edu

Travis Johnson

Department of Biomedical Informatics, Ohio State University
Columbus, Ohio, 43210, USA
Email: Travis.Johnson@osumc.edu

Jie Zhang

Department of Medical and Molecular Genetics, Indiana University School of Medicine
Indianapolis, Indiana, 46202, USA
Email: jizhan@iu.edu

Kun Huang

Department of Medicine, Indiana University School of Medicine
Indianapolis, Indiana, 46202, USA
Regenstrief Institute
Indianapolis, Indiana, 46202, USA
Email: kunhuang@iu.edu

Single-cell RNA sequencing (scRNA-seq) techniques have been very powerful in analyzing heterogeneous cell population and identifying cell types. Visualizing scRNA-seq data can help researchers effectively extract meaningful biological information and make new discoveries. While commonly used scRNA-seq visualization methods, such as t-SNE, are useful in detecting cell clusters, they often tear apart the intrinsic continuous structure in gene expression profiles. Topological Data Analysis (TDA) approaches like Mapper capture the shape of data by representing data as topological networks. TDA approaches are robust to noise and different platforms, while preserving the locality and data continuity. Moreover, instead of analyzing the whole dataset, Mapper allows researchers to explore biological meanings of specific pathways and genes by using different filter functions. In this paper, we applied Mapper to visualize scRNA-seq data. Our method can not only capture the clustering structure of cells, but also preserve the continuous gene expression topologies of cells. We demonstrated that by combining with gene co-expression network analysis, our method can reveal differential expression patterns of gene co-expression modules along the Mapper visualization.

Keywords: single-cell RNA sequencing; topological data analysis; Mapper

1. Introduction

Single-cell RNA sequencing (scRNA-seq) has provided an unprecedented view of heterogeneity in cell populations. While traditional bulk RNA-seq experiments quantify molecular states of cells by estimating mean expression profiles of millions of cells, scRNA-seq techniques can generate expression profiles of individual cells. Such improvement of resolution has made scRNA-seq a powerful tool to discover previously unknown cellular heterogeneity and functional diversity.[1]

However, the improvement of scRNA-seq techniques also provides new challenges in data analysis and interpretation. Firstly, the dimensionality of scRNA-seq data is very high. Typical scRNA-seq data usually contains RNA sequencing profile of over thousands of genes. Secondly, the number of cells is large. Recent high-throughput platforms are capable of generating data for thousands of cells. Thirdly, different scRNA-seq platforms and biological experiments may produce data with different biases or distributions, which introduces difficulty in comparing data across different platforms.

To address the aforementioned challenges, many computational tools have been developed to analyze and visualize high-dimensional scRNA-seq data, including Monocle,[2] Wishbone,[3] SMILE[4] and FVFC.[5] However, due to its advantage of detecting clusters in low dimensional space, t-distributed Stochastic Neighbor Embedding (t-SNE)[6] has become the most commonly used technique in scRNA-seq data visualization to identify cell type clusters.[7,8] However, cells in a population do not always form clustering structures. Oftentimes, they show continuous trajectories in space of gene expression profiles.[3] Therefore there is a need for a scalable method to capture such continuous gene expression topologies of cells.

Mapper[9] is a Topological Data Analysis (TDA) approach that extracts descriptions of high dimensional datasets in the form of simplicial complexes. As a method of representing data using topological networks, Mapper possesses several advantages when analyzing and visualizing scRNA-seq data. Firstly, similar to t-SNE, Mapper can preserve small-scale similarities among data points. However, while methods like t-SNE often tear apart the continuous structure in the original high dimensional space, Mapper can instead capture such continuous variation. Secondly, topological features are robust to small distortions of data, which makes Mapper robust to noise. Thirdly, Mapper captures the shape of the data by the distance functions chosen instead of depending on a specific coordinate system. Such coordinate-free approach gives Mapper the ability to compare data across different platforms.[10] Fourthly, Mapper produces a compressed representation of the shape of the dataset using a graph, where each node represents a cluster of data points. While t-SNE relies on approximation approaches[11] to scale to large datasets, Mapper is highly scalable to recent scRNA-seq datasets with large number of cells. Finally, Mapper can view data at multiple resolution.[9] This means that Mapper is able to discover patterns at different scales and capture details in large datasets with complex structures. Mapper has been applied to many biomedical problems, including identifying patient subsets in breast cancer,[12] analyzing murine embryonic stem cell (mESC) differentiation[13] and studying dynamical organization of the brain.[14]

In this paper, we used Mapper to visualize scRNA-seq data in order to extract different cell types and understand the lineage relationship among them. Our approach is innovative in the

following ways. Firstly, we visualize scRNA-seq data as combinatorial graphs through Mapper to capture topological features of the data. Mapper can visualize the continuous trajectory of cells over the space of gene expression profiles, which compliments the methods that recover the clusters of cells. Secondly, Mapper enables researchers to explore different biological meanings of scRNA-seq data by using different filter functions. In this paper, we took advantage of gene co-expression network analysis (GCNA) and focused on gene co-expression modules with biological functions. We further summarized gene modules into "eigengenes" and incorporated them into Mapper as filter functions or coloring of nodes. We applied our method on two large scRNA-seq datasets (melanoma and pancrease cell) and demontrated that our method can capture topological structures of scRNA-seq data. Combined with GCNA, Mapper also reveals that gene co-expression modules are differentially expressed between certain branches in the visualization and each is enriched with biological functions relevant to the corresponding cell types.

2. Methods

2.1. *Data*

In this paper, we applied our method on two large scRNA-seq datasets of melanoma tumor cells (GSE72056)[7] and human pancreas cells (GSE85241).[8] Details of datasets are summarized in Table 1 and both datasets can be accessed through NCBI Gene Expression Omnibus.

Table 1. Summary of datasets used in this study.

Dataset	Number of cells	Number of genes	Cell types(number of cells)
GSE72056	4645	23686	unresolved(132), malignant(1257) non-malignant(3256: T(2040), B(512), Macro(119), 62(Endo), CAF(56), NK(51), other(416))
GSE85241	2126	19126	acinar(219), alpha(812), beta(448), delta(193), ductal(245), endothelial(21), epsilon(3), mesenchymal(80), pp(101), unclear(4)

The expression level of gene i in cell j was quantified as $G_{ij} = log_2(TPM_{ij}/10 + 1)$, where TPM_{ij} is transcript-per-million (TPM) for gene i in cell j. In scRAN-seq, due to the low number of RNA transcriptomes, dropout events, where expression measurements of some random transcripts are missed as zeroes, often occur. To account for the dropout events, we filtered out genes with the lowest m_thr percent of mean expression level or the lowest v_thr percent of variance. We used $m\_thr = 95$ and $v\_thr = 95$ for the melanoma cell dataset and retained 775 genes after pre-processing. We used $m\_thr = 90$ and $v\_thr = 90$ for the pancreas cell dataset and retained 500 genes after pre-processing.

2.2. *Mapper*

Mapper, introduced by Singh et al.,[9] is one of the most commonly used TDA approaches. Mapper contains four steps: filtering, binning, clustering and graph generation and we reiterate them as Algorithm 1.

Algorithm 1 Mapper on scRNA-seq data

Input: a pre-processed gene expression matrix $\mathbf{G}$
Output: a graph *Grph* capturing topological features of $\mathbf{G}$
1. filtering: apply a filter function f on $\mathbf{G}$
2. binning: fragment the range of f into overlapping intervals and separate $\mathbf{G}$ into overlapping bins $\{B_1, B_2, ..., B_n\}$
3. clustering: apply hierarchical clustering on each bin and get a series of overlapping clusters $\mathbf{C}$
4. graph generation: create a graph *Grph* to capture the shape of $\mathbf{G}$ based on $\mathbf{C}$

Filtering step uses a filter function f to project gene expression data $\mathbf{G}$ to a lower dimensional space, usually $\mathbb{R}$ or $\mathbb{R}^2$. Different filter functions may generate networks with different shapes and researchers could view data from different perspectives by choosing different filter functions. One of the commonly used filter functions is eccentricity, which is a family of functions capturing the geometry of data. For cell $c_i \in \mathbf{G}$, given p with $1 \leq p < +\infty$, we define the eccentricity of c_i as

$$E_p(c_i) = (\frac{\sum_{c_j \in \mathbf{G}} d(c_i, c_j)^p}{N})^{1/p} \tag{1}$$

where $c_i, c_j \in \mathbf{G}$. $d(c_i, c_j)$ is the distance between c_i and c_j and N is the number of cells in $\mathbf{G}$. When $p = +\infty$, we define L_∞ eccentricity as $E_\infty(c_i) = \max_{c_j \in \mathbf{G}} d(c_i, c_j)$. L_∞ eccentricity has being used as a filter function to identify patient subtypes in breast cancer.[10] Dimension reduction methods such as Principle Component Analysis (PCA),[10] Multi-Dimensional Scaling (MDS)[13] and t-SNE[14] can also be used as filter functions. Researchers can also choose their own pre-computed data as filter functions.

After applying f on $\mathbf{G}$, range of f is fragmented into overlapping intervals $\mathbf{S} = \{S_1, S_2, ..., S_n\}$. The size of each interval is determined by several parameters: number of intervals n, fraction of overlap between adjacent intervals p and the interval generation method, which includes generating each interval with the same size or with the same number of cells. Cells in $\mathbf{G}$ are then put into a series of overlapping bins $\mathbf{B} = \{B_1, B_2, ..., B_n\}$ according to $\mathbf{S}$.

Hierarchical clustering is used to cluster cells in each bin B_i and researchers could choose from different distance metrics and linkage functions. A histogram is plot with the threshold values for each transition in the hierarchical clustering dendrogram and the number of clusters k_i is determined by the number of local maximas in the histogram.

After the clustering step, cells in $\mathbf{G}$ have been separated into a series of clusters $\mathbf{C} = \{C_{1,1}, C_{1,2}, ..., C_{1,k_1}, ..., C_{n,k_n}\}$. A graph *Grph* is constructed where each cluster $C_i \in \mathbf{C}$ is represented as a node and an edge is drawn between C_i and C_j if $C_i \cap C_j \neq \emptyset$. *Grph* is the output

of Mapper and can capture the topological features of the original data $\mathbf{G}$.

2.3. *Gene co-expression network analysis*

For GCNA, we applied local maximal Quasi-Clique Merger (lmQCM)[15] to identify densely connected modules such as quasi-cliques in weighted gene co-expression networks. Different from methods like WGNCA,[16] which partition genes into disjoint sets and do not allow overlap between clusters, lmQCM is a greedy approach that allows genes to be shared among multiple clusters. This is consistent with the fact that genes could participate in multiple biological processes. The lmQCM algorithm has four parameters: γ, α, t and β. γ determines if a new module can be initiated by setting the weight threshold for the first edge of the module, and has the largest influence on the result. We used $\gamma = 0.2$, $\alpha = 1$, $t = 1$ and $\beta = 0.4$ in our experiments. After identifying gene co-expression modules, we further summarized them into "eigengenes" by taking the first principle component of gene expression profiles of the modules.

We used ToppGene Suite[17] for gene set functional enrichment analysis to determine if gene modules detected by lmQCM are biologically meaningful. ToppGene finds biological annotations such as Gene Ontoogy (GO) items that are significant in a set of genes. To provide meaningful results, we only performed functional enrichment analysis on gene modules that contain at least 10 genes and at most 500 genes.

2.4. *Visualizing networks*

The output of Mapper on scRNA-seq data is a network where each node is a cluster of cells and each edge means that two clusters share some common cells. We used a force directed layout algorithm to calculate the position of each node, which means the positions of individual nodes do not have particular meanings and only the connections between nodes are informative.

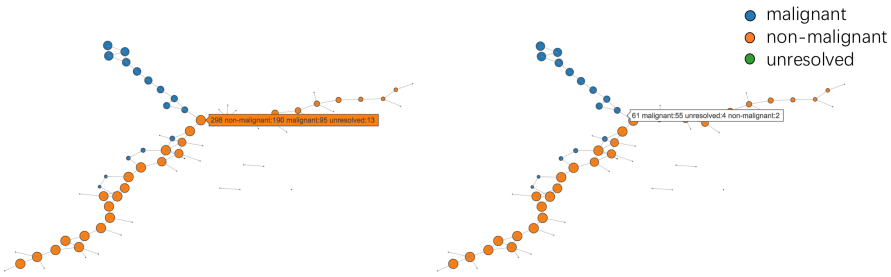


Figure 1. Cursor hovering for detailed information: hovering over a node (left) and hovering over an edge(right).

Each node contains several features of the cluster it represents. The size of a node is proportional to the number of cells in the node. The color of each node represents a specific property of cells, which could be determined by users. For quantitative features, such as the expression level of a gene or an eigengene, mean value is used to represent the cluster. For categorical features, such as types of cells, the majority category is used to represent

the cluster. Pie charts is another option to visualize the category composition of the nodes, but it could clutter the visualization, making perception of composition difficult. However, to compensate the information loss by using the majority as representation, we utilized an interactive visualization technique that allows users to get the cell type composition of a node or an edge by hovering over it. An example of this is shown in Figure 1.

3. Results

3.1. *Visualizing melanoma cells using Mapper*

We first compared Mapper with several commonly used dimensionality reduction algorithms (t-SNE,[6] PCA, Isomap,[18] LLE[19] and Spectral Embedding[20]) by visualizing the melanoma cell dataset and the results are shown in Figure 2. We also compared Mapper with one of the state-of-the-art scRNA-seq visualization methods, Monocle 2.[2] Each node in the Mapper visualization represents a cluster of cells while each point in other visualizations represents a cell.

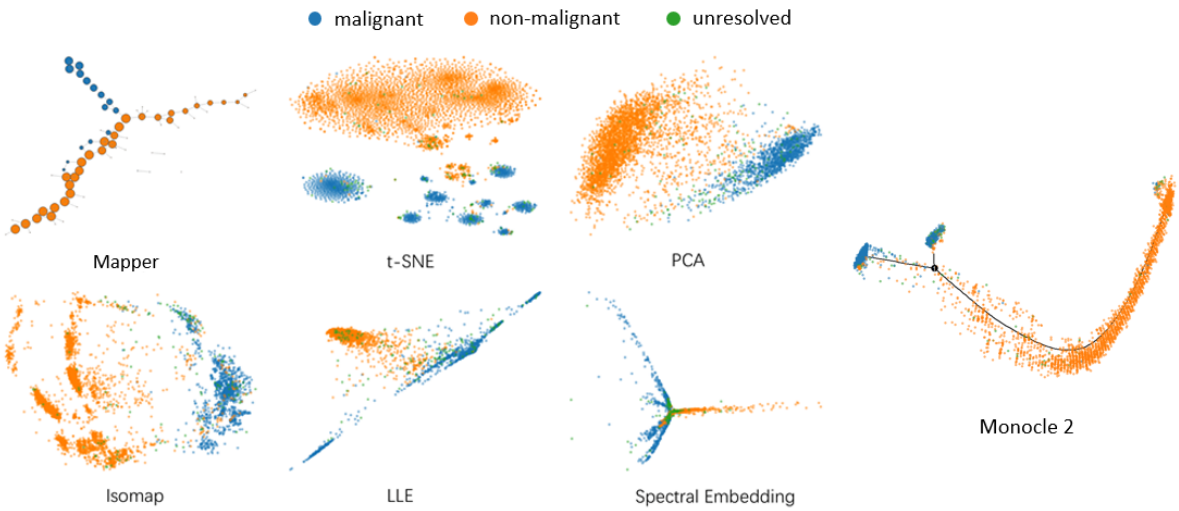


Figure 2. Visualization of melanoma cells.

We observe that all above algorithms are capable of separating malignant cells from non-malignant ones. Particularly, t-SNE separates malignant cells from different tumors into different clusters, which implies that t-SNE may be influenced by batch effects of different cell populations besides differentiating malignant and non-malignant cells. This also suggests that t-SNE often tends to break the continuous trajectory of cells in the space of gene expression profiles. On the other hand, by visualizing the shape of the data, Mapper not only separates malignant cells from non-malignant cells, but also preserves the continuous structure in scRNA-seq data by visualizing malignant cells as a branch separating from non-malignant cells. Monocle 2 also provides an interesting visualization, where non-malignant cells branch out into two clusters of malignant cells. However, further analysis did not find different patterns in expression levels of gene co-expression modules between the two malignant cell clusters.

Another advantage of Mapper is that it can view data under different resolutions and capture patterns of different scales. Figure 3 shows a series of visualizations of melanoma cell dataset with different number of bins (*nbins*) in the binning step. We observe that the graph representation of the data is coarse when the number of bins is low, which captures the global structure of the data. As the number of bins increases, more detailed structures are revealed and we can detect patterns at a higher resolution.

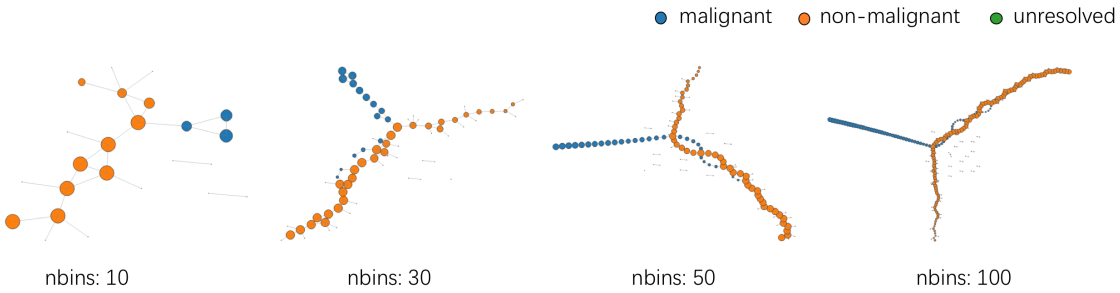


Figure 3. Mapper visualization of melanoma cells with different number of bins.

Moreover, we could still take the advantage of t-SNE within the Mapper framework by using t-SNE as the filter function. Using t-SNE as the filter function can produce a compressed representation that captures the clustering structure of the t-SNE visualization.

3.2. *Using eigengenes for node coloring in Mapper*

GCNA can identify gene co-expression modules with potential biological meanings, which helps the interpretation of our visualizations. One way to utilize information from GCNA is to use expression profiles of eigengenes to color the nodes in graphs produced by Mapper, as shown in Figure 4.

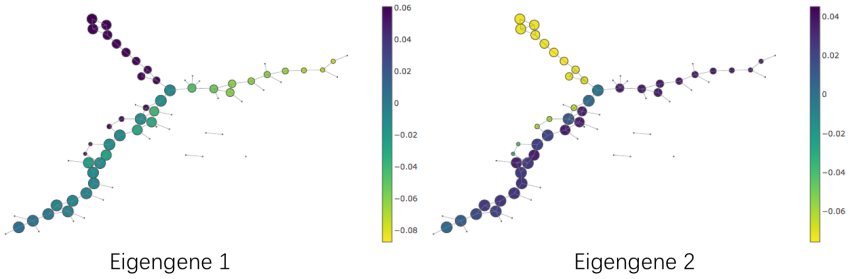


Figure 4. Mapper visualization of melanoma cells with coloring of eigengene expression profiles.

Two gene co-expression modules were identified in the melanoma dataset by applying lmQCM on the pre-processed scRNA-seq data. Gene set enrichment analysis results with false discovery rate corrected p values generated by ToppGene Suite are summarized in Table 2. Figure 4 shows obvious difference of the two eigengene expression profiles between the malignant branch and non-malignant cells. This is consistent with the fact that biological

processes such as cell activation, immune response and regulation of cell migration are strongly associated with malignancy of cells.

Table 2. Gene co-expression modules in the melanoma dataset.

Module ID	Number of genes	Enriched GO items (p value)
1	26	GO:0001775 cell activation (1.983E-12) GO:0006955 immune response (1.983E-12) GO:0045321 leukocyte activation (1.983E-12)
2	16	GO:0042470 melanosome (7.681E-7) GO:0030334 regulation of cell migration (5.356E-4) GO:2000145 regulation of cell motility (5.356E-4)

We further investigated genes in eigengene 2 and two non-overlapping sub-modules were discovered. One contains five genes (TYR, CTSB, MLANA, GPNMB, PMEL) which enriches with proteins associated with melanosome - a structure associated with melanocytes and potentially melanoma. The other contains seven genes (TIMP1, TMSB4X, SGK1, GSN, LGALS3, SERPINE2, APOD) and enriches with regulation of cell migration and extracellular matrix. Figure 5 shows that genes in both sub-modules have lower expression level in malignant cells than non-malignant cells, which indicates functions related to normal melanosome and cell migration activities may be disrupted in malignant melanoma cells.

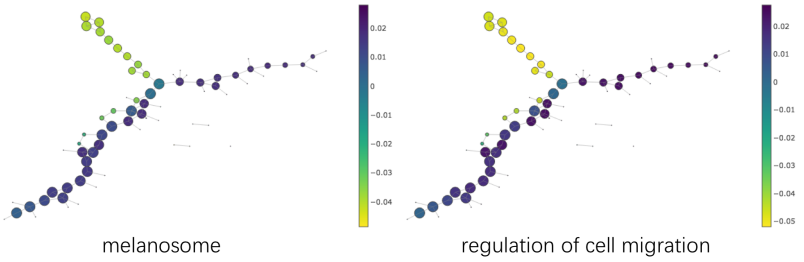


Figure 5. Using eigengene expression profiles of two sub-modules in eigengene 2 as coloring of nodes. The first sub-module is enriched in melanosome proteins and the second sub-module is enriched in cell migration and extracellular space.

3.3. *Using eigengenes as filter functions in Mapper*

By using different filter functions, researchers can rapidly explore different biological hypotheses in scRNA-seq data through Mapper. So, we can also incorporate GCNA into Mapper by using expression profiles of eigengenes as filter functions. Figure 6 shows L_∞ eccentricity, a

commonly used filter function, fails to separate different types of human pancreas cells. On the other hand, t-SNE completely separate different cell types into different clusters. Since similarities between points with long distances are not reliable in t-SNE visualization, we are not able to investigate the relationships between different cell types through t-SNE. By using the expression level of eigengene 2 as a filter function, Mapper can separate different types of pancreas cells with a branch-shape visualization, which preserves the continuity of cells at the same time. More specifically, the exocrine compartment of pancreas, including acinar cells and ductal cells, is visualized as a branch separating from the endocrine compartment. The shape of the visualization is consistent regardless of the linkage function in the clustering step (single or complete linkage). Enrichment analysis shows that eigengene 2 is associated with delta cells and PP cells of mouse adult pancreas in co-expression atlas, which indicates that eigengene 2 may contain genes conserved between species. This suggests that the eigengene 2 is worthy of further investigation for deeper understanding in pancreatic biology.

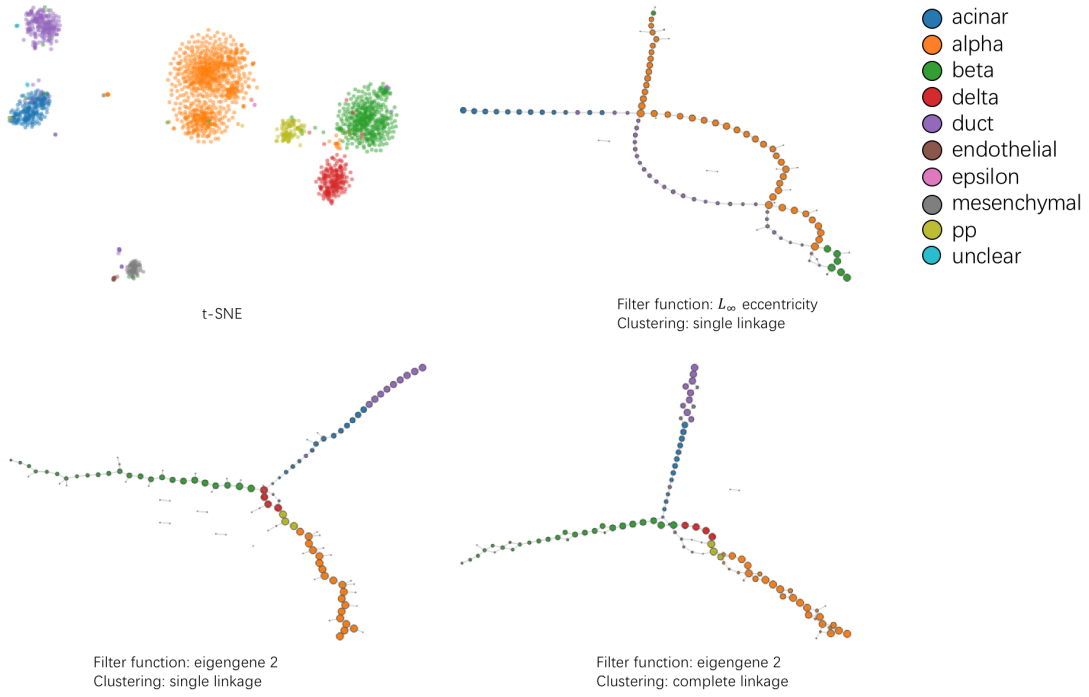


Figure 6. Visualization of pancreas cells using t-SNE and Mapper.

Moreover, we can combine multiple eigengene profiles as filter functions. From Figure 7, we observe that using a single eigengene as the filter function can only differentiate some of the cell types in the melanoma cell dataset. However, combining two eigenegenes as the filter function can further differentiate different types such as macrocells and endothelial cells. Comparing to t-SNE, Mapper visualization using two eigenegenes not only preserves the similarities between B cells and T cells, but also reduces the batch effect by visualizing all malignant cells as a group of tightly connected clusters.

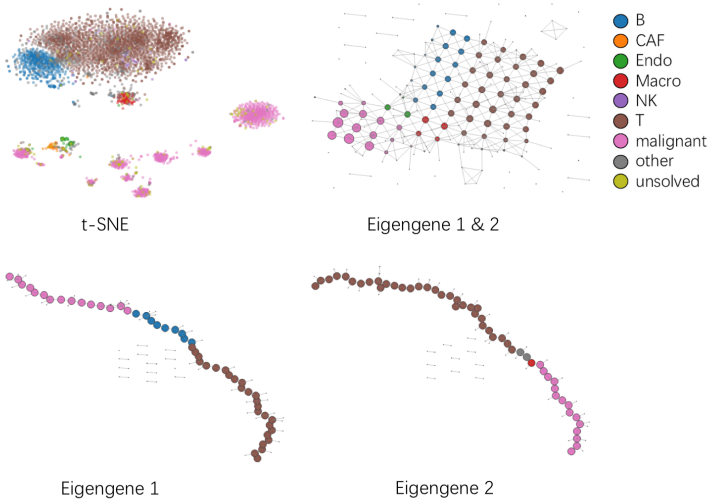


Figure 7. Visualization of melanoma cells using t-SNE and Mapper with eigengene expression profiles as filter functions.

3.4. *Mapper reveals potential functional relationships between exocrine cells in pancreas*

To further investigate the biological significance of Mapper visualization, we used the expression levels of established marker genes for each of the six main pancreatic cell types in the human pancreas cell dataset to color the nodes in our visualization. From Figure 8, we observe that expression levels of marker genes in endocrine cells show significant difference in the corresponding cell types. However, KRT19 and PRSS1 could not well separate ductal cells and acinar cells in the exocrine branch, which indicates potential relationships within exocrine cells. We further applied GCNA on ductal cells and acinar cells separately, as well as combined together. Two gene co-expression modules were identified across all three cell populations. However, as shown in Figure 9, module 1 in the combined cell population shows very small overlap with all the gene modules identified from the ductal-only and acinar-only population. Enrichment analysis shows that module 1 in the combined cell population is associated with neuron part (GO:0097458, p = 1.141E-3) and extracellular space (GO:0005615, p = 5.735E-3), which could relate to enzymes production activities of acinar cells. Module 1 also enriches secretory granule (GO:0030141, p = 7.314E-3), which could relate to the production of bicarbonate-rich secretion in ductal cells.

4. Conclusion

The scRNA-seq technology is becoming a common approach to study cellular heterogeneity and dynamic cellular process. Visualization techniques can help researchers effectively extract that information from scRNA-seq data. In this paper, we applied a TDA algorithm, Mapper, on two large scRNA-seq datasets. We showed that Mapper is able to preserve the continuous structure in gene expression profiles while effectively differentiate different cell types at the same time. This advantage allows us to investigate the relationships and connections between

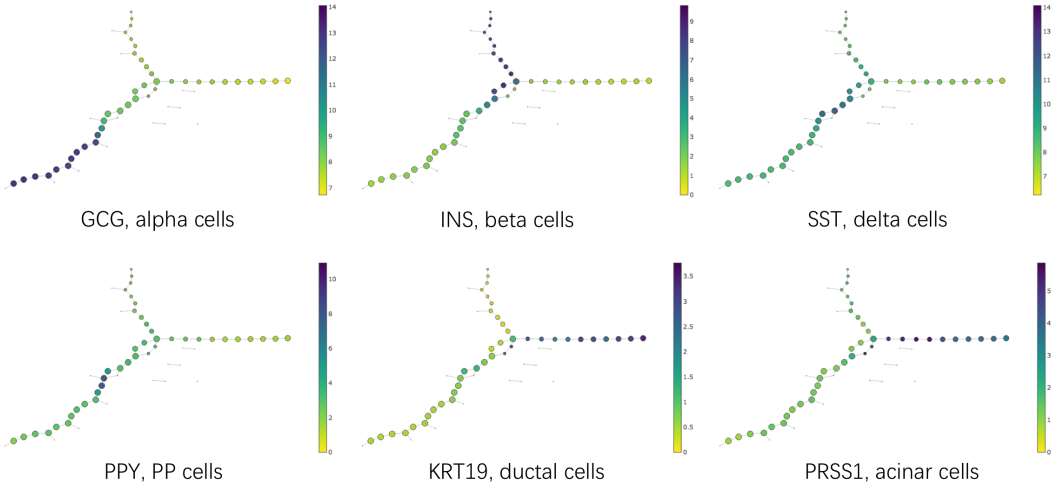


Figure 8. Mapper visualization of pancreas cells, with coloring of marker genes expression levels.

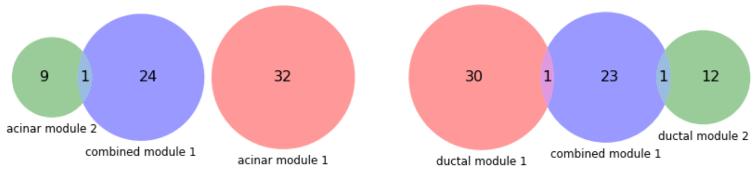


Figure 9. Gene co-expression module detected only in combined cell population of acinar and ductal cells.

different cell types through visualization. Mapper also allows researchers to explore different biological hypotheses through different filter functions and generates results with rich biological information. We took this advantage by incorporating information from GCNA into our visualization. GNCA helps to differentiate different cell types more effectively and enrichment analysis of gene co-expression modules helps the interpretation of the visualization results. Moreover, our method provides various options for researchers to explore the data from different perspectives and is highly scalable to large number of cells.

While our method shows potential in effectively extracting biological insights from scRNA-seq data, some limitations still exists. Firstly, although different filter functions could produce networks with different structures, allowing researchers to explore data from different perspectives, not all filter functions could generate networks with meaningful shapes. Researchers need to work with the data experimentally in order to find informative visualizations. Secondly, enrichment analysis only provides preliminary results of potential biological significance and more rigorous experiments are needed to validate the findings. Finally, we plan to implement our method as a web tool so that more researchers can easily access our method.

Acknowledgements

This work is partially supported by IUSM startup fund, the NCI ITCR U01 (CA188547) and Data Science and Bioinformatics Program for Precision Health Initiative, Indiana University.

References

1. E. Shapiro, T. Biezuner and S. Linnarsson, *Nature Reviews Genetics* **14**, p. 618 (2013).
2. X. Qiu, Q. Mao, Y. Tang, L. Wang, R. Chawla, H. A. Pliner and C. Trapnell, *Nature methods* **14**, p. 979 (2017).
3. M. Setty, M. D. Tadmor, S. Reich-Zeliger, O. Angel, T. M. Salame, P. Kathail, K. Choi, S. Bendall, N. Friedman and D. Pe'er, *Nature biotechnology* **34**, p. 637 (2016).
4. B. Wang, J. Zhu, E. Pierson, D. Ramazzotti and S. Batzoglou, *Nature methods* **14**, p. 414 (2017).
5. Z. Han, T. Johnson, J. Zhang, X. Zhang and K. Huang, *BioMed research international* **2017** (2017).
6. L. v. d. Maaten and G. Hinton, *Journal of machine learning research* **9**, 2579 (2008).
7. I. Tirosh, B. Izar, S. M. Prakadan, M. H. Wadsworth, D. Treacy, J. J. Trombetta, A. Rotem, C. Rodman, C. Lian, G. Murphy *et al.*, *Science* **352**, 189 (2016).
8. M. J. Muraro, G. Dharmadhikari, D. Grün, N. Groen, T. Dielen, E. Jansen, L. van Gurp, M. A. Engelse, F. Carlotti, E. J. de Koning *et al.*, *Cell systems* **3**, 385 (2016).
9. G. Singh, F. Mémoli and G. E. Carlsson, Topological methods for the analysis of high dimensional data sets and 3d object recognition., in *SPBG*, 2007.
10. P. Y. Lum, G. Singh, A. Lehman, T. Ishkanov, M. Vejdemo-Johansson, M. Alagappan, J. Carlsson and G. Carlsson, *Scientific reports* **3**, p. srep01236 (2013).
11. L. Van Der Maaten, *The Journal of Machine Learning Research* **15**, 3221 (2014).
12. M. Nicolau, A. J. Levine and G. Carlsson, *Proceedings of the National Academy of Sciences* , p. 201102826 (2011).
13. A. H. Rizvi, P. G. Camara, E. K. Kandror, T. J. Roberts, I. Schieren, T. Maniatis and R. Rabadan, *Nature biotechnology* **35**, p. 551 (2017).
14. M. Saggar, O. Sporns, J. Gonzalez-Castillo, P. A. Bandettini, G. Carlsson, G. Glover and A. L. Reiss, *Nature communications* **9**, p. 1399 (2018).
15. J. Zhang and K. Huang, *Cancer informatics* **13**, CIN (2014).
16. P. Langfelder and S. Horvath, *BMC bioinformatics* **9**, p. 559 (2008).
17. J. Chen, H. Xu, B. J. Aronow and A. G. Jegga, *BMC bioinformatics* **8**, p. 392 (2007).
18. M. Balasubramanian and E. L. Schwartz, *Science* **295**, 7 (2002).
19. S. T. Roweis and L. K. Saul, *science* **290**, 2323 (2000).
20. U. Von Luxburg, *Statistics and computing* **17**, 395 (2007).

Parameter tuning is a key part of dimensionality reduction via deep variational autoencoders for single cell RNA transcriptomics

Qiwen Hu
Department of Systems Pharmacology and Translational Therapeutics
University of Pennsylvania,
Philadelphia, PA 19104, USA
Email: huqiwen0313@gmail.com

Casey S. Greene
Department of Systems Pharmacology and Translational Therapeutics
University of Pennsylvania,
Philadelphia, PA 19104, USA
Email: csgreene@upenn.edu

Single-cell RNA sequencing (scRNA-seq) is a powerful tool to profile the transcriptomes of a large number of individual cells at a high resolution. These data usually contain measurements of gene expression for many genes in thousands or tens of thousands of cells, though some datasets now reach the million-cell mark. Projecting high-dimensional scRNA-seq data into a low dimensional space aids downstream analysis and data visualization. Many recent preprints accomplish this using variational autoencoders (VAE), generative models that learn underlying structure of data by compress it into a constrained, low dimensional space. The low dimensional spaces generated by VAEs have revealed complex patterns and novel biological signals from large-scale gene expression data and drug response predictions. Here, we evaluate a simple VAE approach for gene expression data, Tybalt, by training and measuring its performance on sets of simulated scRNA-seq data. We find a number of counter-intuitive performance features: i.e., deeper neural networks can struggle when datasets contain more observations under some parameter configurations. We show that these methods are highly sensitive to parameter tuning: when tuned, the performance of the Tybalt model, which was not optimized for scRNA-seq data, outperforms other popular dimension reduction approaches – PCA, ZIFA, UMAP and t-SNE. On the other hand, without tuning performance can also be remarkably poor on the same data. Our results should discourage authors and reviewers from relying on self-reported performance comparisons to evaluate the relative value of contributions in this area at this time. Instead, we recommend that attempts to compare or benchmark autoencoder methods for scRNA-seq data be performed by disinterested third parties or by methods developers only on unseen benchmark data that are provided to all participants simultaneously because the potential for performance differences due to unequal parameter tuning is so high.

Keywords: Single Cell; Variational Autoencoder; Dimensionality Reduction; Latent Spaces.

1. Introduction

Single-cell RNA sequencing (scRNA-seq) profiles the transcriptomes of individual cells [1], allowing researchers to study heterogeneous cell characteristics and responses [2, 3]. Due to the small amount of RNA captured in each cell as well as technical factors related to capture efficiency, scRNA-seq data have a high dropout rate (many genes have no measured expression in each cell).

Researchers often analyze these data by projecting cells into a low dimensional space, which enables downstream analysis such as imputation of missing measurements and visualization.

Widely used approaches include the linear principal component analysis (PCA) [4], which doesn't take dropout into account, and ZIFA [5], which uses zero-inflated factor analysis to model the dropout events and do dimension reduction. The t-distributed stochastic neighbor embedding (t-SNE) method is also widely used [6]. This method uses local structure, but it is time consuming for large datasets and has been reported to be highly sensitive to hyperparameters [7]. The recently proposed Uniform Manifold Approximation and Projection (UMAP) [8] method attempts to address these limitations by preserving more global structure and as much local structure as t-SNE. These approaches do not model the dropout characteristic of scRNA-seq data.

Deep generative neural network models can learn low-dimensional representations from large amounts of unlabeled data and have been successfully applied to many domains, such as image and text generation [9]. Variational autoencoders (VAE) learn this representation by compressing data into a constrained, low-dimensional space [10]. VAEs have been used in biology to analyze large-scale gene expression data and drug response predictions [11, 10]. In recent months, preprints proposing numerous deep neural network models for scRNA-seq data have been posted. Grønbech et al. [12] proposed a Gaussian-mixture VAE model for raw counts from scRNA-seq data and found the model can learn biologically groupings of scRNA-seq dataset. Eraslan et al. developed a deep count autoencoder based on zero-inflated negative binomial noise model for data imputation [13]. Lopez et al. developed single-cell Variational Inference (scVI) based on hierarchical Bayesian models, which can be used for batch correction, dimension reduction and identification of differentially expressed genes [14]. Deng et al. propose an autoencoder that includes a feedback step after zeroes are imputed [15]. These methods often report performance, but while many report hyperparameter selections, few describe how those parameters were reached.

In this work, our goal was to understand the extent to which reported performance of the neural network methods was due modifications for scRNA-seq data. We applied a straightforward VAE developed for bulk gene expression data, Tybalt [10], to simulated and real scRNA-seq data under various parameter settings. Some performance characteristics, including a decrease in performance when the number of examples was increased, suggest substantial sensitivity to hyperparameters. We sought to optimize parameters and adjust the dimensionality of the model to rescue performance. In our prior work from *PSB 2015* using autoencoders for the analysis of bulk gene expression data, performance was relatively stable over many parameter values [16]. In contrast, the performance of the standard VAE, Tybalt, changes from dismal to better than other popular dimension reduction approaches – PCA, ZIFA, UMAP and t-SNE – with only modest parameter tuning.

These results should guide the reporting of new methods. First, it is critically important that reviewers expect manuscripts in this area to report the extent to which hyperparameters affect performance across multiple datasets. Second, manuscripts reporting new techniques should be evaluated both on theoretical grounding as well as empirical results. Because results can be changed easily by light tuning, self-reported performance numbers may provide only weak evidence. Third, assessments and benchmarking should be done by disinterested parties with a realistic amount of parameter tuning or should be performed by first parties on datasets for which the labels are not revealed until after predictions are made.

2. Methods

2.1. *Data Simulation*

We simulated scRNA-seq data using Splatter [17]. We used the default simulation parameters provided by Splatter to generate synthetic scRNA-seq data with variable numbers of genes, cell types, cells, outliers, etc. We simulated data with variable numbers of cells (ncell: 500 - 5000), genes (nGenes: 20000 - 60000), cell types (nGroups: 5 – 15) and probabilities of expression outliers (outlier: 0.1 – 0.5). In total, we generated 40 simulated single-cell datasets. We normalized the raw count matrix by TPM (Transcripts Per Kilobase Million).

2.2. *Model Structure and Training*

VAEs model the distribution P(X) of data in a high dimensional space from a low dimensional latent space z. VAEs consist of two connected neural networks: the encoder and decoder. Data are compressed by the encoder and reconstructed by the decoder. The variation probability Q(z|X) is used to approximate the posterior distribution P(z|X), which is then optimized to minimize the Kullback–Leibler (KL) divergence between Q(z|X) and P(z|X) and reconstruction loss [18, 19]. A baseline model for gene expression data, termed Tybalt and which we use here, was described in [10]. The encoder was a multi-layer (varied from 0 to 2) neural network. The representative latent space z was sampled from a Gaussian distribution $q_\theta(z|X)$, with mean and variance generated by the encoder network. The learned latent space z was used to re-generate the count matrix X' by the decoder, which was also a multi-layer neural network (from 0 to 2) (Figure 1). For the first stage, we trained Tybalt with three structures: a one-layer model with a gene-wise TPM vector connected to 20 latent features and then reconstructed output; a two-layer model with the TPM vector encoded into a 100-node hidden layer, then the 20 latent features, then a 100-node hidden layer, and then the reconstructed output; and a three-layer model which contains two 100-node hidden layers. The model was built in Keras (version 2.0.6) [20] with a TensorFlow (Version 1.0.1) backend [21].

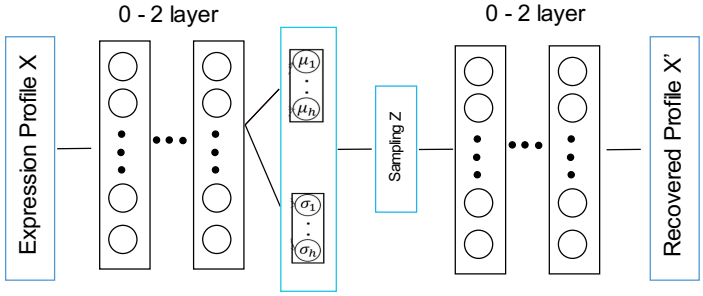


Figure 1: Overview of the structure of variation autoencoder. The model consists an encoder network and a decoder network, both of them are designed as 0-2 layers fully connected neural networks.

2.3. *Parameter Tuning*

We tuned parameters using a grid search over batch size (50, 100, 200), epochs (25, 50, 100, 200), neural network depth (2, 3) and, for models with two or more layers, the dimensionality of the first layer (100, 250, 500). Simulated data were partitioned into training and test data, with

the test set being 10% of the full data. For real data, we selected three single cell datasets with author-assigned cell type labels [25 – 27]. We downloaded count matrices from the Hemberg Group repository of data (https://hemberg-lab.github.io/scRNA.seq.datasets/). We zero-one normalized the count matrix before training the VAE.

2.4. *Performance Measurement*

We used three evaluation metrics to measure performance: 1) *k*-means based 2) *k*NN based 3) average silhouette score. The *k*-means based and *k*NN based measurements measure how well the low-dimensional space allows simple methods to recover simulated cell types. The average silhouette score measures the extent to which clusters are separable in the latent space. An ideal method is accurate and produces separable clusters.

2.4.1. *k-means performance assessment*

In the *k*-means based evaluation we performed iterative *k*-means clustering on the low-dimensional latent space. We compared the predicted clustering results with the known cell types in the simulated data. We performed *k*-means clustering for 50 times to get a stable measurement and – to evaluate a best-case scenario – we set the number of clusters, *k*, to the number of true cell types in the data. We assessed methods by the normalized mutual information (NMI) [22], between cell types and the known categories as well as the adjusted rand index (ARI) [23].

2.4.2. *kNN performance assessment*

For the *k*NN evaluation, we used *k* nearest neighbors to predict cell type from latent space distances and assessed performance by 5-fold cross validation within the simulated dataset. To more closely replicate how methods are used in practice, the model was tuned within only the training data by a sweep over the neighbor number parameter with 3-fold cross validation. We assessed performance using accuracy, precision, recall and f-score, but report only accuracy due to space constriants.

2.4.3. *Average silhouette score performance assessment*

We used the silhouette score [24] to measure the extent to which simulated cell type clusters are internally close in the latent space but separated from other cell types. The silhouette value is between -1 to 1. A silhouette value of 1 indicates that the data point is of distance zero from other points of the same type, while one of -1 indicates that the point is distance zero from all points of a different cluster but some distance from points of the same cluster. Average silhouette score over all points then indicates how separable each cell type is in the latent space.

3. Results

3.1. *The performance of multiple methods on simulated data*

We tested the performance of five dimension reduction approaches: Tybalt, ZIFA, UMAP, t-SNE and PCA using the three different evaluation metrics over the data simulated by Splatter [17] as described in the Methods section. We selected metrics that would be sensitive to the quality of

reduced representations (k-means, knn, and silhouette width) because our goal was to assess these representations and not to build the best possible cell type predictor. The k-means clustering approach evaluates the extent to which a hypersphere in the latent space is capable of capturing cell types accurately. We used both NMI and ARI to measure performance, though results for each are relatively similar so we present only NMI within the main text due to space constraints. The kNN approach evaluates the extent to which local structure in the latent space reflects cell type. The silhouette width approach evaluates the extent to which the within-type distances in the latent space are smaller than the between-type distances.

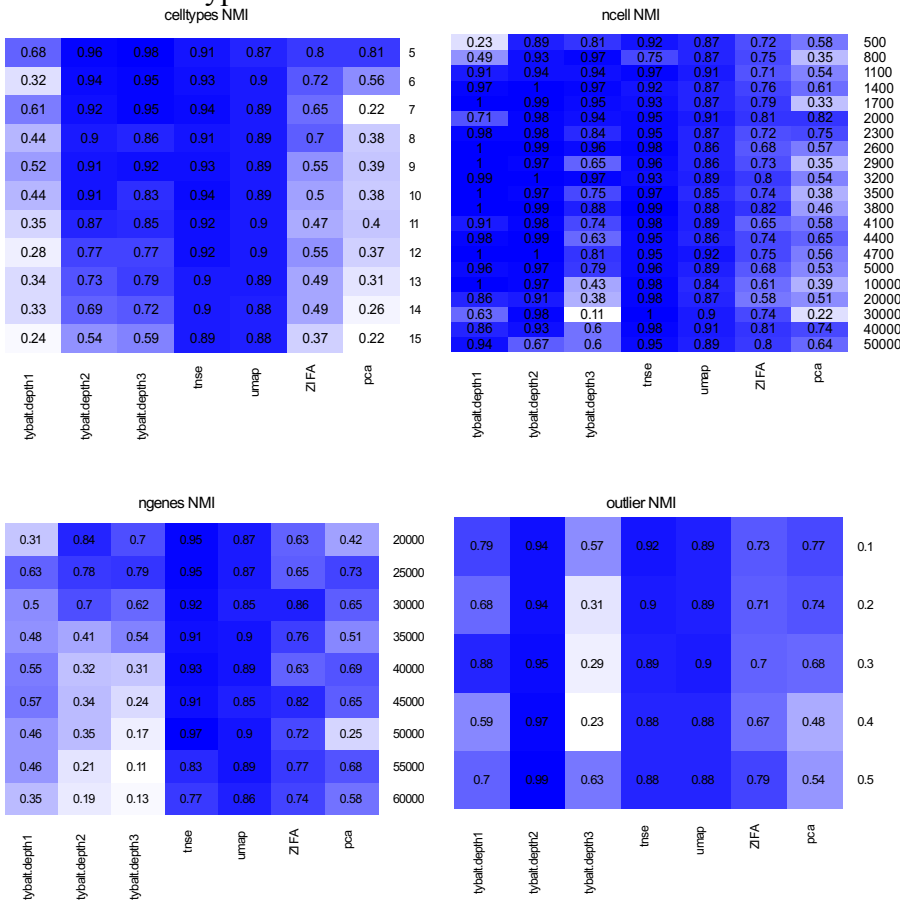


Figure 2: Performance of different dimensionality reduction approaches based on simulated single cell datasets measured by normalized mutual information (NMI)

3.1.1. *k-means based results*

The performance of most methods varied substantially under simulation parameters (Figure 2). As expected, more cell types led to reduced performance, assessed via NMI, of PCA, ZIFA, and the variational autoencoders. As the number of cells changed, the performance of ZIFA and PCA fluctuated. Intriguingly, the three-layer VAE, which had the most parameters to fit and which should have improved with more data, performed worse as the number of cells increased. Later we show that this result is due to substantial parameter sensitivity. Less surprisingly, increasing the number of genes (and consequently parameters) reduced the performance of larger autoencoders. Outliers reduced the performance of PCA but had relatively inconsistent effects on other methods. For the

default parameters, the two-layer Tybalt model was generally high-performing, but both the one- and three-layer models showed variable performance. This surprising sensitivity to simulated data characteristics suggests that VAEs may be very sensitive the fit between parameters and data.

3.1.2. *kNN and silhouette score results*

Results based on the kNN and silhouette evaluations are consistent with the results from k-means. We display results for representative datasets to show variability. The GitHub repository contains complete results. Performing kNN in the latent space revealed relatively poor performance of the linear methods (Figure 3, PCA and ZIFA). UMAP and t-SNE perform well across many combinations, and the VAEs generally performed reasonably well until the number of genes became very high, presumably because the number of parameters leads to insufficient training data.

The silhouette score evaluation tests something slightly different than the k-means and kNN evaluations. While those focus on the extent to which there is some detectable separation between cell types, the silhouette score evaluates the extent to which within cell-type distances are smaller than between cell-type distances. Despite this difference, the results remain consistent (Figure 4) with the other evaluations. As the number of cell types increases, the performance of all method is drops, though the decrease is somewhat less pronounced with t-SNE and UMAP. This evaluation also shows the same unexpected performance drop as the number of cells (thus, examples) increases with three-layer VAE models.

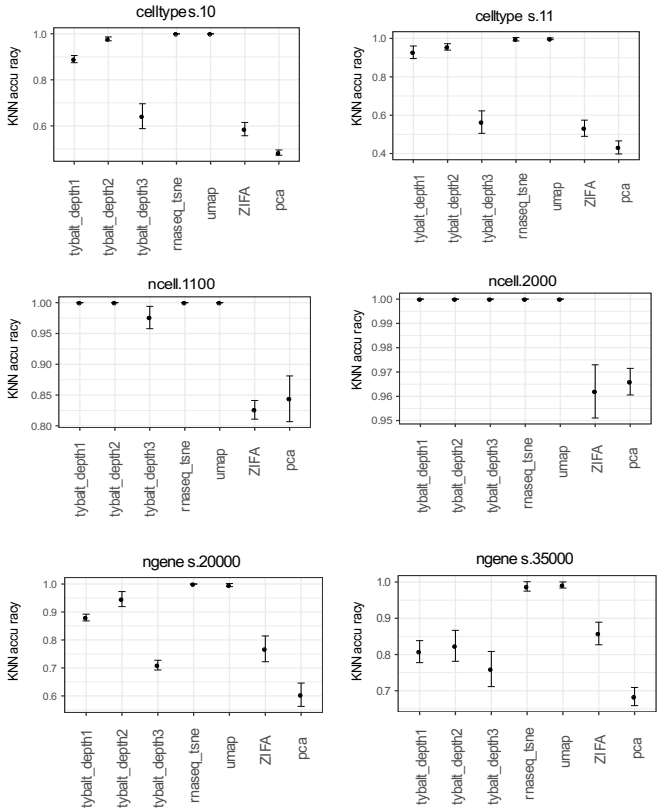


Figure 3: kNN performance for representative simulated single cell datasets under different parameters. Error bars show the standard deviation of accuracy across cross validation intervals, and stability differed between methods.

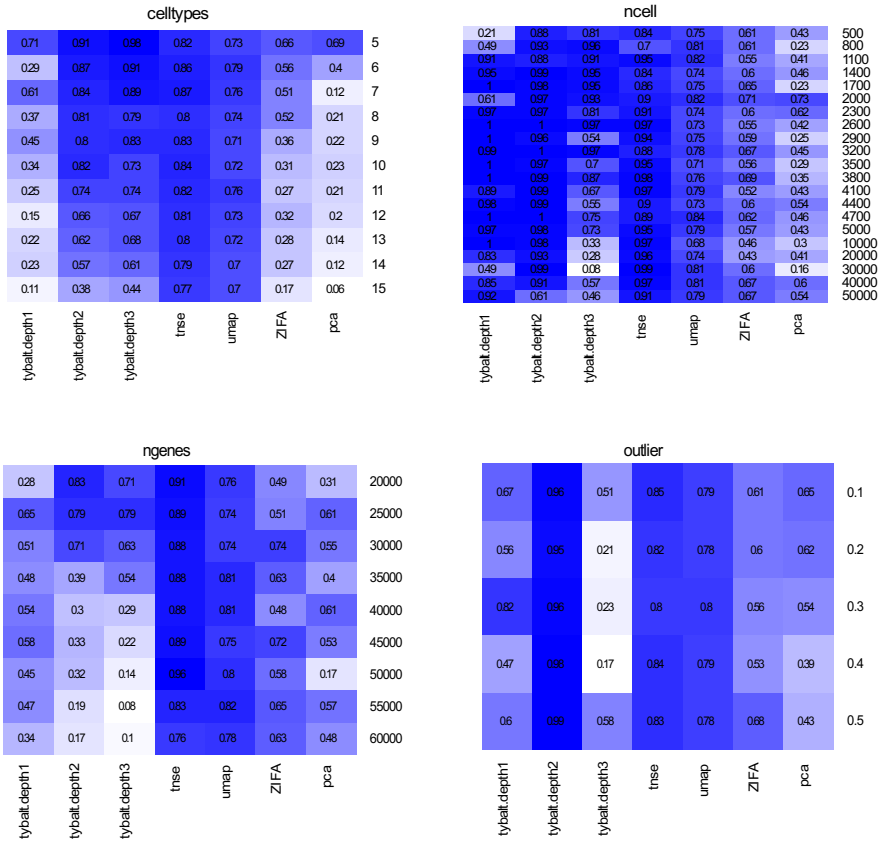


Figure 4: Performance of different dimensionality reduction approaches based on average silhouette score.

3.1.3. *Summary of the performance comparison*

Our results indicate that no dimensionality reduction method outperformed the rest in all cases. The performances of the linear methods (PCA and ZIFA) were generally poorer under the cases that we tested, and the performances of t-SNE and UMAP were generally quite robust within the bounds that were tested. Perhaps the most interesting finding of this stage of the analysis was that the VAE-based methods struggled in expected situations (i.e., when the number of genes, and consequently parameters, increased) but also in unexpected situations (i.e., when the number of cells, and consequently training examples, increased). This suggested that either the model structure or parameter combinations must be poor, because otherwise more examples would always lead to better performance. We explore the implications of this finding more fully in Section 3.3.

3.2. *2-dimensional projection of simulated datasets*

To visualize the results associated with the evaluation described in 3.1, we projected cells into the learned latent spaces and then reduced those spaces to 2-dimensional space via t-SNE on the latent space values while coloring by the simulated cell types (Figure 5). We observe performance characteristics that hint at why the methods exhibited strong or poor performance in different settings. For example, with few outliers the structure of t-SNE remains reasonable, but as the number of outliers increases some points begin to shift to the extremes of the projection. UMAP generally has high between-group distances and low within-group distances and is not affected by cell types.

The linear methods (ZIFA and PCA) along with the single-layer variational autoencoder (Tybalt) appear to unequally space the cell types, even though these are not correlated with each other. The two- and three-layer Tybalt models do not have this relationship, though the three-layer model appears to train poorly with more outliers.

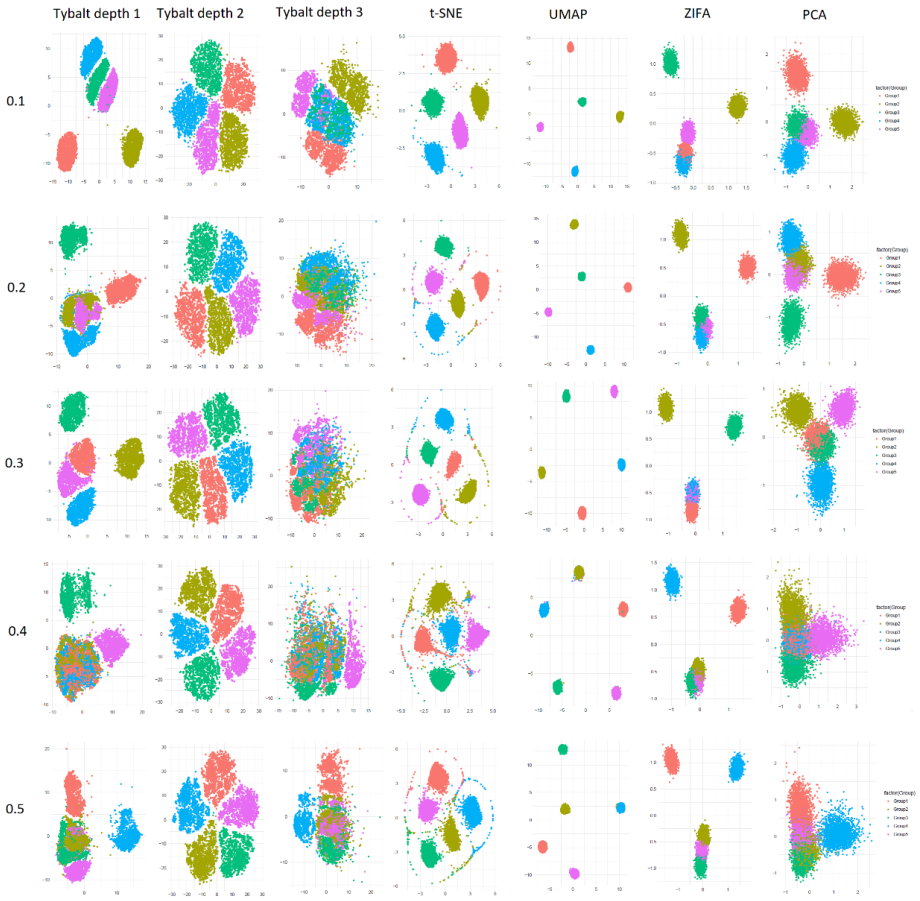


Figure 5: 2-dimensional projection of simulated single cell data using one-, two- and three-layer VAE models (Tybalt), t-SNE, ZIFA, UMAP and PCA based on different proportions of outliers (0.1 – 0.5).

3.3. *Analysis of VAE performance failures*

As observed in the previous section, we found that the three-layer Tybalt model's performance dropped precipitously under certain conditions. Our hypothesis was that the hyperparameters were not appropriate for this setting. We sought to determine the extent to which we could rescue performance under the least expected failure mode from Section 3.1: namely that performance dropped when the number of examples increased. We performed a parameter sweep as described in section 2.3. Note that this grid search is of a very modest size, so we would expect modest performance changes. Results for the k-means evaluation are shown in Table 1. We noticed that the performance of VAE changes dramatically during parameter selection. In this case, performance varies from dismal to better than most the other dimensionality reduction approaches. With 30,000 cells the worst three-layer model has an NMI of zero, while the best has an NMI of 0.96 (Table 1).

Table 1. Best and worst parameter values for two- and three- layer Tybalt models with many cells for simulated datasets. l: learning rate, b: batch size, e: epoch, c: dimensionality of the first hidden layer.

2-layer model

	Best combination						Worst combination					
ncells	l	b	e	c	NMI	ARI	l	b	e	c	NMI	ARI
10000	0.0005	200	25	100	0.99	0.99	0.0005	50	200	500	0.08	0.05
20000	0.001	200	25	250	0.98	0.95	0.001	200	200	500	0.2	0.13
30000	0.0005	100	100	100	0.97	0.97	0.0005	50	100	500	0	0

3-layer model

	Best combination						Worst combination					
ncells	l	b	e	c	NMI	ARI	l	b	e	c	NMI	ARI
10000	0.001	50	100	100	1	1	0.002	200	200	500	0.06	0.04
20000	0.0005	100	25	250	0.97	0.95	0.001	100	100	500	0.02	0.01
30000	0.0005	100	25	250	0.96	0.94	0.0005	50	200	250	0	0

We also selected three single cell datasets of various cell numbers and tissues where author-assigned sample labels were available. Baron et al. [25] and Wang et al. [26] assay the human pancreas with 8569 and 635 cells respectively. Camp et al. [27] measured 777 cells from human liver tissue. As with simulated data, VAE performance changed substantially after parameter tuning, although the range of reasonable parameters appears to be broader than in simulated data (Table 2).

Table 2. Best and worst parameter values for two- and three- layer Tybalt models for real datasets. l: learning rate, b: batch size, e: epoch, c: dimensionality of the first hidden layer.

2-layer model

	Best combination						Worst combination					
Datasets	l	b	e	c	NMI	ARI	l	b	e	c	NMI	ARI
Baron et al.	0.0005	100	25	100	0.64	0.38	0.002	50	200	500	0.36	0.17
Wang et al.	0.001	200	200	500	0.46	0.3	0.0005	200	25	100	0.2	0.11
Camp et al.	0.002	50	25	500	0.81	0.71	0.0005	100	25	100	0.64	0.47

3-layer model

	Best combination						Worst combination					
Datasets	l	b	e	c	NMI	ARI	l	b	e	c	NMI	ARI
Baron et al.	0.0005	100	25	500	0.63	0.36	0.001	100	200	500	0.33	0.17
Wang et al.	0.0005	50	200	500	0.45	0.3	0.0005	200	25	100	0.24	0.13
Camp et al.	0.0005	200	50	500	0.76	0.62	0.0005	200	25	250	0.61	0.42

We projected cells into the latent space learned by Tybalt models pre- and post-tuning to visualize the effect of hyperparameters (Figure 6). The two-layer model was robust within the tested range. With the optimal parameters there was a slightly larger gap between cell types, but the cell types were still clearly separated. For the three-layer model there were substantial differences. Before tuning, the three-layer model shows some signs of a failure to train, which could explain the poor quantitative performance. After tuning, the cell types were clearly separated. These results demonstrate that parameter tuning dramatically affects performance for VAE models in this domain. In the case we evaluated, this appears to be more pronounced with the deeper neural network.

However, it is also possible that the default parameters that we selected to tune around happened to be a relatively robust space for two-layer networks for scRNA-seq data.

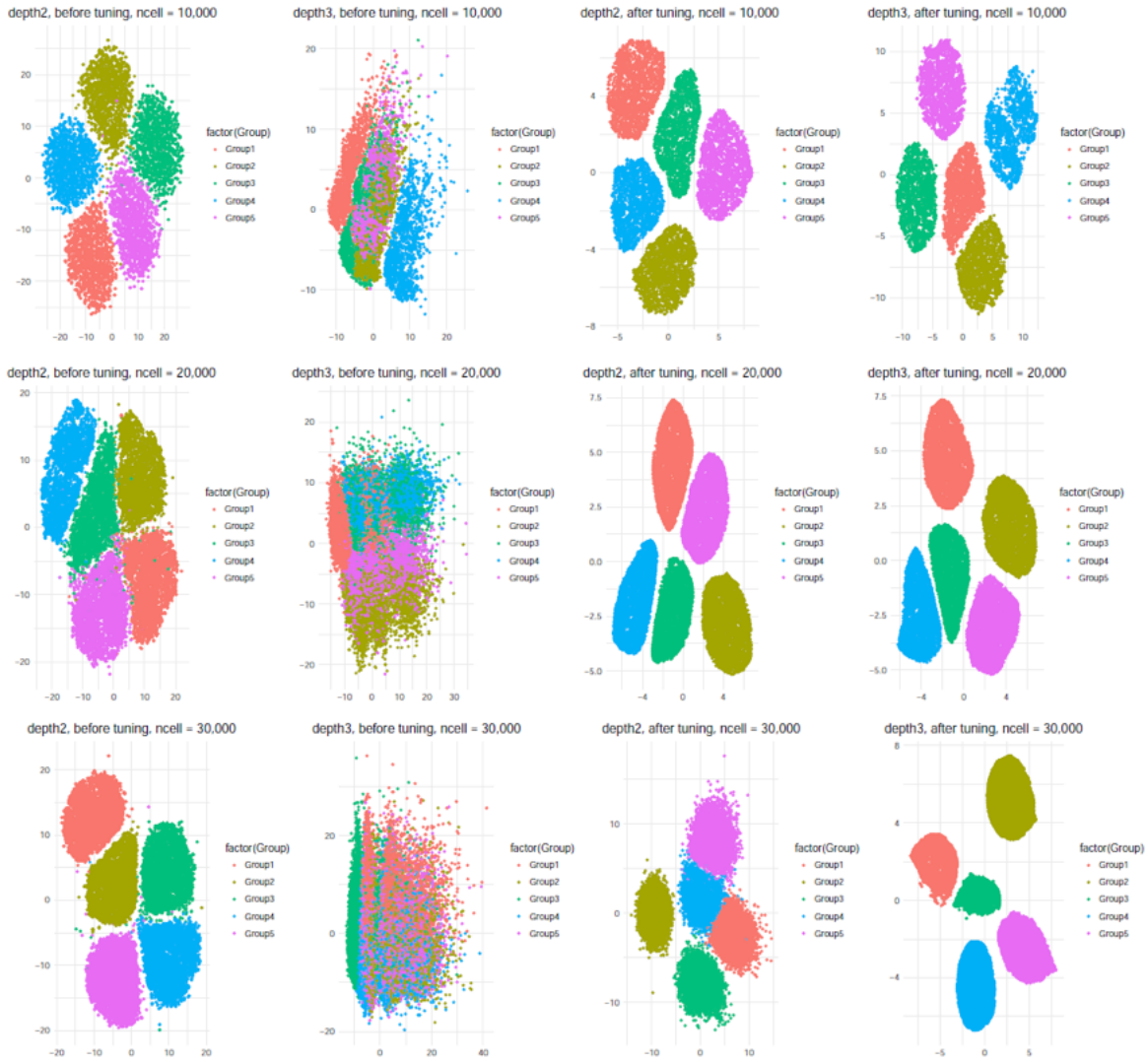


Figure 6: Parameter tuning improve the performance for deeper network. 2-dimensional projection of simulated single cell data using tybalt_depth2 and Tybalt_depth3 before and after parameter tuning.

4. Conclusions

Certain preprints now report good performance for deep neural network methods using VAEs or other types of autoencoders for the analysis of scRNA-seq data [12, 13, 14, 15]. In certain cases, the authors report performance using a set of parameters (see Table 2 of Lopez et al.) without reporting how hyperparameters were tuned or how performance varied through tuning [14]. This poses a particular challenge when authors report performance comparisons with other methods. For example, Deng et al. [15] compare their scScope method with scVI, but they report "we followed the same parameter setting in the original study Lopez et al. [14] and setting the latent dimension to

50." However, Lopez et al., report numerous potential parameter combinations, so which ones were used is impossible to interpret.

We sought to understand the extent to which reported variability in performance was due to differences in methods versus differences in parameter settings. Thus, we evaluated the performance of a simple VAE model developed for bulk gene expression data, Tybalt, under various parameter settings. We find that, in many cases, a base VAE of two layers performs similarly to other methods. However, we also find substantial performance differences with hyperparameter tuning. Though this is not entirely unexpected, the sensitivity of this class of methods under various parameter settings is not widely reported in the literature. Indeed, papers sometimes neglect to report the extent to which parameters were tuned and the extent to which authors optimize the parameter settings of other methods is unclear.

Wolpert and Macready [28] reported a No Free Lunch theorem that states that improved performance of an optimizer on one problem is paired with a decrease in performance in some other area. Our results suggest that algorithms that are more sensitive to parameters, combined with a publication process that encourages method developers to compare their own approaches to others, may experience a Continental Breakfast Included (CBI) effect. We term this the CBI effect because it accrues primarily to certain methods in specific settings. The CBI effect arises when researchers expend more researcher degrees of freedom [29] on their own method instead of other methods. The CBI effect is particularly strong when methods are highly sensitive to parameters, because the results change more substantially with each researcher degree of freedom that is expended.

Our results indicate evaluation of model performance based on empirical results can be misleading in the presence of the CBI effect. For example, we are able to make performance on the same dataset for a three-layer neural network vary from near random to near perfect (Section 3.3). At the current time, we recommend that authors who which to apply these methods expect to perform parameter tuning to achieve acceptable performance, which is likely to require substantially more compute time than is often reported because many manuscripts report only the compute time to train the final model. Moving forward, an unbiased approach is important for model evaluation and comparison. We recommend that authors developing these methods refrain from emphasizing comparisons unless methods are equally tuned and/or some sort of blinded design is used to control researcher degrees of freedom. It may be most practical to rely primarily on disinterested third parties or challenge-based frameworks for comparisons between methods.

5. Reproducibility

We provide the source code and scripts to reproduce the analysis at
https://github.com/greenelab/CZI-Latent-Assessment/tree/master/single_cell_analysis

6. Funding

This work was funded in part by grant 2018-182718 from the Chan Zuckerberg Initiative Donor-Advised Fund (DAF), an advised fund of the Silicon Valley Community Foundation; by grant GBMF 4552 from the Gordon and Betty Moore Foundation; and by R01 HG010067 from the National Institutes of Health's National Human Genome Research Institute.

References

1. E. Shapiro, T. Biezuner, and S. Linnarsson, Nature reviews. Genetics, 9, 618 (2013).
2. A. P. Patel, I. Tirosh, J. J. Trombetta, A. K. Shalek, S. M. Gillespie, H. Wakimoto, D. P. Cahill, B. V. Nahed, W. T. Curry, R. L. Martuza, D. N. Louis, O. Rozenblatt-Rosen, M. L. Suvà, A. Regev, and B. E. Bernstein, Science, 6190, 1396 (2014).
3. S. Semrau, J. E. Goldmann, M. Soumillon, T. S. Mikkelsen, R. Jaenisch, and A. van Oudenaarden, Nature communications, 1, 1096 (2017).
4. S. Wold, Esbensen,K. and Geladi,P., Chemometrics and intelligent laboratory systems, 2, 37 (1987).
5. E. Pierson and C. Yau, Genome biology, 241 (2015).
6. L. V. D. a. H. Maaten, G., J.Mach. Learn. Res., 9, 2579 (2008).
7. E. Becht, C.-A. Dutertre, I. W. H. Kwok, L. G. Ng, F. Ginhoux, and E. W. Newell, bioRxiv, (2018).
8. J. H. Leland McInnes, arXiv (2018-02-09) https://arxiv.org/abs/1802.03426v1,
9. Z. Y. Zhiting Hu, Ruslan Salakhutdinov, Eric P. Xing, arXiv:1706.00550, (2018).
10. G. P. Way and C. S. Greene, Pac Symp Biocomput, 80
11. D. H. Ladislav Rampasek, Petr Smirnov, Benjamin Haibe-Kains, and Anna Goldenberg, arXiv:1706.08203 [stat], June 2017,
12. C. H. Grønbech, M. F. Vording, P. N. Timshel, C. K. Sønderby, T. H. Pers, and O. Winther, bioRxiv, (2018).
13. G. Eraslan, L. M. Simon, M. Mircea, N. S. Mueller, and F. J. Theis, bioRxiv, (2018).
14. R. Lopez, J. Regier, M. B. Cole, M. Jordan, and N. Yosef, bioRxiv, (2018).
15. Y. Deng, F. Bao, Q. Dai, L. Wu, and S. Altschuler, bioRxiv, (2018).
16. J. Tan, M. Ung, C. Cheng, and C. S. Greene, Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing, 132 (2015).
17. L. Zappia, B. Phipson, and A. Oshlack, Genome biology, 1, 174 (2017).
18. S. M. Danilo Jimenez Rezende, and Daan Wierstra, arXiv:1401.4082, (2014).
19. D. P. K. a. M. Welling, arXiv:1312.6114, (2013).
20. F. C. a. others. *Keras (GitHub, 2015)*. 2015.
21. Mart, #237, n. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, USENIX Association, (2016).
22. A. a. J. G. Strehl, Journal of machine learning research, p. 583 (2002. 3(Dec)).
23. L. a. P. A. Hubert, Journal of classification, 2(1), 193 (1985).
24. P. J. Rousseeuw, Computational and Applied Mathematics 20, 53 (1987).
25. M. Baron, A. Veres, S. L. Wolock, A. L. Faust, R. Gaujoux, A. Vetere, J. H. Ryu, B. K. Wagner, S. S. Shen-Orr, A. M. Klein, D. A. Melton, and I. Yanai, Cell systems, 4, 346 (2016).
26. Y. J. Wang, J. Schug, K.-J. Won, C. Liu, A. Naji, D. Avrahami, M. L. Golson, and K. H. Kaestner, Diabetes, 10, 3028 (2016).
27. J. G. Camp, K. Sekine, T. Gerber, H. Loeffler-Wirth, H. Binder, M. Gac, S. Kanton, J. Kageyama, G. Damm, D. Seehofer, L. Belicova, M. Bickle, R. Barsacchi, R. Okuda, E. Yoshizawa, M. Kimura, H. Ayabe, H. Taniguchi, T. Takebe, and B. Treutlein, Nature, 533 (2017).
28. D. H. Wolpert and W. G. Macready, IEEE Transactions on Evolutionary Computation, 1, 67 (1997).
29. J. P. Simmons, L. D. Nelson, and U. Simonsohn, Psychological Science, 11, 1359 (2011).

Shallow Sparsely-Connected Autoencoders for Gene Set Projection

Maxwell P. Gold, Alexander LeNail, and Ernest Fraenkel
Department of Biological Engineering, Massachusetts Institute of Technology, 21 Ames St. Cambridge, MA, 02139, USA
Email: mpgold@mit.edu

When analyzing biological data, it can be helpful to consider gene sets, or predefined groups of biologically related genes. Methods exist for identifying gene sets that are differential between conditions, but large public datasets from consortium projects and single-cell RNA-Sequencing have opened the door for gene set analysis using more sophisticated machine learning techniques, such as autoencoders and variational autoencoders. We present shallow sparsely-connected autoencoders (SSCAs) and variational autoencoders (SSCVAs) as tools for projecting gene-level data onto gene sets. We tested these approaches on single-cell RNA-Sequencing data from blood cells and on RNA-Sequencing data from breast cancer patients. Both SSCA and SSCVA can recover known biological features from these datasets and the SSCVA method often outperforms SSCA (and six existing gene set scoring algorithms) on classification and prediction tasks.

Keywords: autoencoder, variational autoencoder, single-cell RNA-Sequencing, gene set

1. Introduction

RNA-Sequencing (RNA-Seq) experiments can quantify the RNA expression levels for ~20,000 human genes and this data may reveal differences between experimental conditions, such as cancerous tissue vs. healthy tissue. Typically, RNA-Seq analysis begins with identifying genes with differential RNA levels across conditions and determining if such genes are over-represented in any predefined gene sets (i.e. groups of biologically related genes). This standard approach can be useful but is also quite simplistic; it ignores relationships among the genes and assumes all genes in a gene set are equally important to the group.

Consortium projects (such as The Cancer Genome Atlas (TCGA) [1]) and the development of single-cell RNA-Sequencing (scRNA-Seq) [2] have yielded large public datasets for RNA-Seq analysis; this has permitted the use of more complex machine learning techniques, such as autoencoders [3] and variational autoencoders (VAEs) [4], for analyzing those data. These methods can project the high-dimensional gene space onto a lower-dimensional latent space, which may help with visualization, denoising, and/or interpretation [5–7]. Additionally, some neural networks and autoencoders have even been designed to incorporate biological information by using sparsely-connected nodes that only receive inputs from biologically-related genes [8,9].

Many of these neural-network-based and autoencoder-based approaches have focused primarily on increasing accuracy, but recently, groups have used these methods for data interpretation. For example, Way and Greene (2018) used a VAE on TCGA data, wherein they projected RNA-Seq

data onto a reduced latent space, identified nodes that differentiate cancer subtypes, and used the learned model parameters to search for biological significance [10]. Chen *et al.* (2018) detailed a similar approach, whereby they used sparse connections to project genes onto gene sets and then had a fully-connected layer between the gene set nodes and latent nodes [11]; a gene set was considered meaningful if it had a high input weight into a relevant latent superset node.

Here we describe a different approach for using autoencoders for gene set analysis. We present shallow sparsely-connected autoencoders (SSCAs) (Figure 1A) and shallow sparsely-connected variational autoencoders (SSCVAs) (Figure 1B) as tools for projecting gene-level data onto gene sets, wherein those gene set scores can be used for downstream analysis. These methods use a single-layer autoencoder or VAE with sparse connections (representing known biological relationships) in order to attain a value for each gene set. Chen *et al.* (2018) mentioned the SSCA model (Figure 1A) but did not thoroughly explore its utility for gene set projection [11]. There are many statistical methods for gene set scoring (see Section 2.5), but these techniques often rely on assumptions that do not reflect the underlying biology (e.g. all genes are equally important to a gene set). That being said, the machine-learning approaches presented in this work allow for learning a specific nonlinear mapping function for each gene set; thus, each gene within a gene set can be weighted differently and a single gene can have distinct weights across gene sets.

Ideally, the gene set scores should be able to retain high-level information from the gene-level data and provide new insights regarding the relevant gene sets. To test whether the SSCA and SSCVA algorithms can extract such gene set scores, we ran both algorithms on scRNA-Seq data from human blood cells and on RNA-Seq data from patients with breast cancer; we used classification and prediction tasks to compare these new methods to six existing gene set scoring algorithms and assessed the biological interpretability of SSCA and SSCVA by performing differential analysis using the computed scores.

2. Methods

2.1. *Model Summary*

This work explores shallow sparsely-connected autoencoders (SSCAs) (Figure 1A) and shallow sparsely-connected variational autoencoders (SSCVAs) (Figure 1B). Autoencoders learn an encoder function that projects input data onto a lower dimensional space and a decoder function that aims to recover the input data from the low-dimensional projections. The model is trained by minimizing the reconstruction loss (i.e. some measure of distance between the reconstructed output and the original input).

Variational autoencoders (VAEs), however, learn a continuous distribution (typically a multivariate gaussian) to represent the input data. The encoder learns projections onto both a mean vector and a standard deviation vector (which are used to represent a multivariate Gaussian) and the decoder takes samples from the encoded distribution and learns a function to project these samples onto the original space. For VAEs, the model is trained by minimizing both the aforementioned

reconstruction loss and the KL divergence between the learned multivariate Gaussian and a chosen prior distribution (typically the unit Gaussian).

The shallow sparsely-connected autoencoders and VAEs discussed in this work are based on said algorithms, but with two notable restrictions: the encoding/decoding functions are only one layer deep and these layers are sparse (not fully-connected like standard autoencoders), with connections based on known biological relationships. For SSCA, each encoded node represents a gene set and only receives inputs from gene nodes included in the set. For SSCVA, each gene set is represented by a mean vector node and a standard deviation vector node, both of which only receive inputs from the relevant gene nodes. When analyzing the trained SSCVA models, we considered the score for each gene set to be the value of the mean vector node.

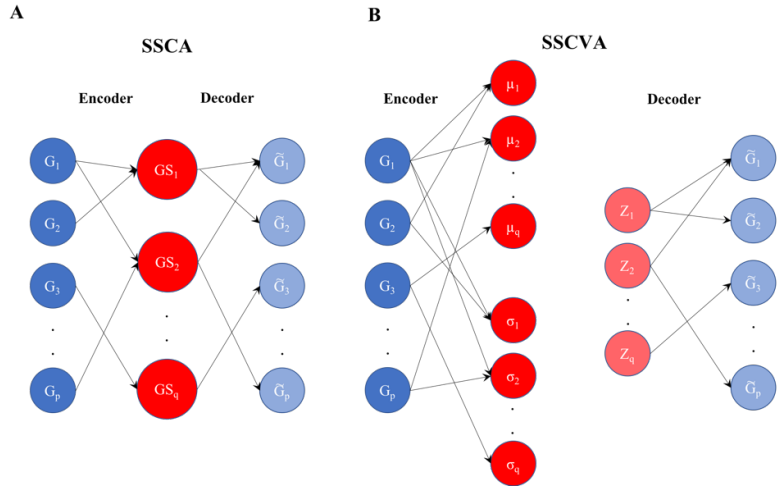


Fig 1. **Diagram for Shallow Sparsely-Connected Autoencoder (SSCA) and Variational Autoencoder (SSCVA)**. A) SSCA model. B) SSCVA model. For SSCA, the input genes (G_1 - G_p) are connected to gene set nodes (GS_1 - GS_q). Each gene set node only receives inputs from the genes within the gene set. Light blue denotes the reconstructed gene values ($\widetilde{G}_1$ - $\widetilde{G}_p$). SSCVA follows the same model, except there is μ node and σ node for each gene set. The z values are collected using the following scheme: $\bar{z} = \bar{\mu} + (\bar{\sigma} * \bar{\epsilon})$ where $\bar{\epsilon} \sim U(0,1)$. Those values are then used to project onto $\widetilde{G}_1$- $\widetilde{G}_p$.

2.2. *Model Coding*

We implemented the models in python using the TensorFlow package [12] (version 1.8.0) and select functions from Keras (version 2.1.6) [13]. We employed hyperbolic tangent (tanh) activation for the encoder functions and sigmoid activation for the decoder functions. For the encoders, we used batch normalization (which scales values to zero mean and unit variance) after linear activation and before tanh activation. Additionally, we trained both models using Adam optimization [14]. The SSCVA code is largely based on public code from Way and Greene (2018) [10] and the sampling procedure follows the scheme where $\bar{z} = \bar{\mu} + (\bar{\sigma} * \bar{\epsilon})$ and $\bar{\epsilon} \sim U(0,1)$ (Figure 1B).

2.3. *Data and Gene Set Summary*

We used two publicly available data sets for this analysis: a single-cell RNA-Seq dataset of 1078 blood cells (dendritic cells and monocytes) [15] and an RNA-Seq dataset from patients with breast cancer from The Cancer Genome Atlas (TCGA) [1,16]. The scRNA-Seq data matrix consists of preprocessed log TPM values for genes for 1078 high-quality cells [15]. For training, the data was scaled to a range of 0-1 using min-max scaling. The breast cancer dataset includes 1093 patients with RNA-Seq data ($\log_2$(FPKM + 1) transformed RSEM values) and matching clinical data [1,16]. A small number of patients have multiple RNA-Seq runs and for these cases, the mean RSEM value for each gene across runs was assigned to the patient. After this step, the breast cancer data was processed in the same manner as the scRNA-Seq dataset.

The gene sets used to create the sparse layers are from the Molecular Signatures Database [17]. We used the transcription factor targets collection (C3.TFT) for scRNA-Seq analysis and the cancer signatures collection (C6) for the breast cancer survival analysis. We then filtered the collections to include only gene sets with more than 15 genes and less than 500 genes, reducing the C3.TFT collection from 615 to 550 gene sets and the C6 collection from 189 to 187 gene sets. Using only the remaining genes, the input matrices were 1078 cells x 10992 genes for the scRNA-Seq data and 1093 patients x 10650 genes for the breast cancer analysis.

2.4. *Hyperparameter Selection*

We considered the following variables for a parameter sweep: learning rate (0.00075, 0.001, 0.002), epochs (50, 100, 150), and L2 regularization (0, 0.05, 0.1). Additionally, we tested warmup (κ) (0.05, 1) for the SSCVA model, where κ controls how quickly the KL loss contributes to the total loss being minimized in the VAE [18]. We kept the optimizer (Adam) and batch size (50) consistent for all trials. We used 90% of the samples for training and 10% for validation and chose the hyperparameters corresponding to the model with the lowest validation loss. For both the blood cell and the breast cancer data, the validation loss for SSCA was lowest for a learning rate of 0.002, 150 epochs, and no L2 regularization. For SSCVA in both analyses, the validation loss was minimized by a learning rate of 0.002, 150 epochs, L2 regularization of 0.1, and κ of 0.05. Hu and Greene [25] recently raised concerns about model comparison analysis when some models are heavily reliant on hyperparameter tuning. Thus, in this work, the SSCA and SSCVA models chosen for comparison are the ones that minimize loss, without any regard for task performance.

2.5. *Other Projection Methods*

In addition to SSCA and SSCVA, we assessed the performance of six other methods for projecting gene data onto gene sets: Average Z-score (Z-Score) [19], Pathway Level Analysis of Gene Expression (PLAGE) [20], Gene Set Variation Analysis (GSVA) [21], single-sample Gene Set Enrichment Analysis (ssGSEA) [22], FastProject (FP) [23], and simple averaging (Average). The Z-Score method normalizes each gene by z-score across samples and considers the gene set score to be the mean normalized value of all genes in a set. PLAGE uses the same z-score normalization

and then performs singular value decomposition (SVD) for each gene set; the gene set scores are the first right singular vector obtained from the SVD. GSVA and ssGSEA are enrichment-based algorithms that utilize distinct methods to rank each gene per sample and then calculate a score for each gene set based on the difference in ranks for genes within the set compared to those outside of it. The averaging method is the arithmetic mean of the RNA-Seq values of all the genes within a gene set. Lastly, FastProject is a tool built for scRNA-Seq data; the algorithm normalizes the data using z-scores while also accounting for sparsity common in scRNA-Seq data and then assigns the gene set score as the mean of the normalized values.

We used the GSVA package in R (version 1.26.0) [24] to calculate GSVA, PLAGE, Z-Score, and ssGSEA scores and ran the FastProject program [23] to compute FP scores. Averaging and autoencoder training were performed in python (per the above procedure). To help with training, we used min-max scaled RNA-Seq values as inputs for the SSCA and SSCVA methods. The other methods used the normalized RNA-Seq values (log TPM for blood cells and RSEM for breast cancer). The only exception is that min-max scaled RNA-Seq values were used for the averaging projection for the breast cancer survival prediction as raw values led to convergence issues.

2.6. *Dendritic Cell Type Classification*

We used the python package Scikit-learn (version 0.19.1) to train the logistic regression models and gaussian mixture models (GMMs) [26]. For the GMMs, we set k = 3 and initialized each model five times (using n_init = 5), with the best result being kept. To compare the predicted clusters to known cell types (provided by [15]), we calculated normalized mutual information using Scikit-learn [26].

2.7. *Breast Cancer Prediction*

We analyzed five-year survival on the breast cancer dataset and only kept patients who survived greater than five years (i.e. TCGA "days_to_follow_up" > 1825 days) or who passed away within five years (i.e. TCGA "days_to_death" < 1825 days). This left 352 patients: 253 survivors and 99 who have passed away. For the survival analysis, we used the lifelines package in python [27] to train a Cox proportional hazards model (Cox PHM) with a step size of 0.3 to help with convergence. Using a 4:1 train/test split, we trained the Cox PHM to predict days of survival from the gene set scores and compared the predicted days of survival to the true values using the concordance index (CI). To assess the importance of gene sets in predicting days of survival, we ranked the gene sets in ascending order by their p-values using a Wald test. We generated boxplots using the python package Matplotlib [28] and performed Mann-Whitney U tests using the python package SciPy [29].

3. Results

We analyzed scRNA-Seq data from blood cells [15] and RNA-Seq data from breast cancer patients [1] to assess the utility of shallow sparsely-connected autoencoders (SSCA) and variational autoencoders (SSCVA) for projecting gene data onto gene sets. We compared the two autoencoder-

based methods to six existing methods for gene set projection (see Methods): GSVA, PLAGE, Z-Score, ssGSEA, FP, and Average.

3.1. *Blood scRNA-Seq Analysis*

When analyzing scRNA-Seq data, it can be difficult to assess the importance of specific transcription factors (TFs) because mRNA levels do not always correlate with protein abundance [30,31], and TF activity is affected by other factors in the cell, such as chromatin accessibility. One potential solution is to use transcription factor target gene sets (i.e. genes whose expression is potentially affected by a given TF); if the genes regulated by a TF are differential between conditions, this could suggest that the TF is biologically relevant. Thus, in order to explore the scRNA-Seq data set from human blood cells, we performed gene set analysis on 550 transcription factor target gene sets from the Molecular Signatures Database [17]. We performed classification tasks using the gene set encodings to determine whether these projections retain high-level information about the dataset and then analyzed the differential features for biological significance.

3.1.1. *Supervised Classification of Cell Types*

The scRNA-Seq data set contains over 1000 individual cells, each of which was assigned one of ten cell types by Villani *et al.* (2017) [15] (six dendritic cell types (DC1-6) and four monocyte cell types (Mono1-4)). We first ran the eight projection methods using the transcription factor target gene sets and then used the resulting gene set scores to train a logistic regression to predict cell type. We used 80% of the samples for training and compared the methods on classification accuracy using test data. This procedure was repeated for multiple distinct cell type combinations (Figure 2).

The cell types used in a given run affected the peak model accuracy, which ranged from 84% (all ten cell types) to 100% (DC1-DC6-Mono1). The model trained using SSCVA gene set scores yielded the highest accuracy in all six trials and was the sole top performer in five of six trials (DC1-DC6-Mono1 being the exception, where many algorithms achieved 100% accuracy). We also compared the performance of SSCVA-based models to logistic regression models trained directly on the gene-level RNA-Seq data; models trained on SSCVA gene set scores never outperformed the RNA-Seq models (Figure 2) but were always within 2% accuracy. Average-based models often led to the second highest accuracy and SSCA-based models typically resulted in the lowest accuracy among the methods tested. These results suggest that for the blood cell dataset, the SSCVA encodings retain more gene-level information about cell type than the other projection methods.

Cell Types	Scaled RNA-Seq	Raw RNA-Seq	GSVA	PLAGE	Z-Score	ssGSEA	FP	Average	SSCVA	SSCA
DC1 - DC6 - Mono1	1	1	1	1	1	0.99	0.98	1	1	1
DC1 - DC3 - DC5	0.984	0.984	0.885	0.852	0.902	0.803	0.902	0.918	0.967	0.885
DC1 - DC2 - DC3	0.878	0.878	0.851	0.797	0.797	0.811	0.797	0.851	0.865	0.73
All Dendritic Cells (DC1-6)	0.919	0.899	0.799	0.812	0.832	0.758	0.866	0.839	0.899	0.758
All Monocytes (Mono1-4)	0.838	0.838	0.75	0.794	0.779	0.735	0.794	0.794	0.838	0.618
All Cells (DC1-6 & Mono1-4)	0.838	0.852	0.773	0.745	0.727	0.722	0.764	0.787	0.838	0.634

Fig 2. **Logistic Regression Test Data Accuracy**. Each row represents a trial with the specific cell types shown in the first column. Additional columns indicate the data type used for training for cell type prediction (i.e. gene-level RNA-Seq data or gene set scores from one of eight algorithms). Values are the classification accuracy of cell types on test data. Yellow emphasizes the highest test accuracy in each row. Scaled RNA-Seq (Min-max scaled gene TPM values from [15]). Raw RNA-Seq (gene TPM values from [15]). See Methods for the full names of gene set projection algorithms.

A

Cell Types	Scaled RNA-Seq	Raw RNA-Seq	GSVA	PLAGE	Z-Score	ssGSEA	FP	Average	SSCVA	SSCA
DC1 - DC6 - Mono1	0.96	0.973	0.95	0.936	0.087	0.37	0.289	0.054	0.946	0.987
DC1 - DC3 - DC6	0.985	0.985	0.524	0.906	0.013	0.38	0.174	0.021	0.704	0.652
DC2 - DC6 - Mono3	0.976	0.686	0.482	0.974	0.133	0.418	0.179	0.107	0.66	0.625
DC2 - DC3 - DC4	0.631	0.598	0.389	0.48	0.027	0.069	0.039	0.049	0.474	0.562
DC1 - DC6 - Mono2	0.971	0.986	0.906	0.942	0.027	0.387	0.207	0.05	0.957	1

B

Cell Types	Scaled RNA-Seq	Raw RNA-Seq	GSVA	PLAGE	Z-Score	ssGSEA	FP	Average	SSCVA	SSCA
DC1 - DC6 - Mono1	1	1	0.959	0.959	0.045	0.326	0.145	0	0.959	0.919
DC1 - DC3 - DC6	1	1	0.494	0.77	0.014	0.647	0.093	0.008	0.719	0.204
DC2 - DC6 - Mono3	1	0.783	0.574	1	0.046	0.34	0.218	0.019	0.708	0.377
DC2 - DC3 - DC4	0.707	0.735	0.324	0.561	0.126	0.058	0.026	0.097	0.637	0.1
DC1 - DC6 - Mono2	1	1	0.807	0.957	0.077	0.226	0.224	0.06	1	0.838

Fig 3. **Gaussian Mixture Model Clustering Normalized Mutual Information (NMI) Values**. A) Training Data normalized mutual information (NMI). B) Test Data normalized mutual information (NMI). Each row represents a trial with the specific cell types shown in the first column. Additional columns indicate the data used for training (gene-level RNA-Seq data or gene set scores from one of eight algorithms). Values are the normalized mutual information scores between output clusters and known cell types. Yellow emphasizes the highest NMI in each row. Scaled RNA-Seq (Min-max scaled gene TPM values from [15]). Raw RNA-Seq (gene TPM values from [15]). See Methods for the full names of gene set projection algorithms.

3.1.2. *Unsupervised Clustering of Cell Types*

We then examined whether unsupervised clustering of the gene set projections could separate samples by cell type. We trained a Gaussian mixture model on the gene set scores from each method for 80% of the relevant samples and this model was used to predict clusters for the training and test data. In order to evaluate the quality of clustering, we calculated the normalized mutual information (NMI) between the predicted clusters and the known cell types. This procedure was repeated for five distinct groups of three cell types and the results are summarized in Figure 3.

For the training data (Figure 3A), SSCA-based and PLAGE-based models performed best with SSCA-based models having the highest NMI in three cases and PLAGE-based models in two cases. SSCVA-based and GSVA-based models also led to comparatively high NMI scores, while Z-Score-based and Average-based models performed poorly in almost all cases. We observed different results for the test data (Figure 3B), however. The DC1-DC6-Mono1 task led to a tie between the models based on scores from GSVA, PLAGE and SSCVA; on the four remaining tasks, SSCVA-based models and PLAGE-based models each scored highest on two. It is noteworthy that the model trained using SSCVA encodings outperformed the SSCA-based model on the test data, a trend also observed in the logistic regression analysis.

3.1.3. *Top Features Detected for SSCVA and SSCA*

In addition to retaining high-level information about the samples, gene set projection methods should help identify biologically meaningful gene sets from the data. In order to assess whether these new methods can recover known biology, we performed differential analysis using the gene set scores. The first trial focused on the DC6 cells, which are also known as plasmacytoid dendritic cells [15]. For each of the 550 gene sets, we calculated the median score for all DC6 samples and the median score for all other dendritic cell samples (DC1-5) and ranked the gene sets based on the absolute value of the difference between these medians. We then performed the same analysis comparing all the dendritic cell types (DC1-6) with monocytes (Mono1-4).

The top hits for these trials are shown in Figure 4. For the DC6 vs. DC1-5 experiment (Figure 4A), STAT5A target genes are the 5th ranked feature for SSCVA. STAT5 plays a substantial role in repressing the development of DC6 cells [32] and thus it makes sense this gene set would distinguish DC6 cells from the other dendritic cells. For the dendritic cells vs. monocytes trial (Figure 4B), the top five hits from the SSCA algorithm include targets of AHR (aryl hydrocarbon receptor), which is noteworthy as AHR has been shown to promote the differentiation of monocytes into dendritic cells [33]. Additionally, CEBPB (also known as C/EBPβ) targets are the top differential feature for SSCVA and this result is reinforced by research showing that CEBPB is one of the key transcriptional regulators of monocyte cells [34]. These few examples support the notion that SSCVA and SSCA may be able to utilize transcription factor target gene sets to help identify transcription factors with differential activity between conditions or cell types.

A

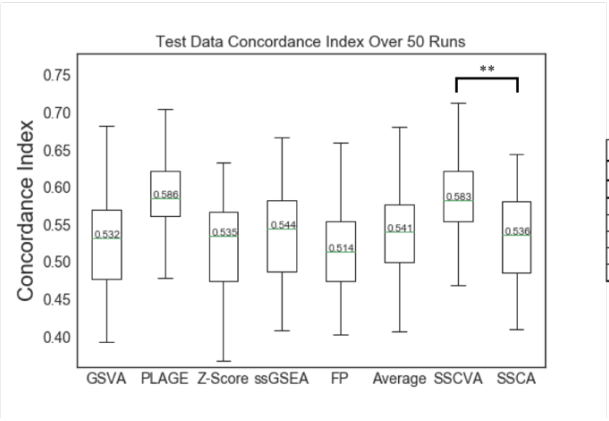
DC6 vs. Other Dendritic Cells (DC1 - 5)		
Rank	**SSCVA**	**SSCA**
1	RACCACAR_AML_Q6	YGTCCTTGR_UNKNOWN
2	ETS_Q4	PAX2_01
3	AAAYWAACM_HFH4_01	SRF_01
4	AREB6_04	EVI1_03
5	STAT5A_03	EVI1_06

B

Dendritic Cells vs. Monocytes		
Rank	**SSCVA**	**SSCA**
1	CEBPB_02	HTF_01
2	ELF1_Q6	YAATNRNNNYNATT_UNKNOWN
3	PU1_Q6	AHR_01
4	ETS2_B	PAX8_B
5	SP1_Q6_01	PAX3_B

Fig 4. Top Five Differential Features for Dendritic Cell Analysis. A) Top features comparing DC6 cells vs. the other five dendritic cell types (DC1 - 5). B) Top features comparing all dendritic cells (DC1 - 6) vs. all monocytes (Mono1 - 4).

A



B

Breast Cancer Survival Prediction				
	SSCVA		**SSCA**	
Rank	**Gene Set**	**Avg. Rank**	**Gene Set**	**Avg. Rank**
1	RB_DN.V1_DN	31.64	E2F1_UP.V1_UP	28.74
2	KRAS.50_UP.V1_UP	33.98	KRAS.LUNG.BREAST_UP.V1_DN	36.88
3	RAPA_EARLY_UP.V1_UP	38.58	CRX_DN.V1_UP	41.8
4	MYC_UP.V1_DN	48.58	KRAS.DF.V1_UP	46.5
5	GCNP_SHH_UP_LATE.V1_DN	49.58	E2F3_UP.V1_DN	49.98

Fig 5. Breast Cancer Survival Analysis. A) Box and Whisker Plot for Concordance Index Values. Each gene set projection algorithm was tested 50 times for survival prediction and the concordance index scores are plotted with the median CI value labeled. ** emphasizes the significant difference between SSCVA and SSCA at $p < 0.005$ (Mann-Whitney U test). SSCVA is also significantly different from GSVA, Z-Score, ssGSEA, FP and Average at $p < 0.005$. B) Top ranked features in predicting breast cancer survival (see Methods). Avg. Rank shows the mean rank out of 187 gene sets over the fifty runs.

3.2. *Breast Cancer Survival Analysis*

We also analyzed a dataset from The Cancer Genome Atlas (TCGA) that includes RNA-Seq data and clinical survival data from 1093 breast cancer patients. In order to attain gene set scores, we first ran the RNA-Seq data through the eight projection algorithms using 187 cancer signature gene sets; since the analysis was focused on predicting five-year survival, the dataset was then reduced to the 352 patients that have been followed for more than five years or have passed away. Once the final datasets were processed, we trained a Cox proportional hazards model (Cox PHM) to predict survival from the encodings for each method using 80% of the training data. The trained Cox PHM was then used to predict survival on the training and test data and success was measured by the

concordance index between the actual and predicted days of survival. This was repeated fifty times with distinct training/test splits.

When analyzing the Cox PHM predictions on the test data, models for all eight gene set scoring methods showed a wide range of concordance index values across the fifty trials (Figure 5A). PLAGE-based and SSCVA-based models performed best (median concordance index ~0.58), while the other projection methods led to models with a median concordance index of ~0.54. There is no significant difference between the SSCVA and PLAGE results, but SSCVA concordance index values are significantly different than the other six models (p value < 0.005, Mann-Whitney U test).

Additionally, each Cox PHM outputs a list of features ranked by their effect on survival (see Methods). We collected this ranked list for each of the fifty models for the SSCA and SSCVA encodings (Figure 5B). For SSCVA, the top ranked feature across the fifty runs is RB_DN.V1_DN and the RB-loss signature (low RB1) is associated with poor disease outcome in breast cancer [35]. Additionally, the top ranked feature for SSCA is E2F1_UP.V1_UP; this result is supported by previous research as well, as E2F1 transcript levels are related to breast cancer outcome [36].

4. Discussion

This work explores shallow sparsely-connected autoencoders (SSCAs) and variational autoencoders (SSCVAs) as methods for projecting RNA-Seq data onto gene sets. When using test data, models trained on the SSCVA encodings often performed as well as the models trained on the gene-level RNA-Seq data and frequently outperformed (or matched) the existing projection algorithms. SSCA-based models, however, performed well on training data, but poorly on test data. These results suggest that the SSCVA encoding space may be better suited to extrapolation than that of SSCA, but future work is necessary to confirm and interpret this trend.

Additionally, it is difficult to assess a method's ability to recover known biology without a ground truth, but we evaluated SSCA and SSCVA on whether differential analysis produced reasonable results. For the blood scRNA-Seq data set, we found the top hits for SSCVA and SSCA included known transcriptional regulators of the groups being tested. Moreover, for the cancer analysis, the top gene sets for both SSCA and SSCVA are cancer signatures related to genes previously associated with breast cancer survival. These observations do not prove that SSCA and SSCVA can uncover insightful biology in all situations, but it is encouraging that the methods identify known features in the data sets tested.

Compared to the other methods discussed, the shallow sparsely-connected autoencoder framework provides greater flexibility for modeling biological phenomena. For instance, if a transcription factor acts as both an activator and a repressor, any given target gene may be up or downregulated. The averaging-based methods (Z-Score, FP, and Average) may miss this trend because the combination of high and low values can reduce the signal. Additionally, the averaging-based approaches and the enrichment-based approaches (ssGSEA and GSVA) both weight all genes equally within a gene set, despite the fact some genes may be more relevant to the gene set than others. PLAGE addresses this issue by learning a specific mapping for each gene set, but the algorithm is limited to finding a linear combination of gene values. SSCA and SSCVA, however,

can learn specific nonlinear mappings for each gene set, which could be useful for modeling complex biological relationships. Moreover, the mapping functions learned by SSCA and SSCVA can potentially provide more information about the importance of genes within specific gene sets.

Further exploration is required to better understand the utility of these models for single-cell omics data sets. For instance, SSCVAs may be particularly useful for analysis of cellular differentiation. Variational autoencoders are designed to produce an encoding space where clusters are distinguishable, but close together, and this can result in smooth transitions between groups of samples; thus, the SSCVA scores can potentially be leveraged for identification and visualization of gene sets that transition in importance throughout differentiation. Additionally, this framework could potentially be applied to other gene-associated omics types, such as methylation.

Unfortunately, a weakness of autoencoder-based methods is that the results may not be entirely consistent between runs; the other six methods tested yield the same result every time, but since autoencoders are initialized randomly each trial, the learned encoder function (and thus the gene set scores) may not be identical across runs. This observation has also been noted by Chen *et al.* (2018) [11] and we are currently exploring whether changes in activation functions, hyperparameters, and/or regularization can improve consistency, while maintaining classification accuracy.

Overall this work supports the use of SSCA and SSCVA for gene set analysis on large RNA-Seq data sets. These methods still require more rigorous testing and evaluation, and future work on this project will be dedicated to improving consistency between runs and understanding situations and data types where SSCA and/or SSCVA may be particularly useful.

Acknowledgements

This work was supported by NIH grants R01NS089076 and 1U01CA18498. We would like to thank the PSB reviewers for their thoughtful comments and helpful suggestions and also want to acknowledge Ludwig Schmidt for informative conversations regarding the models.

References

1. Weinstein, J. N., Collisson, E. a, Mills, G. B., Shaw, K. R. M., Ozenberger, B. a, Ellrott, K., Shmulevich, I., Sander, C. & Stuart, J. M. *Nat. Genet.* **45,** 1113 (2013).
2. Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., Wang, X., Bodeau, J., Tuch, B. B., Siddiqui, A., Lao, K. & Surani, M. A. *Nat. Methods* **6,** 377 (2009).
3. Liou, C. Y., Huang, J. C. & Yang, W. C. in *Neurocomputing* **71,** 3150 (2008).
4. Kingma, D. P. & Welling, M. *Ppt* (2013). doi:10.1051/0004-6361/201527329
5. žurauskiene, J. & Yau, C. *BMC Bioinformatics* **17,** (2016).
6. Xie, R., Wen, J., Quitadamo, A., Cheng, J. & Shi, X. *BMC Genomics* **18,** (2017).
7. Wang, Y., Solus, L., Dai Yang, K. & Uhler, C. *ArXiv Prepr. arXiv* 1705.10220 (2017).
8. Lin, C., Jain, S., Kim, H. & Bar-Joseph, Z. *Nucleic Acids Res.* **45,** (2017).
9. Kang, T., Ding, W., Zhang, L., Ziemek, D. & Zarringhalam, K. *BMC Bioinformatics* **18,** (2017).

10. Way, G. P. & Greene, C. S. *Pac. Symp. Biocomput.* **23,** 80 (2018).
11. Chen, H.-I., Chiu, Y.-C., Zhang, T., Zhang, S., Huang, Y. & Chen, Y. *ArXiv e-prints* (2018).
12. Abadi, M. *et al.* (2015).
13. Chollet, F. *GitHub* (2015). Available at: https://github.com/fchollet/keras.
14. Kingma, D. & Ba, J. *arXiv Prepr. arXiv1412.6980* 1 (2014). doi:10.1109/ICCE.2017.7889386
15. Villani, A.-C. *et al. Science (80-.).* **356,** eaah4573 (2017).
16. Grossman, R. L., Heath, A. P., Ferretti, V., Varmus, H. E., Lowy, D. R., Kibbe, W. A. & Staudt, L. M. *N. Engl. J. Med.* (2016). doi:10.1056/NEJMp1607591
17. Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. a, Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S. & Mesirov, J. P. *Proc. Natl. Acad. Sci. U. S. A.* **102,** 15545 (2005).
18. Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K. & Winther, O. *Nips* **48,** (2016).
19. Lee, E., Chuang, H. Y., Kim, J. W., Ideker, T. & Lee, D. *PLoS Comput. Biol.* **4,** (2008).
20. Tomfohr, J., Lu, J. & Kepler, T. B. *BMC Bioinformatics* **6,** (2005).
21. Hänzelmann, S., Castelo, R., Guinney, J., Kim, S. C., Seo, Y. J., Chung, W., Eum, H. H., Nam, D.-H., Kim, J., Joo, K. M. & Park, W.-Y. *BMC Bioinformatics* **14,** 7 (2013).
22. Barbie, D. A. *et al. Nature* **462,** 108 (2009).
23. DeTomaso, D. & Yosef, N. *BMC Bioinformatics* **17,** (2016).
24. Sonja, H., Castelo, R. & Guinney, J. *Bioconductor.org* 1 (2014).
25. Hu, Q. & Greene, C. S. *bioRxiv* (2018).
26. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. *J. Mach. Learn. Res.* **12,** 2825 (2011).
27. Davidson-Pilon, C. *et al.* (2018). doi:10.5281/zenodo.1252342
28. Hunter, J. D. *Comput. Sci. Eng.* (2007). doi:10.1109/MCSE.2007.55
29. Oliphant, T. E. *Comput. Sci. Eng.* (2007). doi:10.1109/MCSE.2007.58
30. Lundberg, E., Fagerberg, L., Klevebring, D., Matic, I., Geiger, T., Cox, J., Älgenäs, C., Lundeberg, J., Mann, M. & Uhlen, M. *Mol. Syst. Biol.* (2010). doi:10.1038/msb.2010.106
31. Vogel, C. & Marcotte, E. M. *Nat. Rev. Genet.* (2012). doi:10.1038/nrg3185
32. Esashi, E., Wang, Y. H., Perng, O., Qin, X. F., Liu, Y. J. & Watowich, S. S. *Immunity* **28,** 509 (2008).
33. Goudot, C., Coillard, A., Villani, A. C., Gueguen, P., Cros, A., Sarkizova, S., Tang-Huau, T. L., Bohec, M., Baulande, S., Hacohen, N., Amigorena, S. & Segura, E. *Immunity* **47,** 582 (2017).
34. Huber, R., Pietsch, D., Panterodt, T. & Brand, K. *Cellular Signalling* **24,** 1287 (2012).
35. Ertel, A., Dean, J. L., Rui, H., Liu, C., Witkiewicz, A., Knudsen, K. E. & Knudsen, E. S. *Cell Cycle* **9,** 4153 (2010).
36. Hallett, R. M. & Hassell, J. A. *BMC Res Notes* **4,** 95 (2011).

When Biology Gets Personal: Hidden Challenges of Privacy and Ethics in Biological Big Data

Gamze Gürsoy*

Computational Biology and Bioinformatics Program, Molecular Biophysics & Biochemistry, Yale University, New Haven, CT, 06511, USA
Email: gamze.gursoy@yale.edu

Arif Harmanci

Center for Precision Health, School of Biomedical Informatics, University of Texas Health Science Center, Houston, TX, 77030, USA
Email: arif.o.harmanci@uth.tmc.edu

Haixu Tang†

School of Informatics, Computing and Engineering, Indiana University Bloomington, Bloomington, IN, 47405, USA
Email: hatang@indiana.edu

Erman Ayday

Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH, 44106, USA
Email: exa208@case.edu

Steven E. Brenner#

University of California Berkeley, CA, 94720-3012, USA
Email: brenner@compbio.berkeley.edu

High-throughput technologies for biological data acquisition are advancing at an increasing pace. Most prominently, the decreasing cost of DNA sequencing has led to an exponential growth of sequence information, including individual human genomes. This session of the 2019 Pacific Symposium on Biocomputing presents the distinctive privacy and ethical challenges related to the generation, storage, processing, study, and sharing of individuals' biological data generated by multitude of technologies including but not limited to genomics, proteomics, metagenomics, bioimaging, biosensors, and personal health trackers. The mission is to bring together computational biologists, experimental biologists, computer scientists, ethicists, and policy and lawmakers to share ideas, discuss the challenges related to biological data and privacy.

Keywords: biological data privacy, genomics, genetic testing

* This work is partially supported by NIH grant U01EB023686.

† This work is partially supported by NIH grant U01EB023685 and NSF grant CNS-1408874.

This work is partially supported by NIH grant U01EB023686 and NIH grant U41HG007346.

1. Introduction

Data privacy is an important topic of debate crossing many different fields such as ethics, sociology, law, political science and forensic science. Thanks to the rapid reduction of the DNA sequencing cost in the past decade, the number and the volume of available genomic data have exponentially increased [1]. Hence, individuals' genomic data has recently emerged as one of the major foci of studies on privacy as availability of genetic information gives rise to privacy concerns [2]. For example, individuals express concern that genetic predisposition to diseases may bias insurance companies or enable unlawful discrimination by employers [3,4,5]. On a larger scale, imagine the economic repercussions had it been leaked that the CEO of Apple Computer had pancreatic cancer and was not adhering to a typical oncological regimen. Recently it has been also shown that high throughput molecular phenotype datasets such as functional genomic and metabolomics measurements, and microbiome measurements increased the number of quasi-identifiers for participating individuals that can be used by adversaries for re-identification purposes [6,7,8,9]. In addition, the emergence of electronic health records (EHR) with the rise of personalized medicine makes patients vulnerable to breaching privacy. These results indicate that privacy concerns over sharing personalized biological data will increase quickly with the increase in the number of genetic and ancestry testing companies, which collect and distribute very large amount of health related data, including genetic data (such as 23andMe) or health and fitness tracking data (such as fitbit). The data collection and sharing methods that these companies use call for a public discussion of privacy considerations around these new concepts. Moreover, the recent arrest of the Golden State Killer, through long-range familial search on consumer genomics databases, sparked questions over the risk of re-identification based on genetic testing taken by a relative. Recent two studies showed the statistical risk of identifying relatives as being high by using long-range familial searches [10,11].

Protecting the privacy of study participants has emerged as an important issue in genotype-phenotype association studies. Several studies investigated whether a genome of an individual can be detected in a mixture [12,13,14,15]. As a result, various counter-measures have been proposed to protect participant privacy [16]. As the number of genotype-phenotype datasets increase, new routes for breaching privacy such as cross-referencing multiple databases opened up [17,18]. Access control, data anonymization and cryptographic techniques were studied to prevent privacy breaches [4]. Ultimately, the ability to keep these data private is unclear, and so preparations for both small and catastrophic leaks must be made [5]. As the technologies increase, new data types are being released and more studies to investigate the potential privacy breaches will be needed. This area of research has become more and more interdisciplinary, where ethics researchers inform researchers who work on privacy-preserving techniques, while these techniques inform policymakers to reform laws and policies.

On the other side of the privacy problem, the benefit and importance of open data sharing is widely acknowledged, as the solutions such as access control or cryptographic techniques delay the access to the data by average researchers either by creating bureaucratic bottlenecks or technical challenges. Open data sharing harbors the collaboration between different biomedical researchers by allowing rapid exchange of the information. Funding agencies and research organizations are increasingly supporting new means of data sharing and new requirements for making data publicly available while preserving participants' privacy [19]. This increases the value of the techniques and policies that prevent the sensitive information leakage while promoting data sharing.

The papers featured in this session represent various aspects of biological data privacy highlighting a number of problems and solutions that need to be addressed to protect privacy of individuals while encouraging open data sharing. Topics in this session include making inferences on complex phenotypes in

large biobanks, patient re-identification through electronic health records and countermeasures, privacy-preserving GWAS studies as well as efforts on improving informed consents for AllofUs research project.

2. Podium Presentations

After the seminal work by Homer et. al [12], the policies on how to share GWAS results have been changed and only summary statistics are allowed to share publicly. **Gasdaska et al. [20]** explore the possibility of using these summary statistics to make inferences about the hidden, complex phenotypes that are derived from two or more phenotypes. This potentially reveals information about the participants that they may not want to disclose. Investigators validated their statistical derivations on simulated and real datasets.

As **A. Gasdaska** and colleagues [20] showed that sharing statistical aggregates from GWAS might have sensitive information leakages and also demonstrated that how complex phenotypes can be analyzed in terms of simple phenotypes in a privacy preserving fashion, **S. Simmons** and colleagues [21] showed us how we could reduce this leakage by introducing a Laplacian noise to the released data. The investigators presented a novel method for measuring privacy loss in GWAS summary statistics. This was achieved by providing a probabilistic formulation for measuring the risk of releasing summary statistics as the posterior probability of an individual being in the cohort. With the introduction of an MCMC-based method for computing this posterior probability, the authors reduced the degree of privacy leakage with the same amount of data released. This work presented interesting ideas on how to control the privacy risk by setting a noise level and the amount of data to be released.

K. Johnson and colleagues [22] studied the privacy leakages of Electronic Health Records. They showed that lab tests can be used as quasi-identifiers for patients for re-identification of patients' medical records. The investigators used the EHR at Mount Sinai Hospital as a case study. This study took an even more interesting turn when they used variational auto-encoder to encode the lab test results to reduce the privacy risk of re-identification. They showed a substantial decrease in re-identification risks when the lab tests were stored as latent variables while the encoded test results still provide almost the same utility as original results when compared in terms of classification accuracy. Although further work is required to show how decoding-encoding will be achieved in this new representation, the novel idea of storing data will open up the doors for storing other kind of private data in the future.

3. Posters with Published Papers

This year's poster session with papers published in the proceedings will feature a unique study that has not been explored at PSB before by **M. Doerr** and colleagues [23]. They designed a study to give a comprehensive overview of existing jurisdictions for the informed consent process for the AllofUs initiative and its compliance with the state/territory regulations. This study will be of great interest for the investigators of the AllofUs project, which aims to collect a vast amount of biomedical data from a million of Americans.

4. Acknowledgments

We would like to thank the members of the program committee for reviewing all submissions and providing expert critiques used in evaluating manuscripts for inclusion into the session and the PSB proceedings. We would also like to thank the PSB 2018 chairs and Tiffany Murray of Stanford University for their efforts in organizing the meeting.

References

1. Sboner A, Mu X, Greenbaum D, Auerbach RK, Gerstein MB. The real cost of sequencing: higher than you think! Genome Biology, 2011;12(8):125.
2. Brenner SE. Be prepared for the big genome leak. Nature, 2013;498:139
3. Joly Y, Dyke SOM, Knoppers BM, Pastinen T. Are Data Sharing and Privacy Protection Mutually Exclusive? Cell, 2016;167(5):1150-1154.
4. Joly Y, Feze IN, Song L, Knoppers BM. Comparative Approaches to Genetic Discrimination: Chasing Shadows? Trends Genet, 2017;33(5):299-302.
5. Erlich Y, Narayanan A. Routes for breaching and protecting genetic privacy. Nat. Rev. Genet., 2014;15(6):409-421.
6. Harmanci A, Gerstein M. Quantification of private information leakage from phenotype-genotype data: linking attacks. Nature Methods, 2016;13(3):251-256.
7. Harmanci A, Gerstein M. Analysis of Sensitive Information Leakage in Functional Genomics Signal Profiles through Genomic Deletions. Nature Communications, 2018; 9 (1), 2453
8. Gürsoy G, Harmanci A, Green M, Navarro F, Gerstein M. Sensitive information leakage from functional genomics data: Theoretical quantifications & practical file formats for privacy preservation, 2018, Biorxiv
9. Franzosa EA, Huang K, Meadow JF, Gevers D, Lemon KP, Bohannan BJ, Huttenhower C. Identifying personal microbiomes using metagenomic codes. *Proc Natl Acad Sci U S A.* 112(22):E2930-8 (2015).
10. Erlich, Y. *et al.* Identity inference of genomic data using long-range familial searches. *Science.* (2018).
11. Kim J, Edge MD, Algee-Hewitt BFB, Li JZ, Rosenberg NA. Statistical Detection of Relatives Typed with Disjoint Forensic and Biomedical Loci. *Cell.* (2018).
12. Homer, N. *et al.* Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genet.* 4, e1000167 (2008).
13. Im, H.K., Gamazon, E.R., Nicolae, D.L. & Cox, N.J. On sharing quantitative trait GWAS results in an era of multiple-omics data and the limits of genomic privacy. *Am. J. Hum. Genet.* 90, 591–598 (2012).
14. Lunshof, J.E., Chadwick, R., Vorhaus, D.B. & Church, G.M. From genetic privacy to open consent. *Nat. Rev. Genet.* 9, 406–411 (2008).
15. Church, G. *et al.* Public access to genome-wide data: five views on balancing research with privacy and protection. *PLoS Genet.* 5, e1000665 (2009).
16. Jiang X, Zhao Y, Wang X, Malin B, Wang S, Ohno-Machado L, Tang H. A community assessment of privacy preserving techniques for human genomes. BMC Med Inform Decis Mak. 2014;14 Suppl 1:S1.
17. Gymrek M, McGuire AL, Golan D, Halperin E, Erlich Y. Identifying personal genomes by surname inference. Science, 2013;339(6117):321-324
18. Sweeney L. Simple demographics often identify people uniquely. Carnegie Mellon University, unpublished, 2000.
19. National Institute of Health data sharing policy. https://grants.nih.gov/grants/guide/notice-files/NOT-OD-17-110.html

20. Gasdaska A, Friend D, Chen R, Westra J, Zawistowski M, Lindsey W, Tintle N. Leveraging summary statistics to make inferences about complex phenotypes in large biobanks.
21. Simmons S, Berger B, Sahinalp C. Protecting Genomic Data Privacy with Probabilistic Modeling
22. Johnson KW, De Freitas JK, Glicksberg BS, Bobe JR, Dudley JT. Evaluation of patient re-identification using laboratory test orders and mitigation via latent space variable.
23. Doerr M, Grayson S, Moore S, Suver C, Wilbanks J, Wagner J. Implementing a universal informed consent process for the *All of Us* Research Program

Leveraging summary statistics to make inferences about complex phenotypes in large biobanks [a]

Angela Gasdaska[†]
Department of Mathematics and Computer Science and Department of Quantitative Theory and Methods, Emory University, Atlanta, GA 30322, USA
Email: aegasdaska@gmail.com

Derek Friend[†]
Department of Geography, University of Nevada, Reno, NV 89557, USA
Email: derekfriend@outlook.com

Rachel Chen
Department of Statistics, North Carolina State University, Raleigh, NC 27695, USA
Email: rschen@ncsu.edu

Jason Westra
Department of Math, Computer Science, and Statistics, Dordt College, Sioux Center, IA 51250, USA
Email: westrajason@hotmail.com;

Matthew Zawistowski
Department of Biostatistics, University of Michigan, Ann Arbor, MI 48109, USA
Email: mattz@umich.edu

William Lindsey
Department of Math, Computer Science, and Statistics, Dordt College, Sioux Center, IA 51250, USA
Email: William.Lindsey@dordt.edu

Nathan Tintle[*]
Department of Math, Computer Science, and Statistics, Dordt College, Sioux Center, IA 51250, USA
Email: Nathan.Tintle@dordt.edu

As genetic sequencing becomes less expensive and data sets linking genetic data and medical records (e.g., Biobanks) become larger and more common, issues of data privacy and computational challenges become more necessary to address in order to realize the benefits of these datasets. One possibility for alleviating these issues is through the use of already-computed summary statistics (e.g., slopes and standard errors from a regression model of a phenotype on a genotype). If groups share summary statistics from their analyses of biobanks, many of the privacy issues and computational challenges concerning the access of these data could be bypassed. In this paper we explore the possibility of using summary statistics from simple linear models of phenotype on genotype in order to make inferences about more complex phenotypes (those that are derived from two or more simple phenotypes). We provide exact formulas for the slope, intercept, and standard error of the slope for linear regressions when combining phenotypes. Derived equations are validated via simulation and tested on a real data set exploring the genetics of fatty acids.

Keywords: privacy, biobank, genetics, genome-wide association study, single nucleotide variant, computational challenges, data security, phenotypes

---

[†] Contributed equally

[a] Work supported by NIH-2R15HG006915 and Dordt College

[*] Corresponding author

1. Introduction

The continued move to digitize medical records raises a plethora of opportunities and challenges in the search to elucidate the genetic and environmental contributions to human disease. The amount of genetic, environmental, and disease-related data continues to grow rapidly, offering new opportunities to discover relationships between genetic variants and expressed physical characteristics. Of particular interest are the genetic contributions to diseases that can have dramatic impacts on societal well-being (e.g., cardiovascular diseases, mental health, and cancer). The advent of large, publicly available biobanks (e.g., UK Biobank[1]) offers exciting possibilities for leveraging these datasets to have a dramatic impact on human health and disease.

However, this unprecedented opportunity also comes with roadblocks and challenges.[2] The size of datasets in biobanks makes it challenging to transfer, store, and analyze them locally. And even though cloud computing minimizes some of these issues, they bring their own challenges with regard to cost (storage and computation), transfer, and access to cloud computing systems. Furthermore, data security and privacy issues are of paramount importance throughout all aspects of the data access, storage, and analysis pipeline.[3-4] Thus, there is a great demand for simplified data transfer, exploration, visualization, and analysis strategies which simultaneously address privacy, security, storage, and computational challenges, while still allowing researchers to make the best possible use of biobank repositories.

An interesting recent development related to these issues are efforts to provide summary statistics in publicly available formats. For example, GeneAtlas provides basic summary statistics for simple linear regression models of each available single nucleotide variants with each available phenotypic variable for 452 thousand individuals in the UK Biobank.[5] Likewise, Pheweb provides access to the UK Biobank data via a series of easy-to-navigate visualization and summary tools based on publicly available data produced by the Neale lab.[5-6] GeneAtlas and Pheweb mitigate many of the privacy and security concerns mentioned above since no individual information is shared. There is no way to use summary statistics alone to gather information about any one individual. In addition, the size of these repositories are only fractions of the size of the individual level datasets, making transfer and storage of the data much more efficient. Finally, these services have already computed some of the most common summary statistics, which alleviates much of the computational burden on researchers.

However, while these approaches are promising and provide valuable insight, major questions abound about how to best leverage this summary-level information in more complex downstream analyses. While basic exploratory data analysis and data visualization are straightforward and commonplace, using pre-computed genotype-phenotype associations (summary statistics) to explore 'complex' phenotypes, which are functions of existing phenotypes present in a biobank, hasn't been previously investigated. For example, if a researcher is interested in phenotype Y, where $Y = f(\boldsymbol{y_1}, \boldsymbol{y_2}, \boldsymbol{y_3}, \dots \boldsymbol{y_m})$ and $\boldsymbol{y_1}, \boldsymbol{y_2}, \boldsymbol{y_3}, \dots, \boldsymbol{y_m}$ are existing phenotypes present in the biobank (with m being the number of phenotypes), is there a way to utilize the precomputed summary statistics from each linear model fit for each $\boldsymbol{y_1}, \boldsymbol{y_2}, \boldsymbol{y_3}, \dots, \boldsymbol{y_m}$ in order to make conclusions about the relationship between Y and genetic variation? This is the primary question of interest for this manuscript.

In particular, we begin by providing a framework for how to think about using summary statistics from individual phenotypes to investigate general classes of 'complex' phenotypes. We then illustrate how to utilize summary statistics for inferences about a complex phenotype which is a linear combination of an arbitrarily large set of individual phenotypes. Despite extensive literature review we have found little in the way of similar approaches thus most of our work has been built from the ground up. We validate our approach using both simulated data and real data from the Framingham Heart Study.

2. Methods

2.1 *Notation*

Throughout this paper we use y_{ij} to represent the phenotypes, where $i \in \{1, 2, \ldots, m\}$ with m being the number of phenotypes and j $\in \{1, 2, \ldots, n\}$ with n being the number of subjects. Similarly, x_j is used to represent the genotype. We use bolded letters (such as $\boldsymbol{y}_i$ and $\boldsymbol{x}$) to refer to a vector of values across all subjects. The term $\boldsymbol{y}_c$ is used to represent the linear combination of the $\boldsymbol{y}_i$'s ($\boldsymbol{y}_c = c_1\boldsymbol{y}_1 + c_2\boldsymbol{y}_2 + \ldots + c_m\boldsymbol{y}_m$) with the $c_i's$ being constants. For each linear regression model fit for $\boldsymbol{y}_i \sim \boldsymbol{x}$, we use the notation $\boldsymbol{y}_i = \beta_i\boldsymbol{x} + \alpha_i$, where β_i is the slope and α_i is the intercept. The standard error for β_i is represented by SE(β_i). We use $\boldsymbol{\beta_i}$ to represent all betas for phenotype *i* across all genotypes.

In addition, the following formulas are used frequently in this paper and should be kept in mind.

$$\beta_i = \frac{\mathrm{cov}(\boldsymbol{x}, \boldsymbol{y}_i)}{\mathrm{var}(\boldsymbol{x})} = \frac{\sum_{j=1}^{n}(x_j - \bar{x})(y_{ij} - \bar{y})}{\sum_{j=1}^{n}(x_j - \bar{x})^2} \tag{1}$$

$$\mathrm{SE}(\beta_i) = \frac{\sqrt{\frac{\sum_{j=1}^{n}\left(y_{ij} - \hat{y}_{ij}\right)^2}{n-2}}}{\sqrt{\sum_{j=1}^{n}\left(x_j - \bar{x}\right)^2}} \tag{2}$$

2.2. *Linear combination of two phenotypes using only summary statistics*

We will first show the formulas for the slope, intercept, and standard error of the slope in the case of a linear combination of two phenotypes ($\boldsymbol{y}_c = c_1\boldsymbol{y_1} + c_2\boldsymbol{y_2}$), where c_1 and c_2 are any constants. We will then show how these formulas generalize to an arbitrary number of phenotypes. In this portion of the paper we will only state the formulas – detailed derivations for each of the formulas can be found in the supplemental materials.

2.2.1. *Slope*

To determine the slope, $\hat{\beta}_c$, for the combined linear model of a linear combination of two phenotypes ($\boldsymbol{y}_c = c_1\boldsymbol{y_1} + c_2\boldsymbol{y_2}$), formula 1 was manipulated. We begin by inserting $\boldsymbol{y}_c = c_1\boldsymbol{y_1} + c_2\boldsymbol{y_2}$, into the least squares estimate of the slope:

$$\hat{\beta}_c = \frac{\sum_{j=1}^{n}(x_j - \bar{x})((c_1 y_{1j} + c_2 y_{2j}) - (\overline{c_1 \boldsymbol{y_1} + c_2 \boldsymbol{y_2}}))}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \quad (3)$$

After algebraic simplifications, $\hat{\beta}_c$ equals the same linear combination of the two phenotypes except with the slope instead of the phenotype:

$$\hat{\beta}_c = c_1\hat{\beta}_1 + c_2\hat{\beta}_2 \quad (4)$$

2.2.2. *Intercept*

To determine the y-intercept, $\hat{\alpha}$, for the combined linear model of a linear combination of two phenotypes, the mathematical formula for the least-squares estimate of the intercept was manipulated. As before, we begin by inserting $\boldsymbol{y}_c = c_1\boldsymbol{y_1} + c_2\boldsymbol{y_2}$, into the formula for the intercept in a standard least squares linear regression:

$$\hat{\alpha}_c = \overline{c_1\boldsymbol{y_1} + c_2\boldsymbol{y_2}} - \hat{\beta}_c\bar{x}. \quad (5)$$

Simplifying this equation shows that $\hat{\alpha}_c$ equals the same linear combination of the two phenotypes except with the intercepts instead of the phenotypes:

$$\hat{\alpha}_c = c_1\hat{\alpha}_1 + c_2\hat{\alpha}_2 \quad (6)$$

2.2.3. *Standard error of slope*

To determine the standard error of $\hat{\beta}_c$, SE($\hat{\beta}_c$), formula 2 was manipulated. $c_1y_{1j} + c_2y_{2j}$ was substituted for y_i and $(c_1\hat{\beta}_1 + c_2\hat{\beta}_2)x_j + (c_1\hat{\alpha}_1 + c_2\hat{\alpha}_2)$ for $\hat{y}_{ij}$. After some algebraic manipulation of the formula for SE($\hat{\beta}_c$), the formula was determined to be (see supplement 3 for details):

$$\mathrm{SE}(\hat{\beta}_c) = \sqrt{\mathrm{c}_1{}^2\mathrm{SE}(\hat{\beta}_1)^2 + \mathrm{c}_2{}^2\mathrm{SE}(\hat{\beta}_2)^2 + \frac{2\mathrm{c}_1\mathrm{c}_2}{n-2}\left(\frac{\mathrm{cov}(\boldsymbol{y_1}, \boldsymbol{y_2})}{\mathrm{var}(\boldsymbol{x})} - \hat{\beta}_1\hat{\beta}_2\right)} \quad (7)$$

2.3. *Linear combination of an arbitrary number of phenotypes using summary statistics*

Having provided the formulas for the linear combination of two phenotypes, we now explore the more general case of a linear combination of *m* phenotypes.

2.3.1. *Slope*

Following from the demonstration of the resulting $\hat{\beta}_c$ formula for the linear model for a linear combination of two phenotypes, it can be shown that the $\hat{\beta}_c$ from the linear regression of the linear combination of an arbitrary number of phenotypes is simply the same linear combination of the phenotypes except with $\hat{\beta}_i$'s from the simple linear regressions instead of the phenotype (complete

demonstration in supplement 1). Thus if there is a linear combination of m phenotypes the slope of the combined linear model is

$$\hat{\beta}_c = c_1\hat{\beta}_1 + c_2\hat{\beta}_2 + \cdots + c_m\hat{\beta}_m. \quad (8)$$

2.3.2. *Intercept*

Following from the demonstration of the resulting $\hat{\alpha}_c$ formula for the linear model in which there is a linear combination of two phenotypes, it can easily be seen that the $\hat{\alpha}_c$ from the linear regression of the linear combination of an arbitrary number of phenotypes is simply the same linear combination of the phenotypes except with the $\hat{\alpha}_i$'s from the simple linear regressions instead of the phenotypes (complete demonstration in the supplement 2). Thus if there is a linear combination of *m* phenotypes the intercept of the combined linear model is

$$\hat{\alpha} = c_2\hat{\alpha}_1 + c_2\hat{\alpha}_2 + \cdots + c_m\hat{\alpha}_m. \quad (9)$$

2.3.3. *Standard error of beta*

Following from the demonstration of the resulting $\mathrm{SE}(\hat{\beta}_c)$ formula for the linear model for a linear combination of two phenotypes, it can be demonstrated through induction that the $\mathrm{SE}(\hat{\beta}_c)$ from the linear regression of the linear combination of an arbitrary number of phenotypes is the following (complete demonstration in the supplement 4):

$$\mathrm{SE}(\hat{\beta}_c) = \sqrt{\left(\sum_{i=1}^{m} {c_i}^2 \mathrm{SE}(\hat{\beta}_i)^2\right) + \frac{2}{n-2}\left(\frac{\sum_{q=1}^{m-1}\sum_{r=q+1}^{m} c_q c_r \mathrm{cov}(\boldsymbol{y}_q, \boldsymbol{y}_r)}{\mathrm{var}(\boldsymbol{x})} - \left(\sum_{q=1}^{m-1}\sum_{r=q+1}^{m} c_q c_r \hat{\beta}_q \hat{\beta}_r\right)\right)} \quad (10)$$

2.3.3.1. *Estimating terms in the equation for the standard error of beta*

All of the terms in formula 10 for the standard error of the combined $\hat{\beta}$ are summary level statistics. While this eliminates the need for individual level data and thus alleviates many of the previously-discussed privacy issues, there are two summary statistics within that formula that aren't often publicly available. In particular, the covariances between each unique pair of phenotypes and the variance of $\boldsymbol{x}$ are not frequently provided. As such, it would be helpful if there were methods for estimating these terms from the information that is readily available.

We first explore a method for estimating the covariance between a given pair of phenotypes. Since linear models have already been run on the entire data set, slopes are given for each genotype-phenotype combination. Thus, we hypothesized that the correlation between two of the response variables could be estimated by finding the correlation between the betas for the first phenotype and the betas for the second phenotype. However, the quantity needed for the standard

error formula is covariance. Therefore, to find the covariance, we propose the following approximation:

$$cov(y_1, y_2) = cor(y_1, y_2) * \sqrt{var(\boldsymbol{y_1})var(\boldsymbol{y_2})} \approx cor(\boldsymbol{\beta_1}, \boldsymbol{\beta_2}) * \sqrt{var(\boldsymbol{y_1})var(\boldsymbol{y_2})} \quad (11)$$

Note that this, in turn, requires that we have the variance of $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$.

Next, we explore a method for estimating the variance of $\boldsymbol{x}$. Because we can model $\boldsymbol{x}$ by the binomial distribution, the variance of $\boldsymbol{x}$ can be estimated using the minor allele frequency (MAF). Thus, by using the formula for the variance of a binomial distribution we can accurately estimate the variance of $\boldsymbol{x}$ using the known minor allele frequency.

$$2MAF(1 - MAF). \quad (12)$$

While this approximation is close to the true value, the accuracy of the estimate changes with the Hardy-Weinberg equilibrium (HWE) p-value. In the next section we explore this using simulations.

2.4. *Simulations*

2.4.1. *Estimation of covariance of y's simulations*

To test the hypothesis for our covariance estimate, simulations were conducted in R.[7] We wrote a function for performing these simulations, which generated two phenotypes and a large number of genotypes. The parameters altered from trial to trial were the number of observations, the number of genotypes, the covariance between the two phenotypes, and the variance of each of the two phenotypes.

2.4.2. *Estimation of variance of* $\boldsymbol{x}$ *simulations*

To check the accuracy of the variance of $\boldsymbol{x}$, simulations were run in R. Ten thousand genotypes from 1,000, 10,000, 100,000, and 500,000 subjects were generated using a binomial distribution. The genotypes were of varying minor allele frequencies and varying Hardy-Weinberg equilibrium p-values. For each genotype the following statistics were calculated: MAF, HWE p-value, the observed variance, estimated variance, and the difference between the observed variance and the estimated variance. At HWE p-value thresholds of 0.05, 0.5, 0.75, 0.90, and 0.99, the mean difference between the observed variance and the estimated variance of genotypes, and the standard deviations of those differences of the genotypes that met or exceeded the thresholds were also calculated.

2.5. *Real data analysis*

Previous genome wide association studies, investigated the association between 425,380 SNP's and red blood cell fatty acid (RBC FA) levels indicative of cardiovascular health using data from the offspring cohort (n=2384) of The Framingham Heart Study as we've done in other recent publications. [8-11] Two of the RBC FA included were Docosahexaenoic acid (DHA) and Eicosapentaenoic acid (EPA). The sum of DHA and EPA is reported as the omega3 index (O3I).

In the studies, genome wide association analyses were conducted for DHA, EPA, and O3I using residual models adjusting for age, sex, and familial relationships. We will use this data to demonstrate our method. We will show the accuracy of the slope and standard error of the slope calculated using the summary statistics from the individual EPA and DHA models and the method presented in this paper as compared to the slope and standard error that is obtained from running the entire linear model specifically on the O3I. Please refer to the studies cited for more information about the significance of their findings, the collection of red blood cell fatty acids and the Framingham cohort.[8-11]

3. Results

3.1. *Estimating the covariance of phenotypes*

We begin by investigating the performance of our proposed estimation (formula 11) for the covariance of phenotypes (y_i's). As seen in Table 1, our results suggest that the error in our approximation is highest when the correlation between $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$ is close to 0. As the correlation between a pair or $\mathbf{y_i}$'s increases, the standard deviation of the error in the estimated correlation decreases.

The other two parameters (number of genotypes and number of observations) had little to no impact on the standard deviation of the errors (detailed results not shown).

Table 1. This table shows the results from the simulations. The "Correlation" column lists the correlation at which the data was generated. The other two columns display the mean and standard deviation of the error of the estimate.

Correlation	Mean error of estimated correlation	Standard deviation of error of estimated correlation
0	-0.000486	0.050
0.3	0.000400	0.045
0.75	6.23E-05	0.022
0.9	0.000282	0.0096

3.2. *Estimating variance of genotype*

The detailed results of the variance of $\boldsymbol{x}$ simulations can be found in Table 2. Overall, the difference between the observed variance of $\boldsymbol{x}$ and the estimated variance of $\boldsymbol{x}$ across all simulated genotypes was small with a mean of 0.000043 and standard deviation of 0.0064. Thus as the length of the genotype gets larger, the difference between the observed and estimated variances seems to go to zero. While the mean differences are quite small, they are nearly all positive indicating that we are underestimating the variance. Because the standard error formula (formula 7) divides by the variance our standard error will be inflated and thus this method will be slightly conservative. Additionally, as can be seen in Table 2 and Figure 1, genotypes with larger HWE p-values have differences between the observed and estimated variances that are closer to zero.

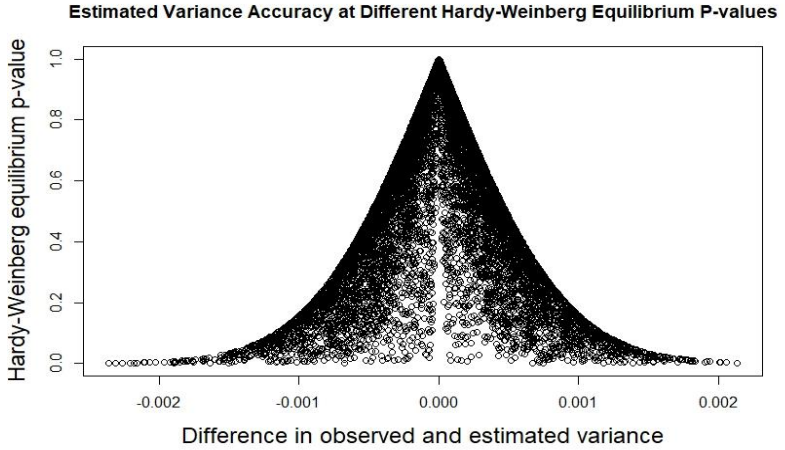


Fig. 1. This plot shows the results of the simulation of 10,000 genotypes from 500,000 subjects. The Hardy-Weinberg equilibrium p-value is on the y-axis and the difference in the variance is on the x-axis.

Table 2. Results for variance of $\boldsymbol{x}$ simulations, with 10,000 genotypes simulated for 500,000, 100,000, 10,000 and 1,000 individuals.

Number of individuals	P-value	Number of genotypes that fall at or above p-value threshold	Mean of the difference between observed and estimated variance	Lower bound of Wald confidence interval for mean	Upper bound of Wald confidence interval for mean
500,000	≥ 0.99	104	1.4E-06	-7.1E-06	1.0E-05
	≥ 0.90	1042	2.6E-06	-7.8E-05	8.3E-05
	≥ 0.75	2510	7.5E-07	-2.0E-04	2.0E-04
	≥ 0.50	5002	4.5E-06	-4.1E-04	4.2E-04
	≥ 0.05	9494	9.6E-06	-9.3E-04	9.5E-04
	All	10000	4.1E-06	-1.1E-03	1.1E-03
100,000	≥ 0.99	98	4.3E-06	-1.3E-05	2.2E-05
	≥ 0.90	1025	1.1E-06	-1.7E-04	1.8E-04
	≥ 0.75	2551	6.8E-06	-4.4E-04	4.5E-04
	≥ 0.50	5015	2.3E-06	-9.2E-04	9.3E-04
	≥ 0.05	9497	6.9E-06	-2.1E-03	2.1E-03
	All	10000	1.2E-05	-2.4E-03	2.4E-03
10,000	≥ 0.99	94	3.7E-05	-2.6E-05	1.0E-04
	≥ 0.90	999	4.5E-05	-5.2E-04	6.2E-04
	≥ 0.75	2481	5.1E-05	-1.4E-03	1.5E-03
	≥ 0.50	4938	5.0E-05	-2.8E-03	2.9E-03
	≥ 0.05	9501	5.5E-05	-6.8E-03	6.7E-03
	All	10000	-8.4E-05	-7.7E-03	7.5E-03
1,000	≥ 0.99	114	3.8E-04	1.2E-04	6.4E-04
	≥ 0.90	962	3.9E-04	-1.4E-03	2.2E-03
	≥ 0.75	2439	3.4E-04	-4.2E-03	4.8E-03
	≥ 0.50	4963	4.1E-04	-8.8E-03	9.6E-03
	≥ 0.05	9452	1.8E-04	-2.1E-02	2.1E-02
	All	10000	2.4E-04	-2.4E-02	2.4E-02

3.3. *Real data results*

3.3.1. *Using exact formulas*

We first consider the accuracy of adding the two residual models after adjusting for covariates. It appears that the predictions for the slope of the combined linear model made using prediction $\hat{\beta}_{EPA} + \hat{\beta}_{DHA} = \hat{\beta}_{RO3I}$ were accurate. The predictions of the model adjusting for covariates after addition ($\hat{\beta}_{O3I}$) had a mean difference of 0.0000469 and a standard deviation of 0.00204. Figure 2 shows the observed values of $\hat{\beta}_{O3I}$ plotted against the estimate values, and appears to show that the estimate is relatively accurate on the entire range of true slopes.

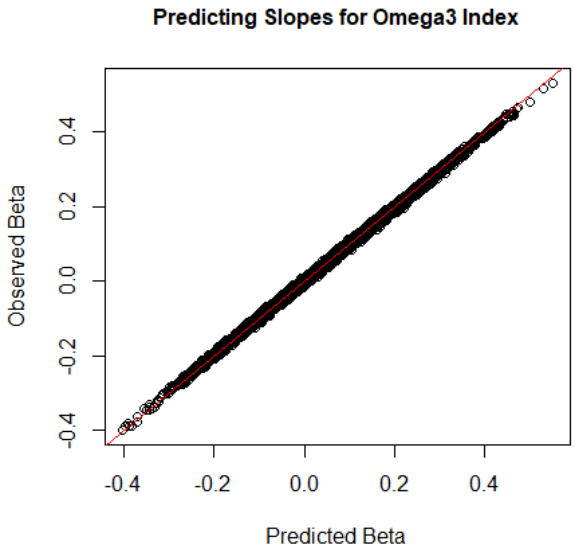


Fig. 2. The observed beta values are on the y-axis and the predicted beta values are on the x-axis. This shows the accuracy of the combined beta formula.

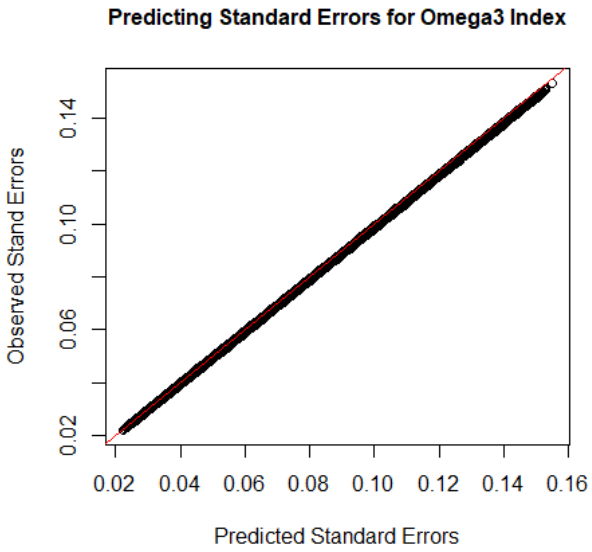


Fig. 3. The observed standard errors for the beta is on the y-axis and the predicted standard errors of the beta is on the x-axis. This shows the accuracy of our standard error estimate.

Using formula 7 for predicting the standard error for the β_{RO3I}, there was a mean error of -0.00000177 with a standard deviation of 0.00004717. When comparing the estimate for standard error to the actual O3I standard error, the mean error was 0.00058 with a standard deviation of 0.000276. Figure 3 demonstrates that when applying the covariates separately to the models DHA and EPA we see a slight over prediction of the standard errors.

3.3.2 *Estimating covariance of the* ***y's***

Using the method described in 2.4 the estimated correlation between EPA and DHA was 0.707 while the actual correlation between the two variables is 0.682. The error between the true value and the predicted value will in turn lead to a slightly inflated standard error estimate.

3.3.3 *Estimating the variance of* ***x***

When using our estimate of the variances of the genotype in the standard error equation, we see some increased variation in the estimations, as seen in Figure 4. However, filtering by Hardy Weinberg equilibrium p-value (eliminate genotypes with HWE p-values less than 0.000001 as

per GWAS standard)[12] removes all of the extreme variation between estimated and predicted estimates of the variation of the genotypes.

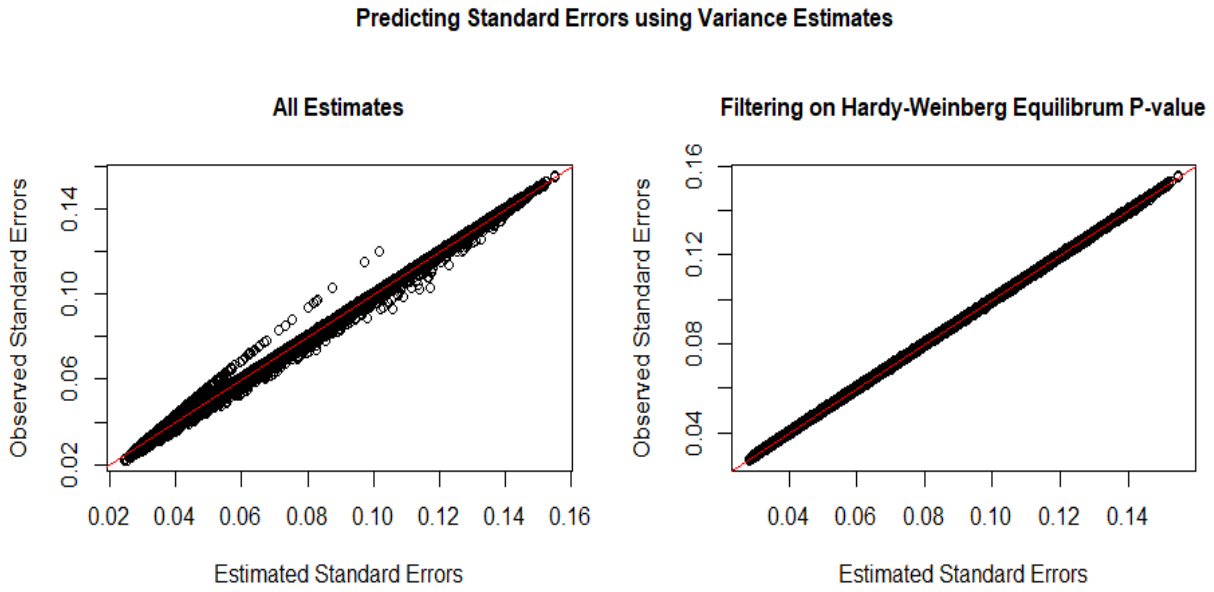


Fig 4. The graph on the left demonstrates the accuracy of the standard error estimates for the beta values using all SNP's in the data set. The graph on the right filters by Hardy-Weinberg equilibrium p-value of 0.000001, which removes most of the less accurate predictions.

3.3.4 *Analysis of p-value*

We examine $-\log_{10}$ p-value plots to see the overarching effect the method presented in this paper has on the significance of the study. In this analysis we compare the p-values obtained from using our summary statistic model with the true p-values from the linear model before adjusting for covariates. When estimating the variance of the genotype we filtered by a Hardy-Weinberg equilibrium p-value of 0.000001.

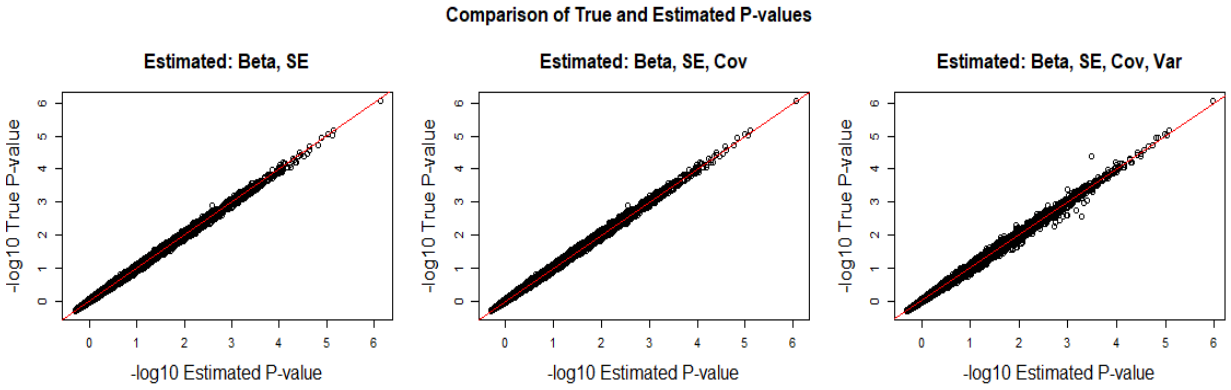


Fig 5. The graph on the left demonstrates the accuracy of the negative log of the p-value when our formulas for the slopes and standard errors are used with the true variance of $\boldsymbol{x}$ and covariances between phenotypes. The middle graph shows the accuracy when covariance of the $\boldsymbol{y's}$ is estimated using our estimation. The graph on the right depicts the accuracy of the p-values when the covariance of the $\boldsymbol{y's}$ and the variance of $\boldsymbol{x}$ are estimated using our given estimates.

3.3.5 *Careful analysis of top hits*

One of the important aspects of using summary level statistics is that it will not greatly affect the most significant genotype phenotype associations. As seen in supplemental tables 5, 6, and 7 the differences in β, SE(β) and overall p-values between the summary statistic model and the traditional model is minimal.

4. Discussion

We have demonstrated how to accurately estimate the strength of association for a linear combination of an arbitrary number of individual phenotypes with a single genotype of interest using only commonly available summary statistics from large biobanks. In addition, we have provided a mathematical overview of why these relationships hold, demonstrated how to estimate these values from summary statistics and distributions of summary statistics, and then evaluated their performance on both simulated and real data.

Practically, we have now provided a tool for researchers to perform genome-wide and related analyses on linear combinations of phenotypes using only summary statistics, which has the potential to dramatically reduce computational time and storage, simplify data transfer, and grossly mitigate privacy and security concerns, especially for large biobank-style datasets. For example, in our data analysis of The Framingham Heart Study the Rdata file size needed to run the analysis was reduced from 1.2 GB to 0.04 GBs. Notably, the reduction in file size and processing time should increase significantly with an increased sample size. While linear combinations of phenotypes are a powerful tool (e.g., averaging multiple measurements of a trait of interest), future work is needed to explore more general ways of combining phenotypes which will have broader applicability. For example, multiplicative combinations of phenotypes ($\boldsymbol{y_1 * y_2}$ or $\boldsymbol{y_1/y_2}$) and exponentiated phenotypes are also a powerful and common class of complex phenotypes (e.g., BMI = Weight/Height^2).). If future work is able to establish a similar class of methods for multiplicative phenotypes as has been shown in this manuscript for linear combinations, we would then be in position to also derive general methods for 'logical' combinations of dichotomous phenotypes. Logical combinations can be expressed as arithmetic operations. The 'and' operation can be expressed as $\boldsymbol{y1 * y2}$ and the 'or' operation can be expressed as $(\boldsymbol{y1 + y2}) - (\boldsymbol{y1 * y2})$. Future work also includes consideration of multi-allelic models, the impact of different assumptions in models/software creating summary statistics on downstream inference using our proposed method, and direct comparison and evaluation of changes in computation time.

Some limitations of our method are worth noting. First, we have been able to accurately estimate the variance of x ($\boldsymbol{x}$ in other words, the genotype) using the variance formula for a binomial distribution and the minor allele frequency. This estimate has been verified through simulations and we have shown that as the genotypes reach perfect Hardy-Weinberg equilibrium the difference between the observed and estimated variances of x approaches 0. While in practice,

variants out of HWE are removed from the data, variants that are 'nearly' out of HWE using standard GWAS quality thresholds[11] (e.g., HWE p-value < $1x10^{-6}$) may experience more noise in downstream estimates. Secondly, while our simulations and real data application are reasonably comprehensive, application to additional datasets and consideration of additional simulated datasets (e.g., with different sample sizes; different proportions of and distributions of missing data; different levels of correlation between phenotypes) is recommended.

The use of summary statistics from large biobanks in downstream statistical analyses offers great promise to address numerous hurdles in the use of biobank data and dramatically increase the opportunity to leverage biobanks to understand the etiology of complex human diseases. We have provided precise equations to leverage summary statistics for linear combinations of phenotypes. The method presented in this paper sets the essential foundation and provides a necessary building block for being able to investigate the genetic associations of millions of complex phenotypes with summary statistics alone. Future work is needed to explore multiplicative and other more complex ways to combine phenotypes to provide a complete approach to phenotype combinations.

Supplemental materials can be found here:
http://www.nathantintle.com/supplemental/supplement_leveraging_summary_statistics.pdf

Acknowledgments

The authors of this work were partially supported by a grant from NIH/NHGRI (2R15HG006915) and Dordt College.

References

1. C. Sudlow *et al.*, *PLoS Med* **12**, e1001779 (2015).
2. B Huppertz *et al.*, *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics*, 317-330 (2014).
3. R Heatherly, *The Journal of Law, Medicine &* Ethics **44,** 156-160 (2016).
4. E.M. Jones *et al.*, *Norsk Epidemiologi* **21**, 231-239 (2012).
5. O. Canela-Xandri, K. Rawlik and A. Tenesa, *bioRxiv* preprint (2017). doi:10.1101/176834
6. Abbot, Liam. *Et al., biobank improving the health of future generations*, www.nealelab.is/uk-biobank/. Accessed 6 Aug. 2018
7. R Development Core Team, *R Foundation for Statistical Computing* (2008).
8. A. Kalsbeek *et al.*, *PLoS One* **13**, e0194882 (2018).
9. N. L. Tintle *et al.*, *Prostaglandins Leukot Essent Fatty Acids* **94**, 65-72 (2015).
10. J. Veenstra *et al.*, *Nutrients* **9**, (2017).
11. W.S. Harris *et al.*, *Atherosclerosis* **225(2)**, 425-431 (2012).
12. P. Sasieni, *Biometrics* **53**, 1253-1261 (1997).

Protecting Genomic Data Privacy with Probabilistic Modeling

Sean Simmons
Stanley Center, Broad Institute,
Cambriadge, MA 02142, USA
E-mail: ssimmons@broadinstitute.org

Bonnie Berger *
CSAIL and Department of Mathematics, MIT
Cambriadge, MA 02142, USA
E-mail: bab@csail.mit.edu

Cenk Sahinalp *
Department of Computer Science, Indiana University,
Bloomington, Indiana 47405, USA
E-mail: cenksahi@indiana.edu

The proliferation of sequencing technologies in biomedical research has raised many new privacy concerns. These include concerns over the publication of aggregate data at a genomic scale (e.g. minor allele frequencies, regression coefficients). Methods such as differential privacy can overcome these concerns by providing strong privacy guarantees, but come at the cost of greatly perturbing the results of the analysis of interest. Here we investigate an alternative approach for achieving privacy-preserving aggregate genomic data sharing without the high cost to accuracy of differentially private methods. In particular, we demonstrate how other ideas from the statistical disclosure control literature (in particular, the idea of disclosure risk) can be applied to aggregate data to help ensure privacy. This is achieved by combining minimal amounts of perturbation with Bayesian statistics and Markov Chain Monte Carlo techniques. We test our technique on a GWAS dataset to demonstrate its utility in practice. An implementation is available at https://github.com/seanken/PrivMCMC.

Keywords: Genomic Privacy; GWAS; MCMC

1. Introduction

There is a tension in modern human genomics between data sharing and privacy concerns.[1] On the one hand, genomic data holds the promise of greatly improving human health, so the ability to share it is paramount. On the other hand, our genomes are some of the most private pieces of information we have, and the risks of sharing it openly are far from understood. Even releasing aggregate genomic data (statistics calculated on an entire group of individuals, such as odds ratios or minor allele frequencies - MAF) can raise privacy concerns.[2–5]

*Corresponding Authors.

Numerous approaches have been suggested for enabling the sharing of aggregate genomic data while respecting participants privacy.[1,6] In practice, most data is either open access (posted online with minimal privacy considerations) or controlled access (only shared with trusted individuals). Recently, there has been a push to develop alternative methods for sharing this data publicly while still preserving privacy.[3,7,8] Most of these methods rely on strong assumptions about the background population (such as independent SNPs, lack of stratification, etc). As such, it is unclear how accurate a measure they provide on real world populations. Moreover, it is unclear how to extend them to more general classes of statistics (beyond MAF, etc). Alternatively, there have been methods suggested that give strong privacy guarantees with little to no assumptions about the underlying data (namely differential privacy, see section 2.5 for a definition).[9–15] Though these methods are effective at sharing small amounts of data while preserving strong privacy guarantees, current methods become inaccurate when scaled to more than a few genomic loci.[14,15]

Here we introduce a method for preserving privacy that begins to address both concerns. We build upon ideas from the statistical disclosure literature. In particular, our approach is based off of measuring the risk of reidentification using Bayesian approaches.[16] Our method aims to protect private disease status information for participants in GWAS studies, while making minimal assumptions about how the genomic data was generated (in particular, we assume that the individual trying to learn private information, known as the adversary, does not have any information allowing them to distinguish cases from controls a priori), and allowing release of more accurate statistics than that achieved by current differentially private methods. Moreover, unlike differential privacy, it is straightforward to apply our approach to almost any statistic of interest.

2. Methods

2.1. *The Model*

In the model underlying our method, we are given two pieces of information: the genotype data of each individual, and their disease status. Let $D = \{d_1, \cdots, d_n\}$, where $d_i \in \{0,1,2\}^m$ for $i = 1, \ldots, n$, be the genotype data of all individuals in our study. Let $y = (y_1, \cdots, y_n) \in \{0,1\}^n$ be the vector of disease statuses ($y_i = 1$ if individual i has the disease, $y_i = 0$ otherwise). Let n_1 be the number of times a 1 occurs in y (number of cases), n_0 the number of times 0 occurs (number of controls), n the total number of individuals, and m the number of SNPs we want to share aggregate data about.

Let Y be a random variable which takes values in $\{0,1\}^n$. This variable represents the adversaries prior belief about how likely each individual in the study is to be a case or control.

In particular, we will define it so that $Pr(Y = y')$ is equal for all $y' \in \{0,1\}^n$ such that y' with exactly n_1 ones, and 0 otherwise; i.e.:

$$Pr(Y = y') = \frac{1}{\binom{n}{n_0}}$$

This represents the prior probability of each individual in the study being either a case or control, assuming all such assignments are equally likely. In essence, this model is meant to

represent an adversary who knows everything about the study (genotypes, etc) except has no idea who is a case and who is a control.

2.2. *The Privacy Approach*

We want to release some statistics based on y and D. In order to protect participants privacy, however, we add a small amount of noise to them.

More formally, consider a statistic X that takes in both genotype data and disease status information, and outputs a vector of statistical information in $\mathbb{R}^k$ for some integer k. We want to release $X(y, D)$ while preserving privacy. In order to do this, we instead release $X + \epsilon$, where $\epsilon = (\epsilon_1, \cdots, \epsilon_k)$ is a random noise term.

For our purposes, we will assume that each ϵ_i is either a Laplacian random variable or a truncated Laplacian random variable. This choice is so as to be consistent with the Laplacian mechanism, a standard diferentially private technique.[17] In particular, for given parameter λ and bound δ, we have that:

$$Pr(\epsilon_i = z) \propto \begin{cases} exp(-\frac{|z|}{\lambda}), & -\delta < z < \delta \\ 0 & \text{otherwise} \end{cases}$$

Here λ controls the variance, and δ the maximum/ minimum amount of noise added. When δ is set to infinity, we get a standard (unbounded) Laplacian random variable, represented by $Lap(0, \lambda)$.

2.3. *The Privacy Measure*

Having specified a method for releasing privacy-preserving statistics, we want to be able to measure how much privacy is lost upon releasing them. Instead of using differential privacy based measures, however, we suggest an alternative approach based on prior probability, specified by the random variable Y. The measure of privacy we use is based on the assumption that anyone looking at the statistics does not know which participants are in the case versus the control cohort. In particular, we consider all possible permutations of participants disease status, and assume that all such assignments are equally likely from an outsiders point of view. This probabilistic model is inspired by the model used to justify k-anonymity (a standard technique in the statistical disclosure literature) and related techniques.[16,18,19]

For a given statistic X, genetic dataset D, and a disease status vector y, we want to release χ, a noisy version of X, defined as:

$$\chi(y, D) = X(y, D) + \epsilon$$

In order to measure the disclosure risk of releasing this data, we consider, for the ith individual, the probability

$$Pr(Y_i = 1 | \chi(Y, D) = \chi(y, D))$$

This can be seen as measuring the probability that the adversary believes the ith individual has the disease based on the perturbed statistic $\chi(y, D)$. Our goal is to keep this quantity as small as possible, particularly when $y_i = 1$ (when the individual has the disease). It is worth noting that, for a randomly chosen i, this has an expected value of $\frac{n_1}{n}$. Note that we do not consider the probability that $Y_i = 0$, since in general revealing that a given individual does not have a disease is not considered a privacy breach. This decision is consistent with the membership privacy idea used in previous work,[10,20] though our method can be easily modified to consider the probability $Y_i = 0$ as well.

2.4. *Estimating the Posterior*

In theory, we would like to have an exact estimate of $Pr(Y_i = 1|\chi(Y, D) = \chi(y, D))$. In practice, however, there does not seem to be an easy way to do this. Short of brute force, there does not seem to be a general method that works for more than a handful of statistics. As such, we use a form of Markov Chain Monte Carlo (MCMC) known as the Metropolis-Hastings algorithm[21] to estimate this probability.

In order to achieve this, we first draw $y' \sim Pr(Y = y'|\chi(Y, D) = \chi(y, D))$ using a two step process.

(1) Pick $y' \sim Pr(Y = y'|X(Y, D) + Lap(0, \lambda) = \chi(y, D))$ using Metropolis-Hastings, where $Lap(0, \lambda)$ is a k-dimensional unbounded Laplacian variable. The proposal distribution, q, we use to do this is chosen so that $q(y_1, y_2) \propto 1$ if $|y_1 - y_2|_1 = 2$ and equals 0 otherwise.

(2) If $\max_{\forall i} |X_i(y', D) - \chi_i(y, D)| < \delta$ return y', else go back to the previous step

Here, the proposal distribution dictates the probability of each step in the random walk used for MCMC. Our choice ensures each such jump corresponds to swapping one case and one control.

Note that, if the noise is not truncated, then step 1 suffices. We can use the above algorithm to generate a series of samples which can be used to estimate $Pr(Y_i = 1|\chi(Y, D) = \chi(y, D))$. It can be shown that this approach results in a correct asymptotic estimate of the probabilities of interest. Note that we use 100,000 steps as burn-in with 10,000 steps between samples in the Metropolis-Hastings algorithm.

2.5. *Comparison to differential privacy*

Differential privacy[17] is a common definition of privacy in the cryptographic literature. Formally:

Definition 1. *A random function F is ϵ-differentially private, if for all datasets D and D' that differ in exactly one entry, and for all sets S, we have that*

$$Pr(F(D) \in S) \leq exp(\epsilon) Pr(F(D') \in S)$$

Note that it is hard to directly compare the privacy guarantees of differential privacy to the privacy guarantees provided by risk based methods, since there is no clear correspondence

(they have different assumptions, one gives a risk bound and one a risk estimate, etc). In an attempt to overcome this qualitative difference, we note that, under reasonable assumptions (namely that the distribution is a mutually independent distribution,[10] assumptions we believe reasonable to assume in our setting), differential privacy can be thought of as ensuring that, for any dataset D_1 and any $d \in D_1$

$$log(Pr(d \in D_1|F(D_1))/Pr(d \in D_1)) \leq \epsilon$$

In our setting, if we take D_1 to be the case cohort and our statistic of interest to be the MAF (see Section 2.8), this corresponds to

$$log(Pr(Y_i = 1|\chi(Y, D) = \chi(y, D))/Pr(Y_i = 1)) \leq \epsilon$$

Therefore, in order to compare differential privacy— in particular a well-known differentially private mechanism known as the Laplacian mechanism[17]— to our method, we compare this upper bound to the maximum value of our MCMC based estimate of $log(Pr(Y_i = 1|\chi(Y, D) = \chi(y, D))/Pr(Y_i = 1))$ taken over all individuals in the case cohort. The smaller this quantity, the smaller the risk relative to the background risk. Though not a perfect comparison, it gives us some idea of how our method compares to differential privacy. To vary epsilon in our analysis the number of SNPs is varied from 10 to 50 SNPs, using noise parameter $\lambda = .01$, and the log probability ratios are calculated with both our approach and the Laplacian mechanism.

2.6. *Error in MCMC*

To measure the error in our MCMC approach, we consider a dataset with 1000 individuals, 50 cases and 950 controls, each with 20 SNPs. By error, we mean the difference between our estimated probabilities and the theoretical ones. For each SNP, the controls have 0 copies of the minor allele, and the cases have 2 copies. For this dataset, it is easy to calculate the marginals of interest using simple combinatorial and probabilistic arguments. As such, we are able to compare the exact marginals on this dataset with the marginals estimated by our method.

2.7. *The Data*

In order to test our method, we use genotype data from Plenge et al.[22] This data is from a rheumatoid arthritis dataset. In most of our tests, we used 50 random cases and 950 random controls. Note that we did not use the full dataset, since our method requires that the controls out number the cases by a large margin—in particular, it is aimed at datasets without ascertainment biases (e.g. studies that have the same percentage cases as the background population). Otherwise, simply knowing someone is in the dataset reveals that they have a greater risk of having the disease being studied than someone in the general population. Though historically GWAS have been enriched for cases compared to the background population, recent population level datasets (such as the UK biobank, direct to consumer studies, etc) have started to change this, a trend likely to continue. Note that this dataset was used in all the results below, except for Fig 2 and Fig 4 where we use data from a GWAS of bladder cancer,[23] choosing 50 cases and 950 controls, and in Section 3.4 where simulated data was used. For all results we used randomly chosen SNPs with MAF greater than .05 and no missing values.

2.8. *Statistics of Interest*

We test our method on the MAF of the case cohort and the log odds ratio from the entire dataset. The MAF is defined as:

$$maf(y, D) = \frac{1}{2n_1} \sum_i y_i d_i$$

while the log odds ratio is defined, for the jth snp, as:

$$logOdds_j(y, D) = log\left(\frac{a_j(1-b_j)}{b_j(1-a_j)}\right)$$

where $a_j = maf_j(y, D)$ and $b_j = maf_j(\bar{1} - y, D)$.

3. Results

3.1. *The Privacy Cost of Releasing More Data*

We first apply our method to the minor allele frequency (MAF) of the case cohort. In particular, we use a set of 50 cases and 950 controls from a Rheumatoid Arthritis GWAS (see

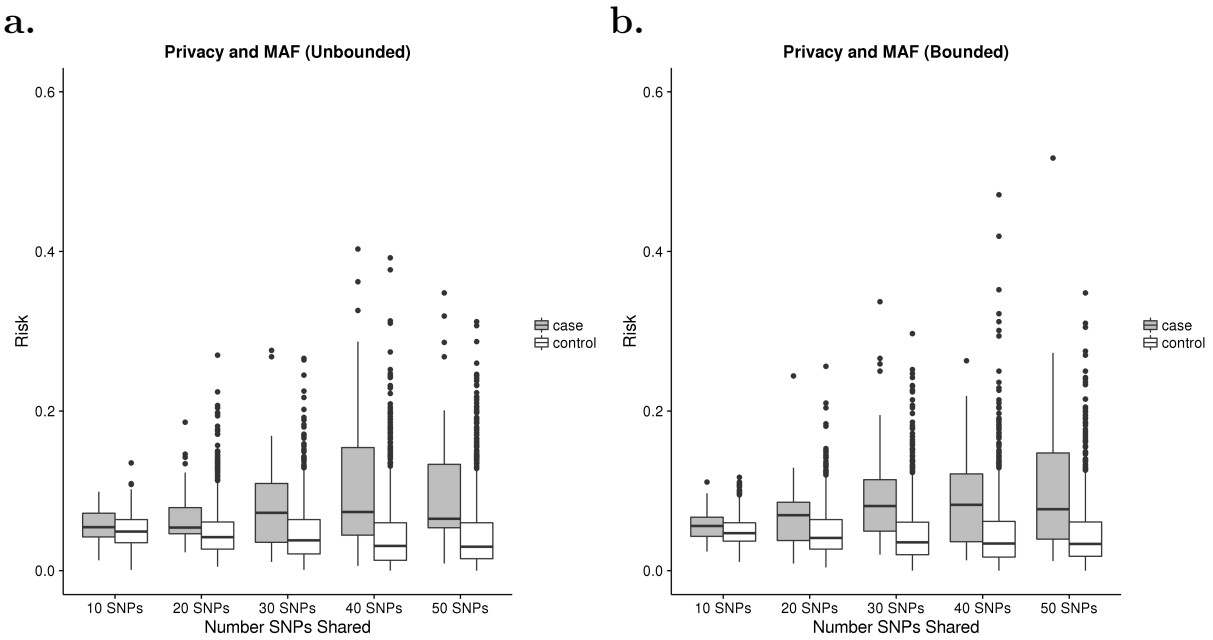


Fig. 1. Number of SNPs versus privacy. We compare the disclosure risk versus the number of SNPs whose MAF data we release from a rheumatoid arthritis GWAS, with both (a) unbounded and (b) bounded noise. This demonstrates that, unsurprisingly, privacy is greatly affected by the amount of data released. Less intuitively, we see most of this privacy loss is suffered by a few individuals, rather than being evenly shared between all individuals in the cohort. More importantly, it demonstrates the utility of disclosure risk to measure the level of privacy concerns. The risk is calculated on a dataset of 50 cases, 950 controls, with the number of SNPs released varying between 10 and 50 SNPs, with noise $\lambda = .01$. The bounded noise is bounded by $\delta = .05$.

Methods). In order to ensure privacy, we add Laplacian noise with parameter $\lambda = .01$ to the output MAF for each SNP (see Methods). This corresponds to an expected error in the returned statistic of .01.

In this setting, we used our method to measure the amount of privacy lost when releasing the MAF from various numbers of SNPS between 10 and 50 (Fig 1a). We look at randomly chosen SNPs, though one could use similar techniques to look at SNPs of particular interest (such as those with low p-values). Unsurprisingly, we see that, as the number of SNPs increases, the disclosure risk (that is to say the amount of privacy loss) increases for individuals in the case cohort. Less intuitively, we see that, though the average risk for cases increases slowly as more data is released, there are a few outliers with much higher risk. This suggests the possibility of removing these individuals in an attempt to lower the risk of releasing the data (assuming that doing so does not introduce bias into the results). We also tested our method on another GWAS of bladder cancer patients and found similar results (Fig 2).

Many practitioners are uncomfortable with the idea of adding unbounded noise to a statistic, even in the name of ensuring privacy. Unlike differential privacy, our method is flexible enough to allow us to bound the error of our output. As such, we considered adding bounded Laplacian noise to the MAF, to see how bounding the noise effects privacy (see Methods). This ensures that the released statistic is within a window of length .1 centered around the true MAF. Using this noise, we ran the same experiment that was run for Laplacian noise above (Figure 1b). Again, we see that disclosure risk increases as the number of SNPs released increases, most notably in a few outliers.

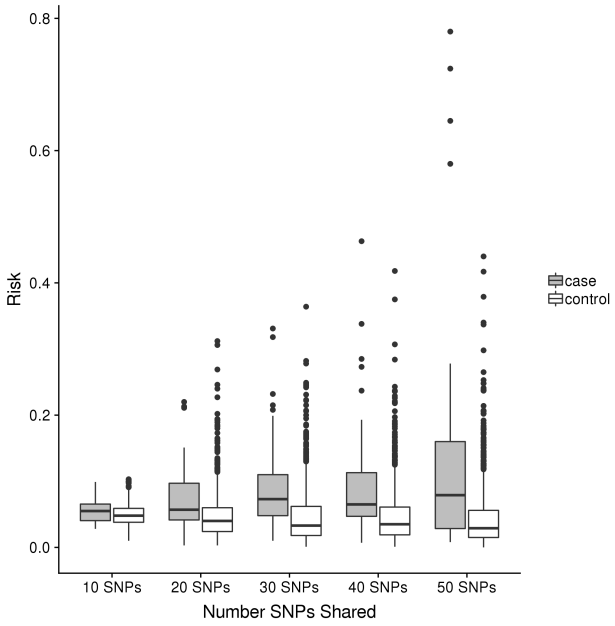


Fig. 2. Number of SNPs versus privacy. We compare the disclosure risk versus the number of SNPs whose MAF data we release, with unbounded noise on a bladder cancer GWAS dataset. The results are, unsurprisingly, similar to those in the Rheumatoid Arthritis dataset.

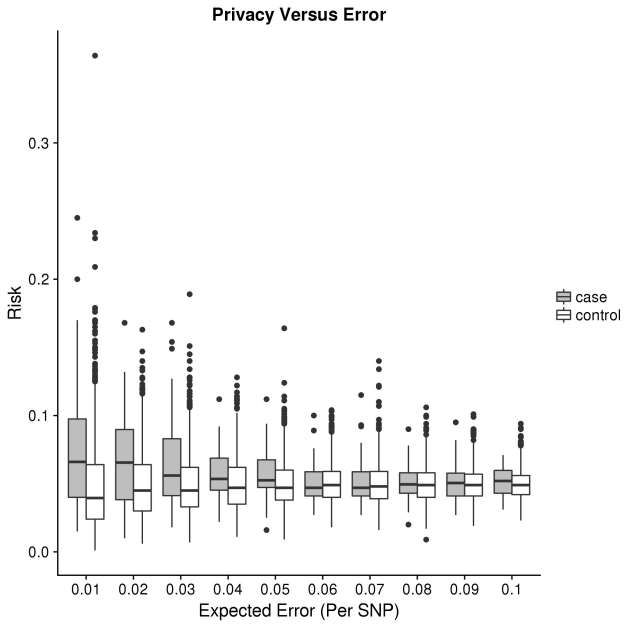


Fig. 3. Privacy versus accuracy. We compare the effect of the amount of noise versus disclosure risk when adding unbounded noise. We see that, as the noise increases, disclosure risk decreases, so privacy level increases. This increase in privacy, however, sees diminishing returns with fairly low errors, suggesting that adding large amounts of noise might not be needed. The risk is calculated on a rheumatoid arthritis GWAS dataset of 50 cases, 950 controls, with 25 SNPs released, and noise varying between $\lambda = .01$ and $\lambda = .1$.

3.2. *Accuracy Versus Privacy*

We are also interested in exploring the trade off between accuracy and privacy. The larger the amount of noise added to our statistics, the less privacy risks are encountered. At the same time, the more noise that is added, the less accuracy that is achieved. As such, we compared the the amount of noise added to the level of risk. In particular, we considered releasing the MAF for 25 SNPs with Laplacian noise added to them. We varied the expected error per SNP between .01 and .1 (Fig 3). This was achieved by varying the λ parameter of the Laplacian distribution. We see that, as the accuracy increases, the risk of disclosure increases as well. The trade off between accuracy and privacy is important, since it can help determine which choice of noise parameter is reasonable in any particular setting. In particular, we see that the privacy gains of increasing the error level off quickly, suggesting that there is not much incentive to add large amounts of noise to the data.

It is also of interest to figure out the amount of privacy lost when publishing unperturbed statistics. Unfortunately, our method relies on the addition of noise to calculate the risk (to enable MCMC). Having said that, as the noise approaches zero, the risk should approach that of releasing the unperturbed statistics. As such, the risk we see in Fig 3 for low levels of noise should give us some idea about the risk of the unperturbed dataset.

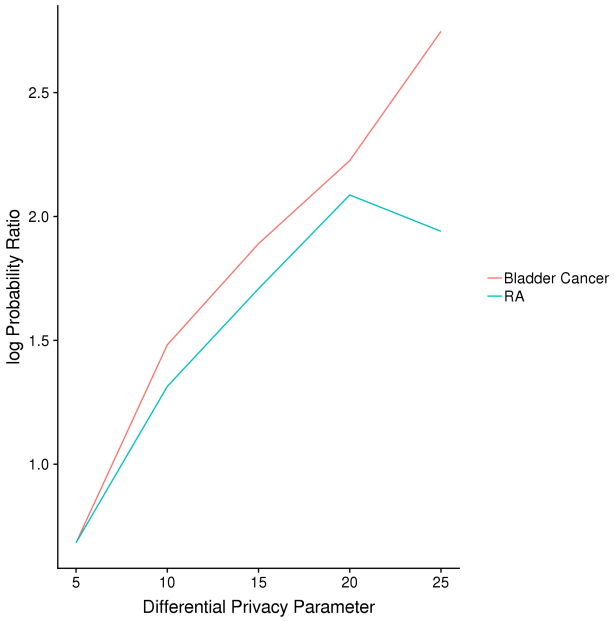


Fig. 4. Comparison to differential privacy. We compare ϵ to the log probability ratio (see methods) when adding unbounded noise for both bladder cancer (BC) and rheumatoid arthritis (RA) cohorts. We see that the log ratio for our measure is much smaller than that predicted by differentially private bounds for the Laplacian mechanism, differing by roughly a factor of 10. This shows that, under reasonable assumptions, the Laplacian mechanism greatly overestimates the level of noise required to achieve a given level of privacy. The risk is calculated on a rheumatoid arthritis dataset of 50 cases, 950 controls, with 10 to 50 SNPs released, and noise $\lambda = .01$.

3.3. *Comparison to Differential Privacy*

One of the main candidates that has been suggested for privacy preserving statistical calculations is known as differential privacy. The informal idea behind differential privacy is that, by adding noise to a released statistics, one is able to achieve a level of plausible deniability that a particular individual was in your dataset. This level of deniability is measured by a privacy parameter, ϵ. The larger the ϵ parameter, the less plausible deniability is preserved.

Under reasonable assumptions ϵ can be seen as being an upper bound on the ratio of the probability of any individual being in the dataset before and after releasing the perturbed statistic of interest (see Methods). The smaller this log ratio, the less information that is being leaked. As such, we wanted to compare this upper bound with the log probability ratio produced by our probability model (Fig 4). Note that we apply the unbounded version of our method, since standard differential privacy techniques do not allow for bounded noise. To this end, we apply a standard differentially private mechanism, known as the Laplacian mechanism. We see that the log ratio for our measure is much smaller than that predicted by the differentially private bounds from the Laplacian mechanism, differing by roughly a factor of 10— far outside the normal range for differential privacy. Importantly, these results shows that our approach allows for the release of much more data, at the cost of a slightly weaker privacy guarantee.

3.4. *Accuracy of MCMC*

Our method aims to estimate the true disclosure risk using a sampling based technique. Such sampling techniques introduce uncertainty in the estimated disclosure risk. In order to quantify this, we generated a dataset were we could directly calculate the probability of any particular individual being in the output (see Methods). We then compared the true probability versus the estimated probability using our MCMC based approach. We see that, for 98.5% of the simulated individuals the error is less than .05, with only one individual having an error of greater than .1. Moreover, the estimates can be improved by increasing the number of samples taken, as well as the number of MCMC iterations per sample.

3.5. *Beyond MAF: applications to log odds ratios*

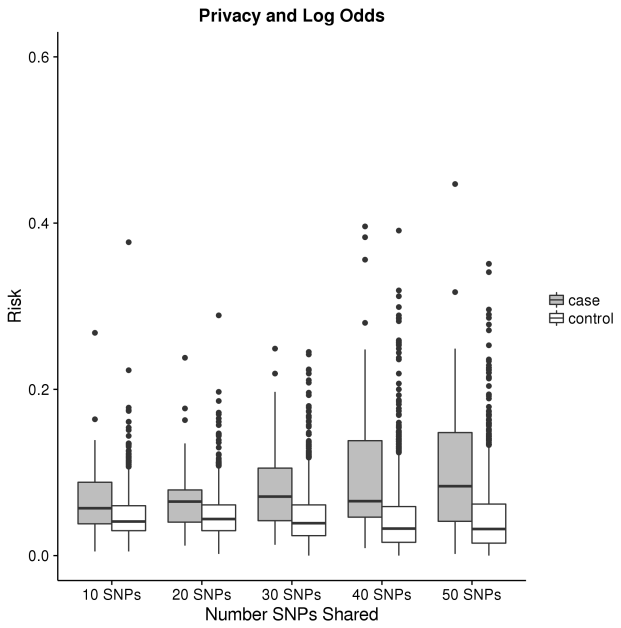


Fig. 5. Privacy and the log odds ratio. We compare the disclosure risk versus the number of SNPs whose log odds ratio data we release, with unbounded noise. The results are qualitatively very close to those we see for the MAF. More importantly, this shows that our technique can be extended to new statistics, and is not just limited to one. The risk is calculated on a rheumatoid arthritis dataset of 50 cases, 950 controls, varying the number of SNPs released from 10 and 50 SNPs, with $\lambda = .1$.

So far we have focused on using our approach to measure the privacy loss when releasing MAF for a large number of SNPs. As mentioned, however, the approach introduced here can be applied to almost any real valued statistic (or collection of statistics). To see this, we apply our method to measure the amount of privacy lost when releasing information about the odds ratio. More specifically, we release the log odds ratio for numerous SNPs.

We calculate these log odds ratios on 50 cases and 950 controls from the rheumatoid arthritis GWAS (Figure 5). We add (unbounded) Laplacian noise with parameter $\lambda = .1$ (this corresponds to an average additive error of .1 in the returned log odds ratio), and measured

the privacy for various number of SNPs. We see that, in this experiment, the disclosure risk is fairly small for most individuals, with a few outliers who have greater risk. In particular, the results are comparable to what we see when releasing noisy MAF. This suggests that releasing noisy log odds ratios has a minimal effect on privacy when $\lambda = .1$.

4. Conclusion

We have introduced a novel method for measuring privacy loss in aggregate genomic data. Our method manages to avoid making the strong assumptions about the background population required by many other methods (assumptions that might not hold in practice), while still achieving better accuracy than standard differentially private methods.

The framework we introduced here can be extended in many ways, enabling more expressive analysis. For example, the current method requires adding noise to the output statistic. If this noise is small enough, the effect on accuracy is minimal, and is similar to the effect of only releasing a small number of significant digits (a common practice in most analysis). Even still, many practitioners are uncomfortable with the idea of adding noise to their data, so extending the method to unperturbed data would be of great use. For example, ideas from approximate Bayesian calculations might be used to help achieve this.[24]

Another important direction for future work is to improve runtime. This direction is of particular importance since most datasets without ascertainment bias (the type of datasets our method is meant to be applied to) are quite large. To address this, we are currently exploring approximate methods, such as variational techniques, to allow for greater scalability.[25]

Our method provides another tool to help understand the privacy risks inherent in sharing genomic data. Many of the arguments between those who want to publicly share genomic data (and health data more broadly) and those who want to keep it under lock and key revolve around the fact that we are still not certain about the real world risks posed by public disclosure of genomic data. As such, continuing to investigate the benefits and risks of sharing this data is paramount in order to be able to improve human health without negatively effecting study participants.

Acknowledgements

We want to acknowledge Jadwiga Bienkowska for introducing us to the data set we used, as well as Noah Daniels, Jian Peng, Hoon Cho, and other members of the Berger and Sahinalp labs for useful discussions.

B.B. and C.S. are partially supported by the US National Institutes of Health grant GM108348. C.S. and S.S. were partially funded by NSERC Discovery Frontiers Program, "The Cancer Genome Collaboratory". C.S. is partially funded by Indiana University Precision Health Initiative.

References

1. Y. Erlich and A. Narayanan, Routes for breaching and protecting genetic privacy, *Nature Reviews Genetics* **15**, 409 (2014).

2. N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. Pearson, D. Stephan, S. Nelson and D. Craig., Resolving individual's contributing trace amounts of DNA to highly complex mixtures using high-density snp genotyping microarrays, *PLoS Genet* **4** (2008).
3. X. Zhou, B. Peng, Y. Li, Y. Chen, H. Tang and X. Wang, To release or not to release: evaluating information leaks in aggregate human-genome data, in *ESORICS*, 2011.
4. E. Schadt, S. Woo and K. Hao, Bayesian method to predict individual snp genotypes from gene expression data, *Nat Genet* **44**, 603 (2012).
5. H. Im, E. Gamazon, D. Nicolae and N. Cox, On sharing quantitative trait GWAS results in an era of multiple-omics data and the limits of genomic privacy, *Am J Hum Genet* **90**, 591 (2012).
6. X. Jiang, Y. Zhao, X. Wang, B. Malin, S. Wang, L. Ohno-Machado and H. Tang, A community assessment of privacy preserving techniques for human genomes, *BMC Medical Informatics and Decision Making* **14** (2014).
7. S. Sankararaman, G. Obozinski, M. Jordan and E. Halperin, Genomic privacy and the limits of individual detection in a pool, *Nat Genet* **41**, 965 (2009).
8. S. Simmons and B. Berger, One size doesn't fit all: Measuring individual privacy in aggregate genomic data, in *GenoPri*, 2015.
9. C. Uhler, S. Fienberg and A. Slavkovic, Privacy-preserving data sharing for genome-wide association studies, *Journal of Privacy and Confidentiality* **5**, 137 (2013).
10. F. Tramer, Z. Huang, J. Hubaux and E. Ayday, Differential privacy with bounded priors: Reconciling utility and privacy in genome-wide association studies, in *CCS*, 2015.
11. S. Wang, N. Mohammed and R. Chen, Differentially private genome data dissemination through top-down specialization, *BMC Medical Informatics and Decision Making* **14** (2014).
12. F. Yu and Z. Ji, Scalable privacy-preserving data sharing methodology for genome-wide association studies: an application to idash healthcare privacy protection challenge, *BMC Medical Informatics and Decision Making* **14** (2014).
13. Y. Zhao *et al.*, Choosing blindly but wisely: differentially private solicitation of DNA datasets for disease marker discovery, *JAMIA* **22**, 100 (2015).
14. S. Simmons and B. Berger, Realizing privacy preserving genome-wide association studies, *Bioinformatics* **32**, 1293 (2015).
15. S. Simmons, C. Sahinalp and B. Berger, Enabling privacy-preserving gwass in heterogeneous human populations, *Cell Systems* **3**, 54 (2016).
16. J. Forster, Bayesian methods for disclosure risk assessment, in *Monographs of Official Statistics*, 2006.
17. C. Dwork and R. Pottenger, Towards practicing privacy, *J Am Med Inform Assoc* **20**, 102 (2013).
18. L. Sweeney, K-anonymity: a model for protecting privacy, *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems* **10**, 557 (2011).
19. K. E. Emam, E. Jonker, L. Arbuckle and B. Malin, A systematic review of re-identification attacks on health data, *PLoS ONE* **6** (2011).
20. N. Li, W. Qardaji, D. Su, Y. Wu and W. Yang, Membership privacy: a unifying framework for privacy definitions, in *SIGSAC*, 2013.
21. W. Hastings, Monte carlo sampling methods using markov chains and their applications, *Biometrika* **57**, 97 (1970).
22. R. Plenge *et al.*, Traf1-c5 as a risk locus for rheumatoid arthritis– a genomewide study, *New England Journal of Medicine* , 1199 (2007).
23. J. Figueroa *et al.*, Genome-wide association study identifies multiple loci associated with bladder cancer risk, *Hum Mol Genet* **23**, 1387 (2014).
24. M. Sunnaker, A. Busetto, E. Numminen, J. Corander, M. Foll and C. Dessimoz, Approximate bayesian computation, *Plos Computational Biology* **9**, p. e1002803 (2013).
25. C. Bishop, *Pattern Recognition and Machine Learning* (Springer, 2006).

Evaluation of patient re-identification using laboratory test orders and mitigation via latent space variables

Kipp W. Johnson[1*], Jessica K. De Freitas[1*], Benjamin S. Glicksberg[2], Jason R. Bobe[1], Joel T. Dudley[1#]

[1]*Institute for Next Generation Healthcare*
Department of Genetics and Genomics Sciences,
Icahn School of Medicine at Mount Sinai,
770 Lexington Ave 15th Fl.
New York, NY 10065, USA

[2]*Bakar Computational Health Sciences Institute*
The University of California San Francisco
San Francisco, CA 10065, USA

**Authors contributed equally*
[#]*Corresponding author: joel.dudley@mssm.edu*

Anonymized electronic health records (EHR) are often used for biomedical research. One persistent concern with this type of research is the risk for re-identification of patients from their purportedly anonymized data. Here, we use the EHR of 731,850 de-identified patients to demonstrate that the average patient is unique from all others 98.4% of the time simply by examining what laboratory tests have been ordered for them. By the time a patient has visited the hospital on two separate days, they are unique in 72.3% of cases. We further present a computational study to identify how accurately the records from a single day of care can be used to re-identify patients from a set of 99 other patients. We show that, given a single visit's laboratory orders (even without result values) for a patient, we can re-identify the patient at least 25% of the time. Furthermore, we can place this patient among the top 10 most similar patients 47% of the time. Finally, we present a proof-of-concept technique using a variational autoencoder to encode laboratory results into a lower-dimensional latent space. We demonstrate that releasing latent-space encoded laboratory orders significantly improves privacy compared to releasing raw laboratory orders (<5% re-identification), while preserving information contained within the laboratory orders (AUC of >0.9 for recreating encoded values). Our findings have potential consequences for the public release of anonymized laboratory tests to the biomedical research community. We note that our findings do not imply that laboratory tests alone are personally identifiable. In the attack scenario presented here, reidentification would require a threat actor to possess an external source of laboratory values which are linked to personal identifiers at the start.

Keywords: Electronic health records, anonymization, patient re-identification, data privacy, variational autoencoder

1. Introduction

Electronic Health Records (EHRs) have been widely adopted as a component of the modern American healthcare system (1). EHRs contain information such as disease-related diagnosis billing codes, lab test orders and results, procedures performed, and medications prescribed. Although EHRs are primarily designed for the purpose of encounter documentation and billing, the data can also be repurposed for efforts to improved clinical care (2, 3) or for biomedical investigation (4–6).

For use in research, EHRs are often de-identified in accordance with the Health Insurance Portability and Accountability Act (HIPAA) (7). HIPAA's Privacy Rule mandates protection for identifiable variables such as name, zip code, date of birth, etc. Because of this, public release of EHR data requires either (1) expert "determination" or (2) "safe harbor" privacy practices. Expert determination involves an individual with appropriate knowledge and experience determining that data poses minimal risk. "Safe harbor" practice is the removal of 18 pieces of information from the EHR, with the 18th being a "catch-all" category for "any other unique identifying characteristic." However, the definition for what constitutes individually identifiable information has been challenged by a variety of re-identification attacks and privacy breaches (8, 9). In practice, the privacy rule does not constrain the types of uses of health data once it has been de-identified by these methods, although covered entities sometimes take additional precautions such as data use agreements that forbid intentional re-identification.

Re-identification is the process of matching anonymized personal data with its owner via linkage with an external resource. Information such as a person's name and address are obviously identifying, but in some circumstances data such as disease diagnoses or lab tests may be identifiable. In fact, there have been several important examples of this type of privacy attack. Loukides et al. demonstrated that existing privacy protection methods were not sufficient to protect against re-identification by identifying a subset of 2800 patients from using EHR diagnosis codes alone (10). Although the diagnosis code dataset from EHRs were anonymized, the risk for re-identification came from cross-referencing with a secondary data source that contained the patient's exact diagnosis codes. Other researchers have developed strategies to anonymize combinations of disease billing codes with linked demographics (11).

In this manuscript, we first demonstrate the uniqueness of the pattern of physician-ordered laboratory tests for specific individuals. After finding that these laboratory orders are highly specific, we propose an algorithm and evaluation framework to re-identify patients using only a single day of laboratory orders. Following this, we explore if latent variables can be constructed using a variational autoencoder which simultaneously preserve information contained within the laboratory orders and also increase patient privacy.

2. Methods

We present an overall workflow of the study in **Figure 1**. While we do use EHRs of real patients in this paper, our dataset is anonymized (i.e., de-identified) and does not include any explicit identifiers for patients such as name, social security number, hospital medical record number, or specific dates of encounters. Our dataset uses pseudo-identifiers for each patient that are internally consistent but do not map to outside datasets. All re-identification methods and results presented do not attempt to match pseudo-identifier to real identities, as that would violate ethical research practice, patient privacy, the Health Insurance Portability and Accountability Act of 1996 (HIPAA), and institutional policies of the Mount Sinai Hospital and Icahn School of Medicine at Mount Sinai.

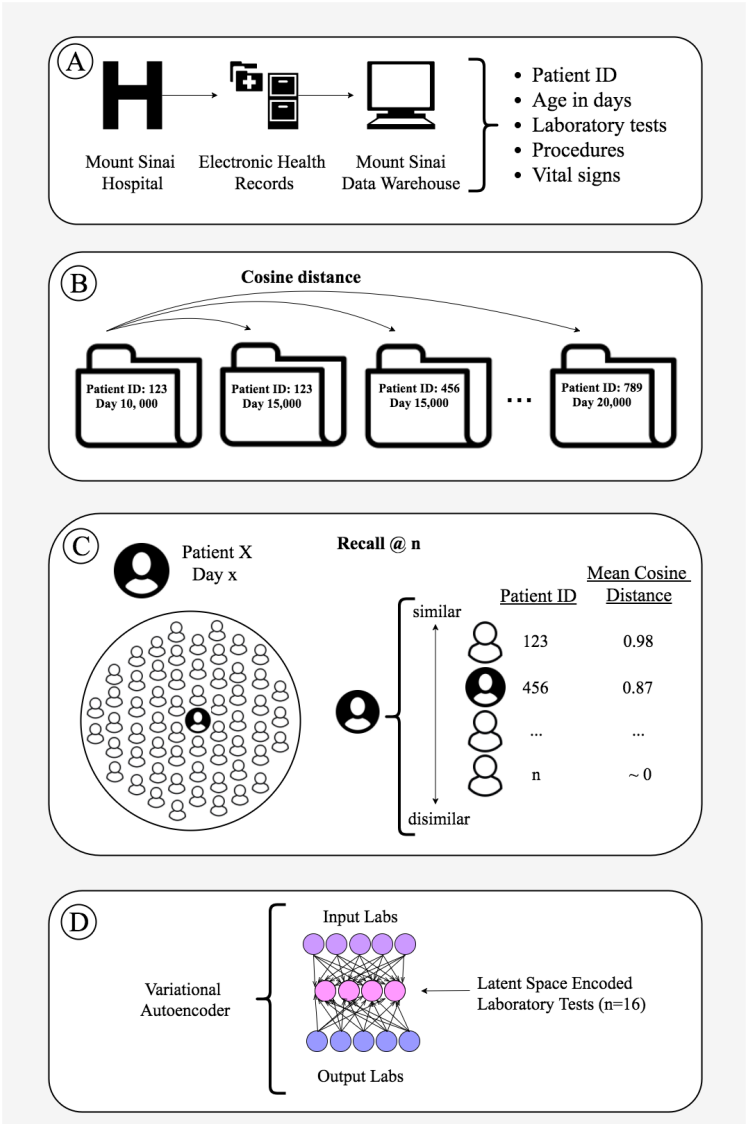


Figure 1. Overall workflow of our study. A) Data for this study was obtained from the Mount Sinai Hospital Data Warehouse B) Cosine distances between each patient-day event were calculated C) Evaluation by Recall @ n from n = 1 to n = 100 D) Use of a variational autoencoder to anonymize laboratory orders.

2.1. *Data preparation of research cohort and laboratory tests*

We used the EHRs of patient visits from the Mount Sinai Hospital (MSH), a tertiary-care urban hospital located on the Upper East Side of Manhattan in New York City. For this study, we obtained the records of all individuals between 18-90 years old. Since we sought to obtain generalized re-identifiability statistics, we did not select for patients based upon any particular criteria.

We queried the MSH EHRs for all possible laboratory tests ordered and their values. We removed laboratory tests that did not have numeric value results, could not be made to give binary data (e.g. positive/negative result), could not be used to give ordinal results (e.g. low/medium/high), had results which were clearly erroneous and nonsensical results (e.g., some labs had values which were long text strings

describing laboratory tests in place of results), or had other missing information such as order date.

2.2. *Assessment of how characteristic patient laboratory tests are for individual patients*

We first characterized how unique each patient's laboratory tests were. To do this, we concatenated all the laboratory tests which had been ordered at any time point for each patients into a single standardized text string. We then computed the MD5 hash of this standardized text string so that each unique combination of laboratory tests could be represented as a unique 128-bit checksum. Ultimately, patients who have received the same permutations of laboratory tests will have exactly the same MD5 hash. Finally, we checked for overlap among the MD5 hashes in order to determine the uniqueness of laboratory test orders.

2.3. *Assessing if using one day of patient records is sufficient to re-identify patients*

We next sought to determine if a single day of patient records would be sufficient to re-identify a patient compared to a random sample of other patients. For computational tractability we included in this analysis only those laboratory tests which had been ordered at least 500 times.

2.3.1. *Creation of patient-day-laboratory vectors*

Each individual patient's laboratory records were collapsed to the day in which they were ordered. If the same laboratory test was ordered more than once on the same day, we took only one occurrence of that test. We thus assembled each patient-day as a vector θ of length *l*, where *l* is the count of all laboratory tests obtained from the EHRs. Laboratory tests for a given patient-day were considered to be a binary variable where 0 denotes absence (laboratory test not ordered for this patient on this day) and 1 denotes presence (laboratory test ordered for this patient on this day)

2.4. *Vector distance metrics*

After computing the binary lab vector for each patient-day, we then determined pairwise similarities between patient-day vectors by computing their cosine distance. The cosine distance is a straightforward measure of similarity between vectors computed by taking the dot product of two vectors divided by the product of the two vectors' magnitude (Eq. 1).

$$\text{cosine distance} = 1 - \frac{\theta_1 \theta_2}{\|\theta_1\| \|\theta_2\|} \tag{1}$$

We thus assembled a symmetric MxM pairwise cosine distance matrix where M is the total number of patient-days. Each *(i,j)* entry in the distance matrix corresponds to the cosine distance between laboratory tests on patient-day *i* and patient-day *j*. We selected cosine distance as the similarity metric because it is a vector space metric commonly used information retrieval settings. The cosine distance in the special case of non-negative binary data (e.g. 0, 1) is also known as the Ochai distance and has a range of [0, 1] where 0 is perfect dissimilarity and 1 is perfect similarity.

2.5. *Patient-day re-identification algorithm*

Because the running time of pairwise distance computation grows according to the square of the patient-day counts (i.e. $O(n^2)$, quadratic complexity), it was not computationally feasible to compute the pairwise distances between all patient-day vectors. We thus posed our re-identification task as an attempt to see if, given a single-day of patient records, we could re-identify the patient based upon his or her other day's records compared to the records of 99 other randomly selected individuals.

Specifically, we randomly selected a single patient-day vector to act as the seed "breached record" for query. Our dataset for re-identification comprised of that breached patient's other patient-day vectors, not including the breached record, and all of the patient-day vectors of another 99 randomly selected patients. We then computed the cosine distance of this query vector from all other patient-day vectors in our sample. Then for each patient, we calculated the mean of the cosine distances of all their vectors from the query vector. Thus, in the end, given one patient-day record we had 100 distances corresponding to the mean distance of 100 other patients from this one patient-day. We computed this for all patient-days in the dataset. We then repeated the entire above algorithm 100 times.

For each iteration of the previous algorithm, we ultimately obtained 100 scores for distance between our query patient-record and 99 randomly selected individuals, plus the other records belonging to the initial patient from whom we extracted the seed "breached" record.

2.6. *Patient-day re-identification evaluation framework*

We evaluated our performance using a modified version of the "*Recall @ n*" metric commonly used in information retrieval. Since there is only one correct patient match to our query "breached" record, we evaluated if this correct patient match was within the scores corresponding to the n closest patients. The score per patient record and n was computed as a binary variable (e.g. patient is within n closest records = 1 or patient not within n closest records = 0). *Recall @ n=1* implies that the correct match was the closest score to our patient. *Recall @ n=100* will always be 100% since that implies that the patient is within the closest 100 patients queried, which will always be the case since we are querying a sample of 100 patients.

The expected *recall @ n is n/100* for a completely random classifier. Thus, we can assess our re-identification algorithm as the improvement over random classification (the null hypothesis). This formulation analogous to the area under the receiver-operating characteristic curve ("AUROC") commonly used for assessing supervised machine learning classification performance.

2.7. *Generalization of patient laboratory test data using a variational autoencoder*

Finally, we sought to determine whether we could encode laboratory tests orders into a reduced-dimensional latent space which was still useful but could reduce re-identifiability. To do this, we

employed a variational autoencoder implemented in Keras (https://github.com/keras-team). Variational autoencoders feature two major architectural components: First, an encoding model which takes a sequence of inputs (in our case, binary presence or absence of lab tests) and encodes them into a latent hidden representation space. A generative decoder then decodes the latent space representation back into a probability distribution representing the input data. We employed a standard VAE loss function which is the sum of the binary cross entropy between the input lab test vectors and output lab test vectors plus the Kullback-Liebler divergence between the learned encoding probability distribution and a unit Gaussian (Eq. 2).

$$l = -\sum_{x} p\left(x_{\text{out}}\right) \log q\left(x_{\text{in}}\right) + KL\left(Z(\text{latent}), N(0,1)\right) \tag{2}$$

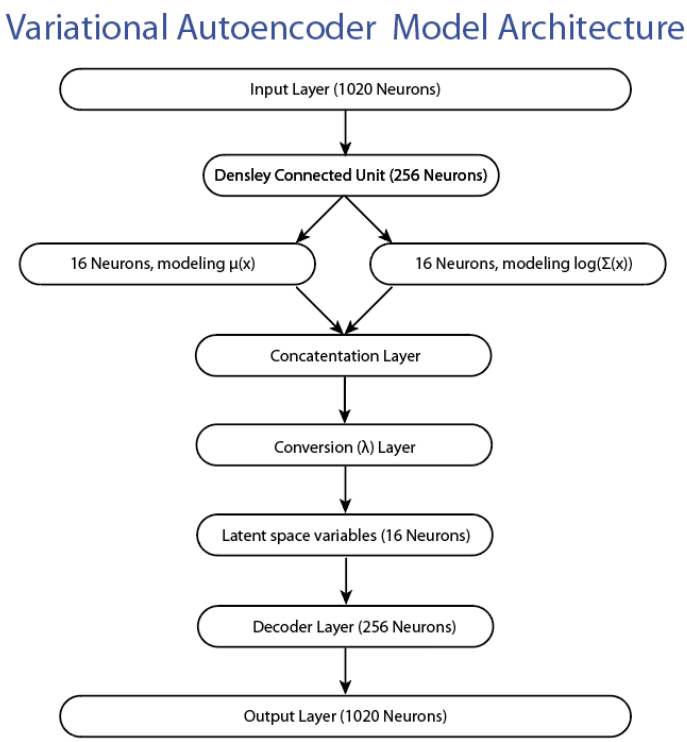


Figure 2: Architecture of variational autoencoder

Our hypothesis is that releasing the latent-state variables instead of the raw laboratory tests would still be useful to researchers, but would importantly reduce the potential for direct re-identifiability. After obtaining latent-space variable from this method, we employed the same technique as in the previous section (cosine distance between person-day latent variable vectors) in order to see if we could re-identify patients given one single day of data (e.g., the latent variables corresponding to that day). For ease of computation, we desired a minimal number of neurons in the latent space which could accurately recapitulate the input vectors. We found that 2^1, 2^2, and 2^3 latent neurons were insufficiently accurate, but 2^4 (16) latent space neurons produced experimentally acceptable results. The architecture of the model is given **Figure 2.**

3. Results

3.1. *Electronic health record data*

From the selected cohort of 731,850 individuals, we obtained laboratory records from those with at least one recorded laboratory test. These individuals had 342,485,583 laboratory test values for 2,635 different possible laboratory procedures. These distinct labs had been ordered on average 468.0 times per patient (standard deviation: 1,415) with a minimum frequency of one and a maximum frequency of 94,749 different laboratory results. This range of results is likely due to the fact that Mount Sinai Hospital sees a unique mix of patients, from everyday office visits to patients who may remain in the intensive care unit for weeks. The average patient had records for 49.8 different kinds of labs (standard deviation of 40.4) with a minimum of one kind of laboratory test and maximum of 442 types of different laboratory tests. The laboratory tests in total represented a period totaling 17,657 years (6,449,310 patient-days). Patients had a mean of 8.81 different days (standard deviation: 20.6) with at least one laboratory test result. The total range of days per patient was from one day to 2.47 years. 81.6% of patients had 10 or fewer days of laboratory results and 90.7% had fewer than 20 days of laboratory values. There were 218 different laboratory tests ordered only once (8.3% of all tests) and 1,186 laboratory values were ordered less than 1000 times (45.0% of all tests).

3.2. *What percentage of patients can be uniquely identified by laboratory tests ordered for them?*

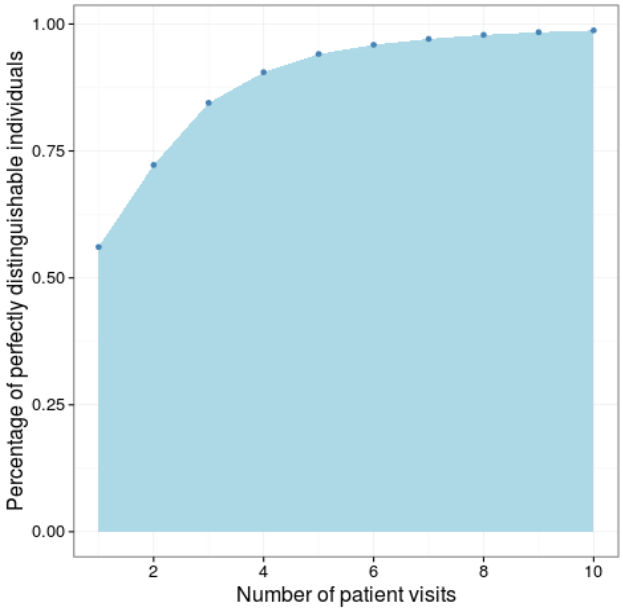


Figure 3: Percentage of patients whose particular pattern of visits are completely unique to them, by number of visits to hospital.

We analyzed the uniqueness of the laboratory tests ordered per patient, e.g. what percentage of patients had perfectly unique laboratory tests different from all other patients (**Figure 3**). This corresponds to the ability to perfectly recognize a patient by simply knowing what laboratory test have been ordered. We did not consider the numerical results for the lab, but merely assessed whether the tests had been ordered for a given patient or not. In total, 56.1% of patients could be perfectly characterized by their laboratory results (e.g., their particular combination of laboratory tests was completely different from all other patients in the EHR). However, the distinguishability increased very rapidly with increasing count of encounters in the EHR. Patients who had at least two days of laboratory values were different from all

other patients 72.3% of the time. Patients who had at least nine days of laboratory values (the mean number of days per patient) were different from all other patients 98.4% of the time.

3.3. *Can we re-identify patients using only 1 day of laboratory tests?*

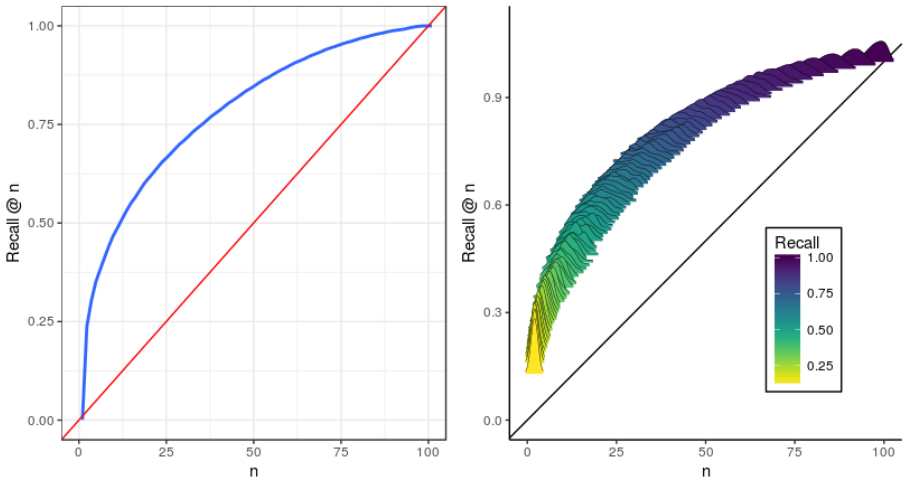


Figure 4: Reidentification performance using only one day of lab values. Panel on right shows distribution from 250 simulations.

We next formulated a theoretical privacy "attack": Given only a single day of records for a patient, could we re-identify this individual from a set of 99 other individuals? We show the performance for this re-identification task in **Figure 4**. Here, the red line represents the probability for random re-identification and the blue line represents the added ability to distinguish above random. We ranked the query individual as the most similar individual 25% of the time. We could place the query individual among the top 10 individuals 47% of the time.

3.3.1. *Assessing the performance of latent variables from variational autoencoder to predict laboratory orders*

After applying a variational autoencoder to encode input EHR variables, we first assessed whether our encoded latent variables indeed adequately model the dataset. This is important, because we do want to ensure they retain adequate information in the data. We then attempted to predict whether a given test would be ordered for a patient or not on a given day. It is important to show that the latent variables are actually associated with laboratory results before we can demonstrate that they may be useful for anonymity.

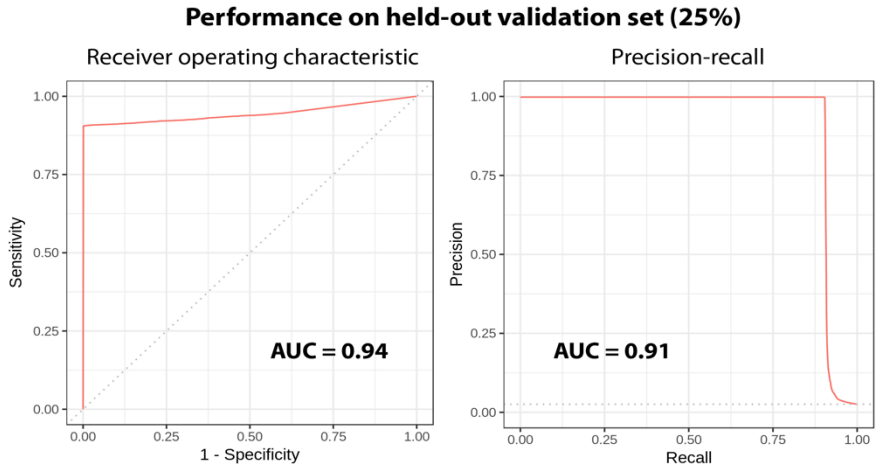


Figure 5: Ability of learned encodings to recapitulate input laboratory order vectors on a held-out validation set. ROC curve on left and Precision-recall curve on right.

We found that our latent variables were highly predictive of the lab order status, achieving an area under the receiver-operating characteristic curve of 0.94 and area under the precision-recall curve of 0.91, as demonstrated in **Figure 5**. This means that they recapitulate the underlying laboratory tests well.

3.3.2. *Comparing raw lab tests to latent-space abstracted laboratory tests for privacy preservation*

We assessed the ability to re-identify patients based upon cosine distance of latent variables. This is the same algorithm as used previously to re-identify patients, but with our encoded variables representing patient labs instead of using the patient labs themselves. We found that in every case, using encoded latent variables gave greater privacy protection compared to the raw lab values used for the same samples (**Figure 6**).

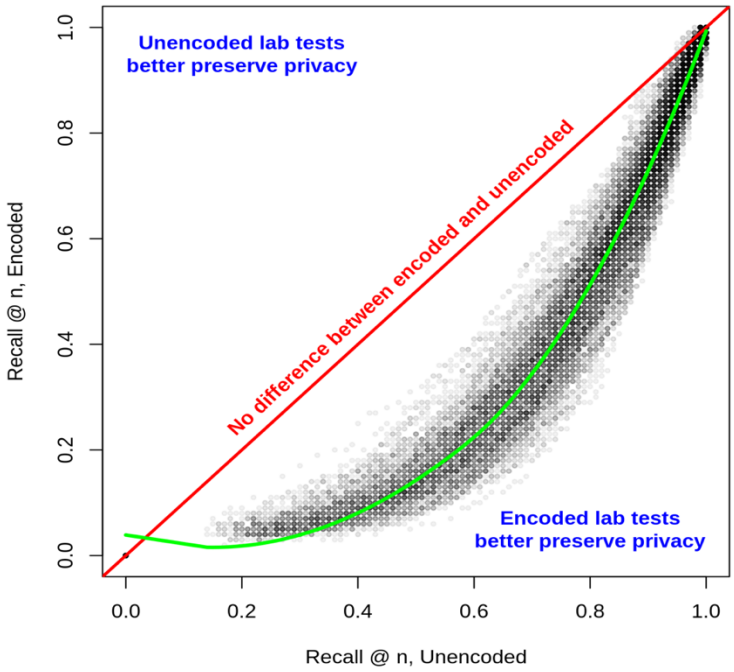


Figure 6: The learned latent space neurons were significantly better at reducing recall @ n for all a values of n. The red line represents the situation where the re-identification recall @ n score is the same between the unencoded lab tests and the latent space lab tests.

For *recall @ 1* through *recall @ 99* (since 100 samples were used per assessment), latent space performed significantly better at all points ($p<10^{-16}$, matched pairs t-test). We were only able to perfectly match the individual from one day of laboratory orders 5% of the time, compared to 25% of the time using the raw laboratory orders as shown previously.

4. Discussion

Using the Mount Sinai Hospital (MSH) patients' laboratory test history and a straightforward similarity measure, we discovered that by the time the patient has had contact with the hospital on nine separate days, their laboratory test orders are completely unique to that patient 98.4% of the time. This is a significant finding, since it implies that public datasets which contain all of the laboratory tests ordered for a specific person may be able to be matched against a known set of electronic health records (EHR) with perfect fidelity in some cases. We also show that we can obtain reasonable re-identification performance using a single day of laboratory values. Finally, we demonstrate that latent encoded variables make the problem of re-identification significantly more difficult without knowing the exact model used to encode the latent variables.

One of our primary motivations for this study stems from the idea that lab tests as are commonly used as covariates in statistical models to help to produce more accurate probability estimates for outcomes. For example, if a researcher intended to study the effect of statin therapy on incident heart disease, he or she would need to adjust for a levels of baseline LDL cholesterol and other lab tests. Instead of using actual lab tests, lower-dimensional encoded variables which contain the same amount of information as the lab tests would serve just as well as control variables. This is exactly analogous to the use of genetic SNP principal components to represent genetic ancestry in genome-wide association studies. One of the major values of our study is that we demonstrate that lower-dimensional representations of the EHR contain similar amounts of information as unprocessed records, while simultaneously preserving privacy.

Our study had several limitations. First, the work was performed with data from only one healthcare institution. However, MSH is a large tertiary care hospital with a significant diversity of patients. We also focused exclusively on laboratory test orders and did not include data such as disease diagnoses, ethnicity, gender, etc. which are often included in EHR. In our re-identification analysis, we attempted to identify an individual against a subset of 99 other random individuals, not the entire cohort of patients. Although this is a realistic scenario when performing biomedical research on a specific patient population, further analysis is needed to understand if our methods hold true when identifying an individual from the entire database. Finally, we assessed here only the binary presence or absence of laboratory test orders. It is quite possible that considering the numeric results of laboratory tests could increase re-identifiability substantially. For example, hypothetical patients with LDL cholesterol test results of 60mg/dL vs. 600mg/dL would be easily separable, although our current method considers only the fact that LDL tests were ordered for both patients. However, as we have demonstrated, considering only the binary absence or presence of orders already works reasonably well and we believe our performance metrics are conservative.

Potentially, replacing our variational autoencoder with other kinds of autoencoders or other dimensionality-reduction methods would also have been effective. Autoencoders essentially work by learning compression and decompression functions which minimize a loss function, whereas variational autoencoders learn a probability distribution which minimizes a loss function. Experimentally, our use of a VAE worked well enough to learn latent variables. By introducing this as a proof-of-concept, we felt that it would not be too valuable to benchmark against other alternatives. Future experimental and theoretical work could explore the dimensionality reduction methods used in this paper more thoroughly. Finally, we cannot release the training dataset since it contains the real patient records of hundreds of thousands of patients and could potentially enable future reidentification attacks.

We must also note here that our findings do not imply a threat model whereby patients may be identified from laboratory tests themselves, without a threat actor having an outside source of information. We show here only that lab tests are highly distinctive. For re-identification, the techniques presented here would require the threat actor to have at least some amount of information from another data source containing laboratory tests which were matched to actual patient identifiers. Furthermore, our re-identification technique only attempted to re-identify from one out of 100 instead of one out of the entire dataset, since our method for computing pairwise vector distances would not scale computationally to that extent.

Taken altogether, we believe that our findings have significant implications for the release of anonymized laboratory test results to the broad biomedical research community. Researchers should consider the possible consequences of making extensive laboratory order data for patients freely available, and should inform patients that this level of detail may potentially make them open to re-identification.

If researchers choose to release data, we suggest they consider providing latent-variable encoded laboratory values instead if this data would remain useful in their particular scientific context. Potentially, the methods we demonstrate here for laboratory test orders could be applied to other forms of data contained within the EHR.

Scientists have an obligation to respect their subjects' generosity in donation of data by maintaining their privacy and here we have demonstrated one method to make re-identification more challenging.

References

1. Adler-Milstein J, Jha AK. HITECH Act Drove Large Gains In Hospital Electronic Health Record Adoption. *Health Aff (Millwood)* 2017;36:1416–1422.

2. Johnson KW, Torres Soto J, Glicksberg BS, et al. *Artificial Intelligence in Cardiology. J. Am. Coll. Cardiol*. 2018;71:2668–2679.

3. Johnson KW, Shameer K, Glicksberg BS, et al. Enabling Precision Cardiology Through Multiscale Biology and Systems Medicine. *JACC Basic Transl Sci* 2017;2:311–327.

4. Glicksberg BS, Johnson KW, Dudley JT. The next generation of precision medicine: observational studies, electronic health records, biobanks and continuous monitoring. *Hum. Mol. Genet*. 2018;27:R56–R62.

5. Glicksberg BS, Miotto R, Johnson KW, et al. Automated disease cohort selection using word embeddings from Electronic Health Records. *Pac Symp Biocomput* 2018;23:145–156.

6. Miotto R, Li L, Kidd BA, Dudley JT. Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. *Sci Rep* 2016;6:26094.

7. Office for Civil Rights, Department of Health and Human Services. Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule and the National Instant Criminal Background Check System (NICS). *Final rule. Fed Regist* 2016;81:382–396.

8. Meeks DW, Smith MW, Taylor L, Sittig DF, Scott JM, Singh H. An analysis of electronic health record-related patient safety concerns. *J Am Med Inform Assoc* 2014;21:1053–1059.

9. Menon S, Singh H, Giardina TD, et al. Safety huddles to proactively identify and address electronic health record safety. *J Am Med Inform Assoc* 2017;24:261–267.

10. Loukides G, Denny JC, Malin B. The disclosure of diagnosis codes can breach research participants' privacy. *J Am Med Inform Assoc* 2010;17:322–327.

11. Poulis G, Loukides G, Skiadopoulos S, Gkoulalas-Divanis A. Anonymizing datasets with demographics and diagnosis codes in the presence of utility constraints. *J Biomed Inform* 2017;65:76–96.

Implementing a universal informed consent process for the *All of Us* Research Program

Megan Doerr[1], Shira Grayson[1], Sarah Moore[1], Christine Suver[1], John Wilbanks[1], Jennifer Wagner[2*]
[1]*Sage Bionetworks, 2901 Third Avenue*
Seattle, WA 98121, USA
Email: megan.doerr@sagebionetworks.org

[2]*Center for Translational Bioethics & Health Care Policy, Geisinger, 100 N. Academy Ave., MC 30-42, Danville, PA 17822, USA*

The United States' *All of Us* Research Program is a longitudinal research initiative with ambitious national recruitment goals, including of populations traditionally underrepresented in biomedical research, many of whom have high geographic mobility. The program has a distributed infrastructure, with key programmatic resources spread across the US. Given its planned duration and geographic reach both in terms of recruitment and programmatic resources, a diversity of state and territory laws might apply to the program over time as well as to the determination of participants' rights. Here we present a listing and discussion of state and territory guidance and regulation of specific relevance to the program, and our approach to their incorporation within the program's informed consent processes.

Keywords: Informed consent, conflicts of law, choice of law, ELSI, bioethics

1. Background

1.1 *The* All of Us *Research Program*

The *All of Us* Research Program (AoURP) is a longitudinal national cohort program funded by the United States (US) National Institutes of Health (NIH) with investigators, study infrastructure, data management systems, and governance schema distributed across the US. All participating institutions signed Reliance Agreements ceding authority to the *All of Us* Institutional Review Board (AoU IRB) for ethical and regulatory oversight.

AoURP aims to enroll one million or more persons living within the US to contribute personal health information, including protected health information and biospecimens, to a central resource designed to accelerate research and improve health. Recruitment goals were established based on US 2040 census projections with purposeful oversampling of populations traditionally underrepresented in biomedical research to ensure sufficient statistical power for subpopulation analysis. The program intends to follow participants for at least 10 years.

Germane to AoURP is the well-documented geographic mobility of the US population, with the percentage of those living in the US who report having moved in the past 5 years at least 2 times greater than most African, Asian, Central and South American, and European nations [1]. Within the US, people who do not self-identify as white and those of lower annual income demonstrate higher geographic mobility, on average, compared to people who self-identify as

white and those of greater annual income [2]. Many populations who have traditionally been underrepresented in biomedical research, such as people who are migrant workers, homeless, or identify as gender and sexual minorities, demonstrate exceptionally high rates of geographic mobility.

1.2 *Overview of the AoURP informed consent process*

All persons wishing to participate in AoURP must complete an informed consent process that unambiguously indicates their consent to join. Given its ambitious recruitment goals, the program decided that the primary modality for the consent process would be electronic (i.e., web- or app-mediated) to allow for broad deployment and rapid scaling. Further, it was the program's desire that the consent process be consistent for all persons regardless of geographic location, enrollment method, or affiliation (participants can enroll directly or through an affiliated healthcare provider organization). Finally, due to the longitudinal and evolving nature of the study and, further, to provide a flexible participant experience, the informed consent for AoURP is modular (Table 1). Following an initial consent experience (Primary Consent), additional "modules" for program activities not included in the Primary Consent can be presented to participants at the program's choosing and completed by participants at their convenience. At this time, all consent modules require an electronic signature from the participant.

Table 1: Overview of AoURP consent modules

Module	Addresses
Primary	Overview of all program activities. Signature indicates consent to take part in surveys and data linkage from external sources (e.g., state cancer registries), and, if invited, physical measurements, biospecimen collection (including biobanking and biomarker/genomic assays), and sensor/wearable technology activities.
HIPAA Authorization	Signature indicates consent to regular collection of electronic health records from all identifiable health care providers/entities including Part 2 (substance use disorder treatment) records and personally identifiable information (PII) from any source.
Return of Genomic Results	Signature indicates consent to receive medically-actionable genomic testing results from the program.

Each consent module is comprised of three informational components: eConsent screens, formative evaluation questions, and a form requiring signature. The eConsent screens employ visual icons, short videos, and concise, highly structured text blocks to highlight key features of program participation (Figure 1). The formative evaluation is a learning reinforcement tool focusing attention on essential concepts in research participation. Questions specifically target common misconceptions in human subject research (e.g., therapeutic misconception). With the participant's signature, the form serves as the documentation of participant's affirmative consent to take part in a given set of research activities.

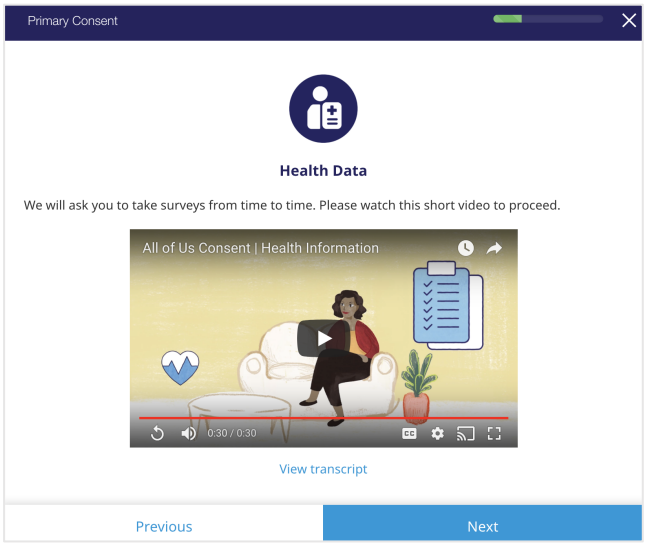


Figure 1: Example AoURP eConsent screen

1.3 *Choice of law and human subjects research*

The AoUPR's distributed structure and planned duration, when coupled with the geographic mobility of the US population, render questions regarding what research conduct is required, permissible, or prohibited challenging to resolve. Different state and territory laws might apply to the study itself over time and, likewise, to the determination of any given participant's rights over time. The desired goal of creating a unified informed consent process is further complicated by the threat of vertical conflicts of law (i.e., misalignment of local, state/territory, and federal requirements), horizontal conflicts of law (i.e., differing requirements as a participant moves from state to state or as research efforts are conducted in one location or another), as well as the varying ways in which these conflicts are resolved when disputes arise in tort or contract theory[2].

A "governing law" or "choice of law" clause allows parties to a contract to specify which jurisdiction's laws, statutes, and regulations will apply to a contract and be used for dispute resolution and thereby resolve much of the uncertainty or variability in contract interpretation. The US Department of Health and Human Services (HHS) specifies that contracted health services, including human subject research, must include a choice of law clause for work conducted overseas (i.e., outside of the US 50 states, 5 inhabited territories, and District of Columbia) [HHSAR 333.215-70(a)]. By contrast, there is guidance, e.g., the US Food and Drug Administration 21 CFR Part 50.25(d), against choice of law for studies conducted within the US.

While establishing a uniform governing law for the AoURP might be desirable for programmatic ease, this is not easily accomplished. Most injuries from research participation are based in tort theory (not contract theory), and, in tort matters, conflicts of law are typically governed by the rule of *lex loci dilicti* (or the law of the place of the injury). Research consent materials conventionally have not been framed as contracts per se but, rather, as documentation of informed consent or an assumption of risks that would be a full or partial defense to a tort action if one were to arise. Additionally, to the extent consent documents could be construed as contracts, exculpatory language that purports to function as a waiver of participants' legal rights or a limit on

[2] Conflicts of law are resolved by courts in a number of ways, including use of the *lex loci dilicti* rule, "most significant contacts" test, "comparative governmental interest" test, or a combination.

tort liability is generally not permissible (see 21 CFR 50.20). The inclusion of a choice of law provision within informed consent materials has, as a result, not been a viable solution for research in the US. For these reasons, a thorough understanding of state and territory-specific variations in regulations pertaining to human subject research is essential to meeting the program's regulatory and ethical obligations.

At the US Federal level, informed consent processes for human subject research are guided by the Common Rule [45 CFR Part 46, Subpart A] and overseen by HHS's Office for Human Research Protections (OHRP). The Common Rule contains a non-preemption clause[3] as well as direct recognition of additional state-specific informed consent requirements[4]. At least 27[5] of the 50 states, 5 inhabited territories, and District of Columbia have enacted further jurisdiction-specific regulations regarding human subject research generally, although several simply reference the Common Rule as the guidance standard (Appendix A).

The release of protected health information from covered entities[6] for research (as well as for other purposes) is regulated by the Health Insurance Portability and Accountability Act of 1996 (HIPAA) Privacy Rule [45 CFR Part 160, Subparts A and E; 45 CFR Part 164] and overseen by HHS's Office of Civil Rights (OCR). The HIPAA Privacy Rule sets forth a specific set of protections and, for a limited set of enumerated circumstances, allows for state/territory law to offer additional protections[7] [45 CFR Part 160, Subpart B]. At least 26 of the 50 states, 5 inhabited territories, and District of Columbia have specified further guidance, creating a patchwork of additional regulations across the country (Appendix A).

To enable research regarding substance use disorders to reduce stigma and advance our understanding toward more effective prevention and treatment, AoURP includes records regarding substance use disorder treatment within its request for access to a participant's protected health information records. In addition to the HIPAA Privacy Rule, release of substance use disorder records is regulated by 42 CFR Part 2, the Confidentiality of Substance Use Disorder Patient Records (Part 2) overseen by HHS's Substance Abuse and Mental Health Services Administration (SAMHSA). Part 2 details the requirements for release of these records.

Consistent with the core values of AoURP, participants will have access to the full complement of data they contribute to the program. Additionally, with participant consent, the program will interpret a limited set of data for participants; these interpreted data are considered individual research results (IRR). At this time, although the Common Rule applies to IRR equally to all other aspects of human subject research participation, the only Federal law considered by some to be specific to IRR is the Clinical Laboratory Improvement Amendments of 1988 (CLIA) although some have argued the HIPAA Privacy Rule may apply to IRR from non-CLIA certified

[3] "This policy does not affect any State or local laws or regulations which may otherwise be applicable and which provide additional protections for human subjects"

[4] "The informed consent requirements in this policy are not intended to preempt any applicable Federal, State, or local laws which require additional information to be disclosed in order for informed consent to be legally effective."

[5] This count includes regulations specific to HIV testing and status, as well as general regulations

[6] Defined by 45 CFR § 160.103 as "(1) A health plan. (2) A health care clearinghouse. (3) A health care provider who transmits any health information in electronic form in connection with a transaction covered by this subchapter."

[7] When "State law has the specific purpose of protecting the privacy of health information or affects the privacy of health information in a direct, clear, and substantial way" (i.e., the state/territory law "relates to the privacy of individually identifiable health information" as defined by HIPAA at 45 CFR 160.202) and "is more stringent" than HIPAA (as "more stringent" is defined by HIPAA at 45 CFR 160.202).

laboratories [3]. However, at least 17 states' laws further guide IRR, especially the return of genomic results (Appendix A).

There has been no comprehensive documentation of US state/territory-specific guidance and requirements to date. Therefore, to ensure compliance with all applicable Federal and state/territory guidance and regulation, AoURP consulted with OHRP, OCR, and SAMHSA, sought guidance from the NIH Office of General Counsel, and conducted an independent legal review of the informed consent and HIPAA Authorization processes for this national research program.

2. Implementation

We have developed a "parent" version of each consent module. Parent module versions are consistent with the greatest number of state and territory regulations. However, some states and territories have regulations that, if applied to other jurisdictions, might be considered to limit or additionally burden participants. To address these distinctive requirements, we have modified the parent version of modules, creating specific "child" versions of modules for use in those jurisdictions.

2.1 *State/territory compliant primary consent*

To determine the prospective participant's pathway through the program's informed consent modules, we ask participants a series of questions. First, we ask participants their state or territory of residence. Those who answer California are presented an Experimental Subject's Bill of Rights as described by the Protection of Human Subjects in Medical Experimentation Act (California Health and Safety Code 24170-24179.5) in advance of the primary consent. We then ask the participant to confirm they have reached the age of majority for research participation within their state or territory of residence: 18 years of age in all US states and territories with the exception of Alabama (age 19) and Puerto Rico (age 21) (Appendix A). Of note, the Northern Mariana Islands do not have regulations regarding the age of majority; we have elected to use age 18, consistent with the majority of other states and territories. Finally, we ask participants the state or territory in which they receive most of their healthcare.

2.2 *State/territory compliant HIPAA Authorization/Part 2 data release*

We link the version of HIPAA Authorization/Part 2 data release to the state or territory in which the participant reports receiving most of their healthcare. The majority of state- and territory-specific regulations additional to the HIPAA Privacy Rule focus on the term of expiry for the HIPAA Authorization, with states requiring a specific date of expiry or specific term of expiry where the HIPAA Privacy Rule allows for an event of expiry (e.g., the end of the research project). Please see Appendix B for further detail. Of note, the Illinois statute that requires a date of expiry (the Mental Health and Developmental Disabilities Confidentiality Act, 740 ILCS 110), relates only to "therapists." Therapist is defined as, "a psychiatrist, physician, psychologist, social worker, or nurse providing mental health or developmental disabilities services or any other person not prohibited by law from providing such services or from holding himself out as a therapist if the recipient reasonably believes that such person is permitted to do so"[740 ILCS 110/2 from Ch. 91 1/2, par. 802 section 2], however state convention is to apply this requirement to all Authorizations.

A subset of states require that the release of "sensitive data" such as HIV status, drug and alcohol use, and sexual history be specifically highlighted to the signatory of the release (i.e., MA

104 CMR 31. 05; ORS 192.566; Tex. Bus and Com code 602.051). While the release of these data are highlighted within the parent version of AoURP HIPAA Authorization form to all participants, participants in Massachusetts, Oregon, and Texas are additionally presented with a "sensitive data confirmation" screen as part of the HIPAA Authorization eConsent (Appendix B).

It is important to note that the AoURP HIPAA Authorization does not provide participants the option of granular release of electronic health records; participants either agree to the release of all available records or they decline to give permission for any of their records' release. This decision was taken by the program based on the program's core principle of transparency and the technical difficulty of ensuring a completely "clean" data release. We did not want to allow participants the opportunity to request the hold back specific classes of health information only to have that information inadvertently released, for example, within a free-text clinician report about treatment for a separate condition.

Finally, also based on the IL Mental Health and Developmental Disabilities Confidentiality Act, Illinois convention is that HIPAA Authorizations require a "witness signature" in addition to the signature of the participant themselves (Appendix B). The witness can be any person who can attest to the identity of the participant. Interestingly, in Illinois, based on the same statute, withdrawal of consent is also conventionally interpreted to require a witness signature.

2.3 *State/territory compliant consent for the return of genomic results*

AoURP participants may consent to receive medically-actionable genomic testing results, a form of IRR. The specific set of medically actionable results are defined by the program based on professional society guidelines and similar sources (e.g., those of the American College of Medical Genetics and Genomics [4]), and will evolve over time. Given the additional potential risks and benefits the return of medically actionable findings may pose to participants [5], AoURP will use an explicit opt-in informed consent module for the return of genomic results.

Among the relevant state and territory regulations that govern the return of genomic results (Appendix 1), many do not specify if they pertain to clinical care, research endeavors, or both. Further muddying the waters, definitions of genetic information vary [6]. AoURP has elected to use the broad federal definition referenced in the Genetic Information Nondiscrimination Act (GINA) of genetic information which includes family history in addition to information regarding genetic tests [42 U.S.C. § 300gg-91].

Most jurisdiction-specific laws require that the informed consent process for the return of genomic results include a general purpose or description of the genetic tests to be performed, as well as potential uses and limitations of those tests [e.g., Del. Code 16 §1201 (4)]. However, both the State of New York and Commonwealth of Massachusetts require that the consent process include a description, "of each specific disease or condition tested for" [NYCL (CVR) §79-L(2)(b); MGL Public Health 111 §70G(a)]. Notably, NYCL (CVR) §79-L(2)(f) allows for modification of this requirement if, "the research protocol does not permit such degree of specificity." Additionally, NYCL (CVR) §79-L(9)(a) provides that, "samples may be used for tests other than those for which specific consent has been obtained for purposes of research conducted in accordance with applicable law and regulation and pursuant to a research protocol approved by an institutional review board [IRB] provided that the individuals who provided the samples have given prior written informed consent… and did not specify time limits or other factors that would restrict use of the sample for the test." Thus, a broad description of the diseases

or conditions tested for is allowed under IRB oversight for participants within the State of New York.

In the case of Massachusetts, there is no explicit clause within MGL Public Health 111 §70G that specifies any ability to modify the requirement for inclusion of a general description of each specific disease or condition tested for within the consent process. However, current research convention mirrors New York's: with the oversight of an IRB, participants of genomic research are consented to the return of genomic results of broad description. In practice, both the AoURP parent eConsent and consent form for the return of genomic results will link out to an inventory of conditions being tested for with explicit notation that this list may be updated over time. Additionally, in consideration of subpart (c) of the Massachusetts statue, this inventory will address for all participants each tests' reliability and predictive value.

Massachusetts further specifics a discussion with, "the medical practitioner ordering the test" regarding the reliability and certainty of test results prior to consent. Given the research context of AoURP's return of genomic results, genetic counseling will be made available to all participants prior to completing the consent process, regardless of their state of residence, but will not be required. This is also consistent current practice in Massachusetts.

In FLA. Stat. Ann 760.40(3), the State of Florida sets forth a number of requirements for DNA analysis and the return of results.[8] Two of these requirements are incorporated into the parent version of the return of genomic results consent process for all AoURP participants. First, AoURP will enable participants to track the journey of their sample from receipt by the biobank, to analysis for tests specified in the return of genomic results inventory, to its receipt by the genetic counseling core and/or deposit in their AoURP participant record. Secondly, the parent consent form includes a statement that AoURP, as a research program, is not engaged in any decisions to grant or deny insurance, employment, mortgage, loan, credit, or educational opportunities and, therefore, that these results will not be used for those purposes by the program.

The one FLA. Stat. Ann 760.40(3)-required customization of the return of genomic results consent process not incorporated into the parent consent process will be accommodated by an addition to the eConsent (Appendix B). Within the eConsent process, residents of the state of Florida will be able to specify a healthcare provider to whom the participant would like their results sent [FL 760.40 (3)]. This feature will likely be made available to all participants (once trialed in Florida), pending review of relevant state-specific considerations. In the interim, study participants may independently choose to share their test results with healthcare providers.

3. Conclusion

The *All of Us* Research Program is an ambitious national cohort study designed to accelerate understanding of human health. The diversity of laws, statutes, and regulations across the US challenge large, dispersed research efforts such as AoURP in ways not unlike those faced by international research efforts [7]. Creating a pattern of distinct informed consent interactions over time, with each consent module having its own specific ask, including potential risks, benefits, and set of scientific "unknowns" that arise naturally in cutting edge research, supports participant

[8] FLA. Stat. Ann 760.40(3), a civil rights statute which predates GINA, was drafted to prevent "surreptitious," and potentially discriminatory, genetic testing without a focus on the statues' potential implications for research. For this reason, IRBs in Florida have generally rejected a narrow interpretation of this statute when considering research initiatives like AoURP.

autonomy while allowing for flexibility in the face of legal and regulatory uncertainty. Empirical legal research will be essential to facilitate this and similar biomedical research efforts and to enable research teams in their efforts to respect and promote participant's rights.

There are several limits to our analysis. First and foremost, despite having consulted with experts across the nation, there is no central clearinghouse or curated resource for the most current US state/territory research regulations. Additionally, as we noted in our analysis, it is sometimes difficult to tease apart state/territory requirements and convention. As the clinical and research genetics community knows well, few of these rules and regulations have been adequately stress-tested in the courtroom, leaving a dearth of guidance for researchers and policy makers alike. It is also important to note that while this analysis is, to the best of our knowledge, complete as of January 1, 2018, laws, technologies, research practices, and societal norms are constantly evolving; AoURP will engage in regular re-review of its consent materials and approaches to ensure their currency.

4. Acknowledgements

The research reported in this publication was supported by the Department of Health and Human Services of the National Institutes of Health under award number U24OD023176, by the National Human Genome Research Institute (NHGRI) Grant No. 5R00HG006446-05, and by the National Institutes of Health, Office of the Director via 1OT2OD024609-01. Approval for the release of this research was granted by the *All of Us* Research Program Publication Board (Approval date: July 3, 2018). The authors take full responsibility for any errors or omissions within this manuscript; the content does not necessarily represent the official views of the National Institutes of Health.

The authors acknowledge with gratitude: Eva Yin, Patrick Kayne, and Farrah Gerdes, attorneys at Wilson Sonsini Goodrich & Rosati, P.C.; Katherine Blizinsky, Policy Director, *All of Us* Research Program, National Institutes of Health; Stephanie Devaney, Deputy Director, *All of Us* Research Program, National Institutes of Health; the Consent Work Group of the *All of Us* Research Program; *All of Us* Research Program Consortium members, especially Laura Beskow, Louise Bier, Robert C. Green, Mitch Dean, Joyce Ho, Rosario Isasi, Shenela Lakhani, and Matthew Lebo; Erin C. Fuse Brown, Professor of Law, Georgia State University College of Law; and Leslie E. Wolf, Professor of Law and Director for the Center for Law, Health & Society, Georgia State University College of Law.

5. Appendices

Appendix A: State/territory laws informing the *All of Us* Research Program primary informed consent process, HIPAA Authorization, and Return of Genomic Results consent

Domain	State	Statue
Age of Majority	Alabama	Ala. Code § 26-1-1- Infants and incompetents Although not germane to AoURP, nb subpart (f), "a person who is 18 years of age or older may consent to participate in research conducted by a college or university that is accredited by a federally recognized accrediting agency if the research has been approved by the Institutional Review Board of the institution."
	Puerto Rico	31 L.P.R.A. §971
Bill of Rights	California	California Health and Safety Code 24170-24179.5
Primary Consent	Alabama	AL Code § 22-56-4 AL Code § 22-11A-51; § 22-11A-53
	Arizona	AZ Rev Stat § 36-663
	California	Cal Health & Safety Code § 24173; Cal Pen Code § 3521 CA Health & Safety Code § 121075; § 121105
	Colorado	Col Rev Stat § 25-4-410
	Connecticut	CT Gen Stat § 19a-583; § 19a-585; § 19a-582
	Delaware	16 DE Code § 715
	District of Columbia	DC Code § 7-1305.09
	Guam	Ch 24 Guam Research Review Board § 24106
	Hawaii	HI Rev Stat § 325-16
	Illinois	Illinois Mental Health and Developmental Disabilities Confidentiality Act; AIDS Confidentiality Act 410 ILCS 50/3.1 410 ILCS 305/8
	Kansas	KS Code § 65-4974
	Massachusetts	104 CMR 31.05 ALM GL ch. 111, § 70F
	Montana	TITLE 53. SOCIAL SERVICES AND INSTITUTIONS CHAPTER 21. MENTALLY ILL 53-21-147
	Nebraska	NE Rev Stat § 71-531
	New Hampshire	NH Rev Stat 141-F:5
	New Jersey	NJ Stat. § 26:14-4; N.J. Stat. § 26:14-5
	New Mexico	NM Stat § 24-2B-2
	New York	NY CLS Pub Health § 2441; NY CLS Pub Health § 2442 NY CLS Pub Health § 2781; NY CLS Pub Health § 2782
	Oklahoma	63 OK Stat § 63-3102A
	Oregon	ORS 433.075
	Pennsylvania	35 P.S.§ 7605
	South Carolina	SC Code § 44-26-180
	South Dakota	SD Codified L § 27B-8-41.
	Texas	Texas Health & Safety Code § 81.105; §81.106.
	Virginia	VA Code Ann. § 32.1-162.20; § 32.1-162.16; § 32.1-162.18
	Washington	RCW § 70.24.330
	West Virginia	WV Code § 16-3C-2

(continued next page)

Appendix A (continued)

Domain	State	Statue
HIPAA Authorization	Alabama	AL Code § 22-11A-22; AL Code § 22-11A-54
	Arizona	AZ Rev. Stat. § 36-664
	California	CA Civ Code § 56.10
	Colorado	CO Rev Stat § 25-1-1201
	Connecticut	CT Insurance Information and Privacy Protection Act § 19a-581; § 19a-585; Conn. Gen. Stat. § 20-7c; Conn. Gen. Stat. § 52-146g
	Delaware	Del. Code Ann. tit. 16, § 717; Del. Code § 1212
	District of Columbia	DC Code § 7-1605; § 7-1203.06
	Florida	FL Stat § 381.004
	Georgia	GA Code § 24-12-2; § 24-12-21; § 24-12-12; § 31-33-8; § 37-4-125
	Hawaii	HI Rev Stat Ann § 325-101
	Illinois	Personal Information Protection Act § 50; 735 ILCS 5/8-2001
	Indiana	IN Code § 16-39-2-5
	Iowa	Iowa Code § 228.2; § 228.3; § 228.4; § 141A.9
	Louisiana	LA Rev Stat. § 22:1023
	Maine	ME Rev Stat § 1711-C
	Maryland	MD HEALTH-GENERAL Code Ann. § 4-303
	Minnesota	MN Stat § 144.293;144.294;144.295
	Montana	MT Code § 50-16-502; § 50-16-527; § 50-16-1009
	New Mexico	NM Stat § 24-2B-6; § 24-2B-7; § 43-1-19; § 24-14B-6
	Ohio	OH Rev Code § 3701.17;§ 3701.243; § 5119.27
	Oklahoma	OK Stat §43A-1-109
	Oregon	OR Rev Stat § 192.553; § 192.556; §192.566; § 431A.865
	Pennsylvania	Title 35 P.S. Health and Safety § 7607
	Puerto Rico	Title 26 Subtitle 3 Chapter 112 § 9240
	Rhode Island	RI Gen L § 5-37.3-4
	Texas	INS § 602.051
Return of genomic results[9]	Alaska	AS §18.13.010
	Delaware	Del. Code 16 §1201 et seq.
	Florida	FS §760.40(2)(a); FS §760.40(3)
	Georgia	OCGA §33-54-3(b)
	Iowa	Iowa Code §§729.6
	Massachusetts	MGL Public Health 111 §70G(a)
	Michigan	MCL §333.17520(2)
	Minnesota	MS §13.386 Subd.3(a)
	Nebraska	NRS §71-551(1)
	Nevada	NRS §629.151; §629.161; §629.181; §629.101 et seq.
	New Hampshire	NHS §141-H:1; NHS § 141-H:2
	New Jersey	NJ Rev Stat §10:5-45
	New Mexico	NMSA §24-21-3
	New York	NYCL (CVR) §79-L(2)(b); NYCL (CVR) §79-L(9)(c); NYCL (CVR) §79-L(9)(e)
	Oregon	ORS §192.535; ORS §192.538(5)
	South Carolina	SCCL §38-93 et seq.
	South Dakota	SDCL §34-14-22

[9] Note: regulations related to disclosure authorizations and genetic information definitions are not included in this listing

Appendix B: Summary of State-Specific Variations of the AoURP Consent Process

States/Territories	Primary Consent			HIPAA Authorization		Return of Genomic Results Consent	
	Bill of Rights	*eConsent version*	*Form version*	*eConsent version*	*Form version*	*eConsent version*	*Form version*
AL, AK, AZ, AR, CO, CT*, DC, GA*, HI, ID, IA, KS, KT, MI, MS, MO, NE**, NV, NH, NJ, NM, NY, NC, ND, PA, RI, SC, SD, TN, UT, VT, VA**, WV, WI, Puerto Rico, US Virgin Islands, Guam, American Samoa, Northern Mariana Islands	none	Parent	Parent	Parent	Parent	Parent	Parent
CA	required	Parent	Parent	Parent	Date of expiry: Standard	Parent	Parent
DL, IN, LA, MN, OH, OK, WA	none	Parent	Parent	Parent	Date of expiry: Standard	Parent	Parent
MA, OR, TX	none	Parent	Parent	Sensitive Data Confirmation	Parent	Parent	Parent
ME, MT**	none	Parent	Parent	Parent	Date of expiry: 30-month	Parent	Parent
MD, WY	none	Parent	Parent	Parent	Date of expiry: 12-month	Parent	Parent
IL	none	Parent	Parent	Witness Signature	Access to records	Parent	Parent
FL	none	Parent	Parent	Parent	Parent	Share with healthcare provider	Parent

*In Connecticut and Georgia HIPAA Authorizations are valid for one year from their date of signature to request of records from insurance providers.

** In Montana, Nebraska, and Virginia HIPAA Authorizations are valid for two years from their date of signature for the request of records from insurance providers.

Other notes:

- In the states of Maine and Montana, HIPAA Authorizations are only valid 30 months (in Montana, only if expiry date is provided). Given the nature of rolling enrollment, we will update the form used by those in Maine and Montana on an annual basis to state a date 30 months from January 1st of the enrollment year. At the date of expiry (30 months from January 1st of the enrollment year), all persons consented that calendar year would be contacted for re-authorization on a form listing a date 30 months hence. For example, the 2018 form will expire on 7/1/2020. Those consented in 2018 would be asked to re-sign a form 7/1/2020 expiring 12/31/2022.
- In the states of Maryland and Wyoming, HIPAA Authorizations are only valid for one year. Annually, people who receive care in Maryland or Wyoming will be invited to re-sign the same form with a 12-month expiry (but no date) listed.

References

1. N. Esipova, A. Pugliese, and J. Ray, “381 Million Adults Worldwide Migrate Within Countries” (2013). https://news.gallup.com/poll/162488/381-million-adults-worldwide-migrate-within-countries.aspx.
2. Americans Moving at Historically Low Rates, Census Bureau Reports (US Census Beureau, 2016). https://www.census.gov/newsroom/press-releases/2016/cb16-189.html.
3. Return of Research Results (National Human Genome Research Institute, 2017). https://www.genome.gov/27569049/return-of-research-results/.
4. S. S. Kalia, K. Adelman, S. J. Bale, W. K. Chung, C. Eng, J. P. Evans, G. E. Herman, S. B. Hufnagel, T. E. Klein, B. R. Korf, K. D. McKelvey, K. E. Ormond, C. S. Richards, C. N. Vlangos, M. Watson, C. L. Martin, and D. T. Miller, *Genet. Med.*, **19**(2), 249–255 (2017).
5. Genetic Information Nondiscrimination Act of 2008. Pub. L. 110-233, 122 Stat. 881.
6. A. E. R. Prince, *Brooklyn Law Rev.*, **79**(1) 175-227 (2013).
7. B. M. Knoppers, J. R. Harris, I. Budin-Ljøsne, and E. S. Dove, *Hum. Genet.***133**(7), 895–903 (2014).

Merging heterogeneous clinical data to enable knowledge discovery

Martin G. Seneviratne
Department of Biomedical Data Science, Stanford University
1265 Welch Rd, Stanford
CA 94305, United States
Email: martsen@stanford.edu

Michael G. Kahn
Colorado Clinical and Translational Sciences Institute
Denver, CO 80045, United States
Email: michael.Kahn@ucdenver.edu

Tina Hernandez-Boussard*
Department of Medicine, Biomedical Informatics, Stanford University
1265 Welch Rd, Stanford
CA 94305, United States
Email: boussard@stanford.edu

The vision of precision medicine relies on the integration of large-scale clinical, molecular and environmental datasets. Data integration may be thought of along two axes: data fusion across institutions, and data fusion across modalities. Cross-institutional data sharing that maintains semantic integrity hinges on the adoption of data standards and a push toward ontology-driven integration. The goal should be the creation of query-able data repositories spanning primary and tertiary care providers, disease registries, research organizations etc. to produce rich longitudinal datasets. Cross-modality sharing involves the integration of multiple data streams, from structured EHR data (diagnosis codes, laboratory tests) to genomics, imaging, monitors and patient-generated data including wearable devices. This integration presents unique technical, semantic, and ethical challenges; however recent work suggests that multi-modal clinical data can significantly improve the performance of phenotyping and prediction algorithms, powering knowledge discovery at the patient- and population-level.

Keywords: Data fusion, interoperability, multi-modal data, big data, phenotyping

The quantity of digitized health information has increased exponentially over the past decade, with growing data repositories across all sectors of the health system [1]. The rise of electronic health records has enabled the creation of large datasets containing structured, semi-structured and unstructured data, ranging from diagnostic codes and laboratory results to continuous monitoring signals, clinical notes, medical imaging and pathology. However, there are also rich clinical, molecular and environmental datasets held by government agencies, disease registries, employers, pharmaceutical companies and research organizations. Meanwhile, the proliferation of health tracking apps, wearables and home sensors have created new clinical data streams controlled by the patient, which capture granular information about lifestyle and micro-environmental exposures. Even an individual's social media footprint may be considered as a source of clinical insights. Weber *et al.* have described the spectrum of clinical data available for an individual as a "tapestry of high-value information sources" ranging from the micro (genomic/molecular data) through to the macro (behavioral/lifestyle data) [2].

Many have predicted that the convergence of rich clinical, molecular and environmental data streams will accelerate knowledge discovery in biomedicine and help us to move toward the high-level goal of precision medicine [3,4]. Certainly, larger datasets combining information from numerous sources will improve the performance of diagnostic and prognostic machine learning algorithm, fuelling observational research and improving clinical decisions at the point of care. The critical challenge is how to integrate disparate clinical data streams in a flexible, query-able format while preserving patient privacy and data governance. This integration challenge may be thought of along two axes: data fusion across institutions, and data fusion across modalities.

The first challenge involves cross-institutional data sharing. Federal incentive programs launched through the Health Information Technology for Economic and Clinical Health (HITECH) Act supported the creation of health information exchanges (HIEs) as a platform for clinical data sharing; however based on a 2015 survey, only 23% of HIEs currently supported research, with a further 47% planning to support secondary use in the future [5]. Furthermore, a 2016 review found that the number of HIEs had declined between 2012 and 2014 and only half report being financially sustainable [6]. In 2015, the Office of the National Coordinator of Health IT (ONC) published an Interoperability Roadmap, which outlines a national agenda for improving health information exchange [7]. One key objective is achieving syntactic and semantic interoperability by adoption of common vocabularies, including SNOMED-CT and RxNorm, and common data formats, including consolidated clinical document architecture (C-CDA) and Fast Health Interoperability Resources (FHIR). The roadmap also calls for the adoption of secure transport standards and outlines best practices for matching patient identities between sites. In parallel, there have been a number of academic endeavors to build platforms for observational clinical research, including the Observational Health Data Sciences and Informatics (OHDSI) network [8], SHARPn project [9], and the Informatics for Integrating Biology and the Bedside (i2b2) initiative [10].

An emerging theme throughout these cross-institutional data fusion efforts, from industry to academia, is the power of ontology-driven data integration, inspired by the rise of semantic web technologies [11–13]. This approach has a number of distinct advantages including the ability to synthesize across many disparate data sources via high-level ontologies and the ability to reason over a knowledge base [14]. Ongoing technical challenges include representing data provenance, temporal relationships and data quality [15]; however the prevailing challenge is operational - how to shift organizational culture toward interoperability and data sharing [16]. Beyond this, the infrastructure for interoperability may vary, with successful examples of centralized data warehouses [17], decentralized blockchain-based health records systems [18], and patient-controlled health records [19].

The second major component of data fusion is cross-modality integration. Most EHRs contain a diversity of data types that have traditionally been analyzed independently, ranging from structured diagnosis codes to signal data, clinical notes and imaging. Furthermore, the interoperability advances mentioned above are making it possible to harmonize traditional EHR data with novel clinical data streams including genomic, microbiome, metabolic and patient-generated health data (PGHD). There is an expanding evidence base showing that multi-modal data integration can support precision medicine by stratifying patients based on their ‘deep phenotype’ [20]; improving the performance of clinical decision support algorithms for diagnosis and prediction [21]; and uncovering new phenotypes altogether [22]. For example, Zhao *et al.* developed a risk prediction model for cardiovascular events using EHR data, but found a significant performance boost when those data were fused with

patient-level genomic information [23]. Meanwhile, by using unsupervised learning on a combined dataset of metabolome, microbiome, genetics and imaging data, Shomorony *et al.* were able to identify a signature of biomarkers that identified diabetic patients more accurately than traditional clinical metrics (glucose, insulin resistance, and body-mass-index) - suggesting novel pathways that may be involved in the development of diabetes [24].

The combination of traditional health data with PGHD or social media data has enabled knowledge discovery in the realms of both precision medicine and population health. Santillana *et al.* combined hospital visit data with Twitter, Google searches, and posts on an online health forum to predict influenza incidence [25]. Vilar *et al.* describe efforts to identify drug-drug interactions by combining social media posts with the biomedical literature [26]. On a more granular level, there is a push to integrate patient-reported outcomes (PROs) into EHRs as a way to promote patient-centric care (an example of heterogeneous data fusion potentially driving behavior change) [27] which has fueled interesting insights into the relationship between PROs and clinical outcomes such as mortality [28]. The rise of the 'Internet of Things' in healthcare - the ecosystem of connected monitoring devices that surround a patient - as well as ambient information such as geo-location are creating opportunities for even richer multi-modal datasets [29–31]. These data no longer reside exclusively in hospitals. Private sector initiatives such as Verily's Project Baseline and Apple's HealthKit program are enabling patients to aggregate multiple medical data sources [32,33]. Meanwhile, the *All Of Us* initiative is a National Institutes of Health program to collect molecular, clinical and environmental data on a diverse cohort of volunteers for research purposes [34]. As the pathophysiology behind chronic disease is a complex interplay of clinical, molecular and behavioral factors acting over extended time periods, the datasets required to tackle the global epidemic of chronic disease will need to be similarly layered and sophisticated. There is both a clinical opportunity and an economic one, with increasing evidence to suggest that data integration can reduce overall healthcare costs [35].

Cross-modality data integration is associated with a number of challenges, of which we highlight three below. First, there is the issue of how to harmonize data from distant parts of a knowledge graph reflecting radically different levels of abstraction e.g. diagnosis codes (high-level) with proteomic data (low-level). This creates challenges for data storage and makes it difficult to generate feature vectors to train classifiers. Several recent studies have shown that deep learning can be used to create efficient abstract representations of structured and unstructured EHR data, for example the *DeepPatient* representation using stacked denoising autoencoders [36]. A similar approach might be considered for a broader range of input data. A second caveat is around data stewardship, particularly with respect to privacy and security [37]. Fusion of data streams may accelerate scientific discovery and clinical care, but this comes with an increased risk of patient re-identification. Further work is needed around de-identification, consent processes and access control when data are contributed to shared repositories. The increasing volume of digital health information available to clinicians also raises questions around liability and duty of care i.e. the extent to which clinicians are responsible for the full expanse of information in an aggregated health repository. A third challenge is around equity and inclusion. A 2018 report by Ferryman *et al.* on 'Fairness in precision medicine' highlights the potential for bias in large-scale biomedical training data, stemming from historical discrimination in the health system and recruitment biases at academic medical centers [38]. Data-fusion efforts must be cognizant of the distribution of important demographic variables, such as gender, ethnicity and socioeconomic status in their input data.

The fusion of heterogeneous datasets from different institutions and across different modalities presents a powerful opportunity to drive knowledge discovery in biomedicine. There are technical and operational challenges to enable data sharing across borders of institutional ownership, which we are beginning to overcome with interoperability standards and data sharing platforms. Arguably the more nuanced problem today is how to grapple with extremely diverse data types that encompass the micro and macro scales of a patient's data signature, including how to create flexible data storage and machine learning architectures, and how to design stewardship processes to govern these data appropriately. Holzinger *et al.* claimed in 2014 that "biomedical research is drowning in data, yet starving for knowledge". Today we have more health data than ever before, but the challenge remains how to harmonize, structure and learn from multi-modal datasets [39].

Acknowledgements

This work is partially supported by the National Cancer Institute of the National Institutes of Health (NIH) under Award Number R01CA183962 and by NIH/NCATS Colorado CTSA Grant Number UL1 TR002535. Contents are the authors' sole responsibility and do not necessarily represent official NIH views.

References

1. Murdoch TB, Detsky AS. The inevitable application of big data to health care. JAMA. 2013;309: 1351–1352.
2. Weber GM, Mandl KD, Kohane IS. Finding the missing link for big biomedical data. JAMA. 2014;311: 2479–2480.
3. Roski J, Bo-Linn GW, Andrews TA. Creating value in health care through big data: opportunities and policy implications. Health Aff . 2014;33: 1115–1122.
4. Obermeyer Z, Emanuel EJ. Predicting the Future - Big Data, Machine Learning, and Clinical Medicine. N Engl J Med. 2016;375: 1216–1219.
5. Parker C, Reeves M, Weiner M, Adler-Milstein J. Health Information Exchange Organizations and Their Support for Research: Current State and Future Outlook. Inquiry. SAGE Publications; 2017;54. doi:10.1177/0046958017713709
6. Adler-Milstein J, Lin SC, Jha AK. The Number Of Health Information Exchange Efforts Is Declining, Leaving The Viability Of Broad Clinical Data Exchange Uncertain. Health Aff . 2016;35: 1278–1285.
7. Connecting Health and Care for the Nation A Shared Nationwide Interoperability Roadmap. Office of the National Coordinator for Health Information Technologies; 2015. Report No.: 1.0.
8. Hripcsak G, Duke JD, Shah NH, Reich CG, Huser V, Schuemie MJ, et al. Observational Health Data Sciences and Informatics (OHDSI): Opportunities for Observational Researchers. Stud Health Technol Inform. 2015;216: 574–578.
9. Rea S, Pathak J, Savova G, Oniki TA, Westberg L, Beebe CE, et al. Building a robust, scalable and standards-driven infrastructure for secondary use of EHR data: the SHARPn project. J Biomed Inform. 2012;45: 763–771.
10. Murphy SN, Weber G, Mendis M, Gainer V, Chueh HC, Churchill S, et al. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). J Am Med Inform Assoc. 2010;17: 124–130.
11. Mate S, Köpcke F, Toddenroth D, Martin M, Prokosch H-U, Bürkle T, et al. Ontology-based data integration between clinical and research systems. PLoS One. 2015;10: e0116656.
12. Hsu W, Gonzalez NR, Chien A, Pablo Villablanca J, Pajukanta P, Viñuela F, et al. An integrated, ontology-driven approach to constructing observational databases for research. J Biomed Inform. 2015;55: 132–142.
13. Zhang H, Guo Y, Li Q, George TJ, Shenkman E, Modave F, et al. An ontology-guided semantic data integration framework to support integrative data analysis of cancer survival. BMC Med Inform Decis Mak. 2018;18: 41.
14. Lezcano L, Sicilia M-A, Rodríguez-Solano C. Integrating reasoning and clinical archetypes using OWL ontologies and SWRL rules. J Biomed Inform. 2011;44: 343–353.
15. Johnson SG, Speedie S, Simon G, Kumar V, Westra BL. A Data Quality Ontology for the Secondary Use

of EHR Data. AMIA Annu Symp Proc. 2015;2015: 1937–1946.
16. Ong T, Pradhananga R, Holve E, Kahn MG. A Framework for Classification of Electronic Health Data Extraction-Transformation-Loading Challenges in Data Network Participation. EGEMS (Wash DC). 2017;5: 10.
17. Seneviratne M, Seto T, Blayney DW, Brooks JD, Hernandez-Boussard T. Architecture and Implementation of a Clinical Research Data Warehouse for Prostate Cancer. eGEMs (Generating Evidence & Methods to improve patient outcomes). 2018;6.
18. Azaria A, Ekblaw A, Vieira T, Lippman A. MedRec: Using Blockchain for Medical Data Access and Permission Management. 2016 2nd International Conference on Open and Big Data (OBD). 2016. pp. 25–30.
19. Chan D, Howard M, Dolovich L, Bartlett G, Price D. Revolutionizing patient control of health information. Can Fam Physician. 2013;59: 823–824.
20. Robinson PN, Mungall CJ, Haendel M. Capturing phenotypes for precision medicine. Cold Spring Harb Mol Case Stud. 2015;1: a000372.
21. Rajkomar A, Oren E, Chen K, Dai AM, Hajaj N, Hardt M, et al. Scalable and accurate deep learning with electronic health records. npj Digital Medicine. 2018;1: 18.
22. Beaulieu-Jones BK, Greene CS. Semi-supervised learning of the electronic health record for phenotype stratification. J Biomed Inform. The Author(s); 2016;64: 168–178.
23. Zhao J, Feng Q, Wu P, Lupu R, Wilke RA, Wells QS, et al. Learning from Longitudinal Data in Electronic Health Record and Genetic Data to Improve Cardiovascular Event Prediction [Internet]. bioRxiv. 2018. p. 366682. doi:10.1101/366682
24. Shomorony I, Cirulli ET, Huang L, Napier LA, Heister RR, Hicks M, et al. Unsupervised integration of multimodal dataset identifies novel signatures of health and disease [Internet]. bioRxiv. 2018. p. 432641. doi:10.1101/432641
25. Santillana M, Nguyen AT, Dredze M, Paul MJ, Nsoesie EO, Brownstein JS. Combining Search, Social Media, and Traditional Data Sources to Improve Influenza Surveillance. PLoS Comput Biol. 2015;11: e1004513.
26. Vilar S, Friedman C, Hripcsak G. Detection of drug-drug interactions through data mining studies using clinical sources, scientific literature and social media. Brief Bioinform. 2018;19: 863–877.
27. Gensheimer SG, Wu AW, Snyder CF, Basch E, Gerson J, Holve E, et al. Oh, the Places We'll Go: Patient-Reported Outcomes and Electronic Health Records. The Patient - Patient-Centered Outcomes Research. 2018; doi:10.1007/s40271-018-0321-9
28. Basch E, Deal AM, Dueck AC, Scher HI, Kris MG, Hudis C, et al. Overall Survival Results of a Trial Assessing Patient-Reported Outcomes for Symptom Monitoring During Routine Cancer Treatment. JAMA. 2017;318: 197–198.
29. J. Andreu-Perez, D. R. Leff, H. M. D. Ip, G. Yang. From Wearable Sensors to Smart Implants--Toward Pervasive and Personalized Healthcare. IEEE Transactions on Biomedical Engineering. 2015;62.
30. Schinasi LH, Auchincloss AH, Forrest CB, Diez Roux AV. Using electronic health record data for environmental and place based population health research: a systematic review. Ann Epidemiol. 2018;28: 493–502.
31. Saelens BE, Arteaga SS, Berrigan D, Ballard RM, Gorin AA, Powell-Wiley TM, et al. Accumulating Data to Optimally Predict Obesity Treatment (ADOPT) Core Measures: Environmental Domain: ADOPT: Environmental Domain. Obesity . 2018;26: S35–S44.
32. Barr A. Google to Collect Data to Define Healthy Human. Wall Street Journal. wsj.com; 27 Jul 2014. Available: https://www.wsj.com/articles/google-to-collect-data-to-define-healthy-human-1406246214. Accessed 16 Oct 2018.
33. North F, Chaudhry R. Apple HealthKit and Health App: Patient Uptake and Barriers in Primary Care. Telemed J E Health. 2016;22: 608–613.
34. Collins FS, Varmus H. A new initiative on precision medicine. N Engl J Med. 2015;372: 793–795.
35. Bates DW, Saria S, Ohno-Machado L, Shah A, Escobar G. Big data in health care: using analytics to identify and manage high-risk and high-cost patients. Health Aff . 2014;33: 1123–1131.
36. Miotto R, Li L, Kidd BA, Dudley JT. Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. Sci Rep. 2016;6: 26094.
37. Ross MK, Wei W, Ohno-Machado L. "Big data" and the electronic health record. Yearb Med Inform. 2014;9: 97–104.
38. Ferryman K, Pitcan M. Fairness in Precision Medicine. Data & Society; 2018 Feb.
39. Holzinger A, Jurisica I. Knowledge Discovery and Data Mining in Biomedical Informatics: The Future Is in Integrative, Interactive Machine Learning Solutions. In: Holzinger A, Jurisica I, editors. Interactive Knowledge Discovery and Data Mining in Biomedical Informatics: State-of-the-Art and Future Challenges. Berlin, Heidelberg: Springer Berlin Heidelberg; 2014. pp. 1–18.

Workshop during the Pacific Symposium of Biocomputing, Jan 3-7, 2019: Reading between the genes: interpreting non-coding DNA in high-throughput

Joanne Berghout[†] , Yves A. Lussier[†], Francesca Vitali[†]
Center for Biomedical Informatics and Biostatistics, Dept. of Medicine, University of Arizona, 1230 Cherry Ave, Tucson, AZ 85719, USA
Emails: jberghout@email.arizona.edu, yves@email.arizona.edu, francescavitali@email.arizona.edu

Martha L. Bulyk[†]
Division of Genetics, Dept. of Medicine & Dept. of Pathology
Brigham and Women's Hospital and Harvard Medical School, Boston, MA 02115, USA
Email: mlbulyk@genetics.med.harvard.edu

Maricel G. Kann[†]
Dept. of Biological Sciences, 1000 Hilltop Circle
University of Maryland, Baltimore County, Baltimore, MD 02115, USA
Email: mkann@umbc.edu

Jason H. Moore[†]
Institute for Biomedical Informatics, 3400 Civic Center Blvd, Bldg 421
University of Pennsylvania, Philadelphia, PA 19104, USA
Email: jhmoore@upenn.edu

Identifying functional elements and predicting mechanistic insight from non-coding DNA and non-coding variation remains a challenge. Advances in genome-scale, high-throughput technology, however, have brought these answers closer within reach than ever, though there is still a need for new computational approaches to analysis and integration. This workshop aims to explore these resources and new computational methods applied to regulatory elements, chromatin interactions, non-protein-coding genes, and other non-coding DNA.

Keywords: non-coding; bioinformatics; epigenetics; transcription factor; systems biology

1. Introduction

GWAS studies have frequently identified variation in non-coding regions as associated with a variety of complex traits and diseases. However, it remains difficult to assign and validate the functional consequences of these variants or to suggest a mechanism by which they actually influence an outcome. These challenges become even more difficult to address at scale, or in high-throughput, when a clear biological candidate molecule and hypothesis cannot be readily tested through experimentation. The most commonly considered mechanisms for altered function are

[†] all workshop co-chairs contributed equally, and are listed alphabetically

altered regulatory activity of an effector molecule (including eQTL, transcription factor binding, enhancers/insulators, epigenetic marks, etc.), alternative splicing, changes to chromosome conformation, or altered biology of non-coding RNA genes. 2017 saw the public release and extension of multiple major data resources exploring biological and biochemical functions of non-coding regions at genome scale, including across multiple tissue and cell contexts (e.g., GTEx (1), ENCODE(2)). Emerging genetic engineering and molecular editing technologies have also accelerated, with use of CRISPR expanding beyond hypothesis-driven gene knockouts into targeting of non-coding elements, unbiased tiling assays, and genome-wide screening applications (3).

Advances in computational methods are required to analyze these new data types, identify patterns, integrate across biological scales, and derive biologically and/or clinically useful insights during primary analyses, by secondary exploration of data in publicly-available resources, and/or by integrating across data sets and using new models. In addition to targeted biomedical questions, there also arises a unique opportunity for computational biologists to identify network and systems properties of non-coding DNA, linking evidence from these assays, genetics, and evolutionary biology with other datasets(4).

2. Speakers and abstracts

From genetics to therapeutics: uncovering and manipulating the circuitry of non-coding disease variants

Manolis Kellis, *Professor, MIT Computer Science and Artificial Intelligence Lab*
Institute Member, Broad Institute of MIT and Harvard

Perhaps the greatest surprise of human genome-wide association studies (GWAS) is that 90% of disease-associated regions do not affect proteins directly, but instead lie in non-coding regions with putative gene-regulatory roles. This has increased the urgency of understanding the non-coding genome, as a key component of understanding human disease. To address this challenge, we generated maps of genomic control elements across 127 primary human tissues and cell types, and tissue-specific regulatory networks linking these elements to their target genes and their regulators. We have used these maps and circuits to understand how human genetic variation contributes to disease and cancer, providing an unbiased view of disease genetics and sometimes re-shaping our understanding of common disorders. For example, we find evidence that genetic variants contributing to Alzheimer's disease act primarily through immune processes, rather than neuronal processes. We also find that the strongest genetic association with obesity acts via a master switch controlling energy storage vs. energy dissipation in our adipocytes, rather than through the control of appetite in the brain. We also combine genetic information with regulatory annotations and epigenomic variation across patients and healthy controls to discover new disease genes and regions with roles in Alzheimer's disease, heart disease, prostate cancer, and to understand their pleiotropic effects by integration with electronic health records. Lastly, we develop systematic technologies for systematically manipulating these circuits by high-throughput reporter assays, genome editing, and gene targeting in human cells and in mice, demonstrating tissue-autonomous therapeutic avenues in Alzheimer's disease, obesity, and cancer. These results provide a roadmap for translating genetic findings into mechanistic insights and ultimately therapeutic treatments for complex disease.

Modeling methyl-sensitive transcription factor motifs with an expanded epigenetic alphabet

Michael M. Hoffman, *Principal investigator, Princess Margaret Cancer Centre & Assistant Professor, Departments of Medical Biophysics and Computer Science, University of Toronto*

Introduction: Many transcription factors (TFs) initiate transcription only in specific sequence contexts, providing the means for sequence specificity of transcriptional control. A four-letter DNA alphabet only partially describes the possible diversity of nucleobases a TF might encounter. Cytosine is often present in the modified forms: 5-methylcytosine (5mC) or 5-hydroxymethylcytosine (5hmC). TFs have been shown to distinguish unmodified from modified bases. Recent chemical probing and sequencing methods provide the opportunity to assess a variety of DNA modifications. Modification-sensitive TFs provide a mechanism by which widespread changes in DNA methylation and hydroxymethylation can dramatically shift active gene expression.

Methods: To understand the effect of modified nucleobases on gene regulation, we developed methods to discover motifs and identify TF binding sites in DNA with covalent modifications. Our models expand the standard A/C/G/T alphabet, adding m (5mC), h (5hmC) and other symbols—permitting computational representations of modified sequence. We also enhanced parts of the MEME Suite and RSAT to handle custom alphabets, expanding the position weight matrix (PWM) formulation of TF binding affinity and enabling clustering of modified PWMs.

Results: We created an expanded-alphabet sequence using whole-genome maps of 5mC and 5hmC in mouse naive T cells and human K562 cells. Using this sequence and ChIP-seq data from ENCODE and others, we identified modification-sensitive *cis*-regulatory modules. We reproduced known binding preferences, including the preference of ZFP57 for methylated motifs and the preference of c-Myc for unmethylated motifs. We have made several novel predictions, and are validating them using ChIP-BS-seq and CUT&RUN. (5)

Quantifying the impact of non-coding mutations on transcriptional regulation

Raluca Gordân, *Assistant Professor, Biostatistics and Bioinformatics, Duke University*

Most disease-associated genetic variants occur in non-coding regions where they can alter gene regulation, rather than gene sequence. Focusing on putative regulatory variants that can affect transcription factor (TF) binding to the genome, I will present new methods for quantifying the change in TF binding due to binding site variants, as well as the statistical significance of the predicted change. Briefly, using as input high-throughput in vitro data for hundreds of mammalian TFs, we developed regression models of TF-DNA binding that implicitly take into account the quality of the training data. Thus, in the case of low-quality data that leads to a large variance in the estimated model parameters, only large changes in TF binding will reach statistical significance; in contrast, high-quality training data sets allow us to identify even subtle changes in TF binding due to genetic variants. To assess the quality of our predictions, we leverage high-throughput enhancer assay data where all possible single base-pair mutations in specific regulatory regions have been tested directly for their effect on gene expression. We find that our TF binding models can explain about ~50% of the variation in gene expression. We are currently using the TF binding change predictions in collaborative GWAS studies to prioritize non-coding variants for further computational and experimental analyses.

CRISPR-SURF: Discovering regulatory elements by deconvolution of CRISPR tiling screen data

Luca Pinello, *Principal Investigator and Assistant Professor, Massachusetts General Hospital & Harvard Medical School*

Tiling screens using CRISPR-Cas technologies provide a powerful approach to map regulatory elements to phenotypes of interest, but computational methods that effectively model these experimental approaches for different CRISPR technologies are not readily available. Here we present CRISPR-SURF, a deconvolution framework to identify functional regulatory regions in the genome from data generated by CRISPR-Cas nuclease, CRISPR interference (CRISPRi), or CRISPR activation (CRISPRa) tiling screens. We validated CRISPR-SURF on previously published and new data, identifying both experimentally validated and new potential regulatory elements. With CRISPR tiling screens now being increasingly used to elucidate the regulatory architecture of the non-coding genome, CRISPR-SURF provides a generalizable and accessible solution for the discovery of regulatory elements. (6)

Delineation and annotation of the human regulatory landscape across 400+ cell types and states

Wouter Meuleman, *Investigator, Altius Institute for Biomedical Sciences*

The human genome encodes vast numbers of non-coding elements whose combined actuation patterns reflect regulatory processes across cellular states and conditions. Despite large-scale technology development for interrogating non-coding parts of the genome, pragmatic annotated high-resolution maps of regulatory regions and their inter-cell type dynamics have been lacking. To address this issue, we applied a joint experimental and computational approach, integrating 733 deeply sequenced DNase I hypersensitivity assays spanning more than 400 distinct human cell types and states. These data enable a systematic and principled approach to studying regulatory architecture and dynamics on a global scale. We define a common coordinate system for regulatory DNA marked by DNase I hypersensitive sites, encompassing over 3 million elements defined and annotated with unprecedented resolution and detail. Through systematic analysis of the dynamics of these regulatory regions across cell types and states, we derive a collection of Regulatory Components, providing a novel multi-component annotation of the human regulome. Using admixtures of multiple components, we show that it is possible to decompose biological features of cell and tissue samples and define the extent to which individual regulatory elements contribute to broader cellular regulatory programs. These previously unappreciated features allow us to characterize the functional properties of genes and pathways. For instance, based solely on their regulatory landscape, we readily identify genes coding for lineage-specifying factors. Moreover, we associate specific regulatory structures with distinct binding site motifs, as well as with gene expression patterns across cell types. Moreover, our Regulatory Components provide a fundamentally new framework for understanding how disease-associated variation maps to genome function, not otherwise appreciated. Taken together, through integrative analysis across hundreds of cell types and states, we provide a novel multi-component annotation of the human regulatory landscape. Our Regulatory Components are predictive for functional and regulatory characteristics

of genes, pathways and genetic variants. As such, they open up new horizons on the architecture of human genome regulation and function.

Genetically explainable non-coding RNA expression by SNVs

Lana Garmire, *Associate Professor, Molecular Biosciences and Bioengineering, U of Michigan*

Long intergenic non-coding RNAs have been shown to play important roles in cancer. However, because lincRNAs are a relatively new class of RNAs compared to protein-coding mRNAs, the mutational landscape of lincRNAs and the impact of mutations on lincRNA expression are not extensively studied. We comprehensively characterize expressed somatic nucleotide variants within lincRNAs using 6118 primary tumor samples from 12 cancer RNA-Seq datasets in TCGA. Due to uncertainty of somatic or germline mutations from analyzing un-paired RNA-Seq data alone, we first build a highly accurate machine-learning model (AUC 0.987) to discriminate somatic variants from germline variants within lincRNAs, using a subset of samples that have both exome-seq and RNA-seq data. We use this model to predict highly confident eSNVs (expressed SNVs) and found that they are especially enriched in chr2p11.2, chr14q32.33, chr22q11.22 and chr3q29 regions. To understand the effect of molecular features on lincRNA somatic eSNVs, we build another model (AUC 0.72) and identify molecular features that are strongly associated with lincRNA mutations, including copy number variation, conservation, substitution type and histone marker features. Finally, we prioritize the lincRNAs by their eSNV influence, and propose a short list of genetically affected lincRNAs to be validated by experimental studies.

Interpreting genetic variants by gene regulatory network

Yong Wang, *Professor, Institute of Applied Mathematics, Academy of Mathematics and Systems Science & National Ctr for Mathematics and Interdisciplinary Science, Chinese Academy of Science*

Interpreting genetic variance (including SNP and structural variants) is the key to precision health. Most of these variants will affect disease risk, response to drugs or other traits such as height in a tissue or condition-specific way. How can we figure out which variants affect the function and regulation of genes in which condition? We propose to use gene regulatory network to integrating omics data and interpret genetic variants. Particularly, we will discuss the models and algorithms to organize, analyze, model, and integrate the genetic variant, DNA accessibility data, transcriptional data, and functional genomic regions together. We believe that the integrative paradigm on chromatin and expression levels will eventually help us to understand the information flow in cell and will influence research directions across many fields.

References

1. A. Battle, CD Brown, BE Engelhardt, SB Montgomery, *Nature.* **550**(7675), 204-13 (2017).
2. CA Sloan, ET Chan, JM Davidson, et al., *Nucleic Acids Research.* **44**(D1), D726-32 (2016).
3. JB Wright, NE Sanjaya, *Trends in Genetics.* **32**(9), 526-9 (2016)
4. M Amorim, S Salta, R Henrique, C Jeronimo *J Transl Med.* **14**, 265 (2016).
5. C Viner, J Johnson, N Walker, et al., *bioRxiv (preprint).* doi.org/10.1101/043794 (2016)
6. JY Hsu, CP Fulco, MA Cole, et al., *bioRxiv (preprint).* doi.org/10.1101/345850, (2018)

PSB 2019 Workshop on Text Mining and Visualization for Precision Medicine

Graciela Gonzalez-Hernandez[1†], Zhiyong Lu[2†], Robert Leaman[2†], Davy Weissenbacher[1], Mary Regina Boland[1,4], Yong Chen[1], Jingcheng Du[5], Juliane Fluck[6,7,14], Casey S. Greene[8,9], John Holmes[1], Aditya Kashyap[10], Rikke Linnemann Nielsen[12], Zhengqing Ouyang[13], Sebastian Schaaf[7], Jaclyn N. Taroni[8,9], Cui Tao[5], Yuping Zhang[14], Hongfang Liu[3]

[1]*Department of Biostatistics, Epidemiology and Informatics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA*

[2]*National Center for Biotechnology Information (NCBI), National Library of Medicine (NLM), National Institutes of Health (NIH), 8600 Rockville Pike, Bethesda, MD 20894, USA*

[3]*Division of Biomedical Statistics and Informatics, Mayo Clinic College of Medicine, Rochester, MN, USA*

[4]*Department of Biomedical and Health Informatics, Children's Hospital of Philadelphia, Philadelphia, PA, USA*

[5] *School of Biomedical Informatics, University of Texas Health Science Center, Houston, TX, USA*

[6] *ZB MED Information Centre for Life Sciences, Bonn, Germany*

[7] *Department of Bioinformatics, Fraunhofer Institute for Scientific Computing and Algorithms (SCAI), Sankt Augustin, Germany*

[8] *Department of Systems Pharmacology and Translational Therapeutics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA*

[9] *Childhood Cancer Data Lab, Alex's Lemonade Stand Foundation, Philadelphia, PA, USA*

[10] *Data Science Masters Program, University of Pennsylvania, Philadelphia, PA, USA*

[11] *Department of Bio and Health Informatics, Technical University of Denmark, Lyngby, Denmark*

[11] *The Jackson Laboratory for Genomic Medicine, Farmington, CT, USA*

[13] *Department of Statistics, University of Connecticut, Storrs, CT, USA*

[14] *Institute of Geodesy and Geoinformation, University of Bonn, Bonn, Germany*

Precision medicine, an approach for disease treatment and prevention that considers "individual variability in genes, environment, and lifestyle" [1] was endorsed by the National Institutes of Health, aided by the presidential Precision Medicine Initiative (PMI), in 2016. PMI provided funding for cancer research and for building a national cohort of one million or more U.S. participants, now known as the "All of Us" Research Program, which aims to expand its impact to all diseases. PMI was the catalyst to a widespread effort around precision medicine, as evidenced by the more than 1000 grants funded by different NIH institutes in just the last two years. The data being generated by these efforts is growing exponentially, and becomes both the greatest treasure and the greatest challenge for researchers. This workshop is a continuation of a similar session in PSB 2018, providing a forum for researchers with strong background in text mining or natural language processing (NLP) and/or machine learning (ML) who are actively collaborating with bench scientists and clinicians to tackle the challenges brought about by this explosion of data.

† Work partially supported by the National Library of Medicine of the National Institutes of Health (NIH) under grant number R01LM011176 (GGH) and its Intramural Research Program (ZL and RL). The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

1. Introduction

According to the National Research Council, "personalized medicine" is an older term with a meaning similar to "precision medicine." However, "personalized" could be thought to imply that treatments and preventions are being developed for each individual; in contrast to what is really intended, which is identifying which approaches will be effective for which group of patients based on shared or similar genetic, environmental, and lifestyle factors. Thus, the preferred term for the presidential initiative launched in 2015 was "precision medicine" rather than "personalized medicine", heralding the switch to the later. The Precision Medicine Initiative (PMI) working group report[2] outlines the goals of precision medicine, "to redefine our understanding of disease onset and progression, treatment response, and health outcome", suggests the means to accomplish this, "more precise measurement of molecular, environmental, and behavioral factors that contribute to health and disease", and the expected outcomes "more accurate diagnoses, more rational disease prevention strategies, better treatment selection, and the development of novel therapies". However, in order to go from the means to the outcomes, one must deal with the onslaught of data that those "more precise" measurements entail.

Big data in health is both a blessing and a curse. It is enabling, promising, but has been the largest roadblock to true progress in precision medicine, as much of key information remains hidden in descriptive text or in patterns that are only obvious after cleverly feeding massive amounts of the right data to machine learning algorithms. Selecting, integrating, and analyzing the right data from medical records (EMRs), standardized clinical data (such as what is required by Medicare), administrative data –from hospitals, insurance companies, and pharmacies-, patient surveys and self-reports in social media or health forums, or via wearable sensors, the published literature, clinical trials, and research data deposited in public collections such GenBank or the Gene Expression Omnibus (GEO) database, and many curated databases of interactions and pathways, to name just a few, is one of the major challenges to precision medicine.

Big data and the advance of machine learning, especially deep learning, has led to an explosion of the application of machine learning techniques in precision medicine. For example, deep learning algorithms have been able to diagnose of pneumonia on chest x-ray images[3], apply for personalized risk stratification based on clinical data[4], and detect spread of breast cancer into lymph node tissue on microscopic specimen images[5]. However, there is no silver bullet. The majority of such studies have not been conducted with scientific rigor regarding data reproducibility and model validity/portability in real-world scenario, and are thus limited to the framework and data used for the study itself.

We have also seen significant advances in NLP methods that have enabled unstructured data to be used for decision support systems and predictive algorithms, given that such data was found exclusively in unstructured form, as recent studies comparing text-mining results with curated databases showed[6-8]. Barriers to progress include ambiguity in the data itself, as variant names in the papers are written irregularly and hard to be grounded and even recognized[9,10], as well as lack of trust and standard validation approaches. For example, whereas there is almost universal acceptance of ICD based cohort selection, NLP does not enjoy the same level of trust, and inclusion

of a patient record in a study based solely on NLP based selection will be frowned upon unless it is followed by manual annotation.

This workshop highlights original research and invited presentations on novel text mining, natural language processing (NLP), and visual analytics approaches at the intersection of lifestyle, environment, and genetics that enable further understanding of disease processes and effective treatment for individuals and cohorts that share specific characteristics.

2. Workshop Summary

The workshop includes a keynote talks by Christopher Chute, plus 6 oral presentations by authors of abstracts submitted for competitive review and selected for presentation based on their innovation and significance. In addition, the workshop closes with presentations by a panel of experts, focusing on 'Current Challenges in Incorporating Genomic, Clinical, Published, and User-generated Data for Precision Medicine', which gives attendees a view of state of the art approaches and roadblocks to the advancement of text mining and machine learning methods that will enable the next big breakthrough in this area.

2.1 *Keynote: Comparability and Consistency of NLP for Biomedical Discovery and Translation*

The keynote talk is given by Dr. Christopher Chute, the Bloomberg Distinguished Professor of Health Informatics, Professor of Medicine, Public Health, and Nursing at Johns Hopkins University, and Chief Research Information Officer for Johns Hopkins Medicine. He received his undergraduate and medical training at Brown University, internal medicine residency at Dartmouth, and doctoral training in Epidemiology and Biostatistics at Harvard. He is Board Certified in Internal Medicine and Clinical Informatics, and an elected Fellow of the American College of Physicians, the American College of Epidemiology, HL7, and the American College of Medical Informatics (ACMI), as well as a Founding Fellow of International Academy of Health Sciences Informatics; he is currently president of ACMI through 2018.

Dr Chute's career has focused on how we can represent clinical information to support analyses and inferencing, including comparative effectiveness analyses, decision support, best evidence discovery, and translational research. He has had a deep interest in semantic consistency, harmonized information models, and ontology. His current research focuses on translating basic science information to clinical practice, and how we classify dysfunctional phenotypes (disease). He became founding Chair of Biomedical Informatics at Mayo Clinic in 1988, retiring from Mayo in 2014, where he remains an emeritus Professor of Biomedical Informatics. He is presently PI on a spectrum of high-profile informatics grants from NIH spanning translational science. He has been active on many HIT standards efforts and chaired ISO Technical Committee 215 on Health Informatics and the World Health Organization (WHO) International Classification of Disease Revision (ICD-11).

2.2 Oral Presentations

In *Development and Validation of the PEPPER Framework (Prenatal Exposure PubMed ParsER) with Applications to Food Additives,* **Mary Regina Boland, Aditya Kashyap, Jiadi Xiong, John**

Holmes, and Scott Lorch, note that although environmental factors contribute to 36% of child deaths worldwide, no comprehensive list of all prenatal environmental exposures exists. They present a method called PEPPER: Prenatal Exposure Pubmed ParsER that utilizes all full-text research articles from Pubmed Central to learn the 'state-of-the-field'. They found that of 31,764 prenatal exposure studies, only 53.0% were methodology studies. When PEPPER is coupled with the FDA's food additive database (called EAFUS), PEPPER is able to capture 56.4% of the studied exposures. Prenatal exposure effects of food additives were studied for 176 compounds out of 3,968 (4.4%) compounds contained in EAFUS. Of 16,832 prenatal exposure methodology studies, only 1,886 (11.2%) investigate food additive effects. In total, 3,117 studies investigated prenatal exposure to food additives. The majority of these were methodology studies (60.5%), followed by non-methodology studies (27.2%), PDF only (8.9%) and systematic reviews (3.4%). Prenatal exposure to commonly used food additives (EAFUS category ASP) are rarely studied with a rate of only 0.24% of methodology studies. Surprisingly, there is also a paucity of research on the effects of banned food additives on prenatal development. Of 2,105 research articles investigating banned food additives, only four (0.19%) investigate effects during the prenatal period and only three (0.14%) were methodology studies.

Jingcheng Du, Yang Xiang, Jing Huang, Xinyuan Zhang, Rui Duan, Jiayi Tong, Jiang Bian, Sahiti Myneni, Yong Chen, and Cui Tao, in *Mining HPV Vaccination Health Beliefs from Twitter Using Deep Learning: A Longitudinal Analysis of Four-Year Data (2014 - 2017),* focus on understanding the public perceptions of vaccines as it is the first step towards developing effective vaccine promotion strategies to fight against the increase of vaccine refusal and delay observed in the last two decades. Traditional surveying methods suffer significant limitations on accessing large-scale public perceptions. The popularity of social media opens a new dimension. However, most of the studies were focusing on analyzing the frequency rather than contents of social media postings. The accurate understanding of the contents in the perspective of grounded behavior change theories is fundamental for the design of precise and targeted vaccination promotion strategies. According to the authors, their study is the first effort to map Twitter vaccination discussion to the grounded behavior change theory - Health Belief Model. They propose and evaluate a deep learning model and apply the model to automatically and accurately extract vaccination health belief from large-scale Twitter data. The deep learning model shows superiority over machine learning baseline model. They identify manifestation of health belief constructs in Twitter corpus of vaccine discussions in a four-year Twitter dataset.

In *Data integration for prediction of time to insulin in type 2 diabetes patients,* the subject of **Rikke Linnemann Nielsen, Louise Donnelly, Agnes Martine Nielsen, Kaixin Zhou, Bjarne Ersboll, Ewan Pearson, and Ramneek Gupta** present Type an approach to predicting risk of a fast or slow disease progression, which varies between individuals. This variation is captured in electronic medical records of T2D patients and identification of biomarkers that are predictive of diabetes progression can possibly reveal relevant patient subgroups characteristics that may assist clinical decisions in T2D treatment management. In their study they analyze electronic medical records from a cohort-based population in Tayside, UK registered from 1994 to 2010 using machine learning approaches. They investigate if integration of life-style data, anthropometry, biochemical data, drug-prescription data and genetic variants could predict slow and fast progression based on

classification of time to insulin (TTI) in T2D patients using random forest and artificial neural network models. TTI is defined as the first day of insulin treatment or as the clinical need for insulin (HbA1c >8.5% treated with two or more non-insulin diabetes therapies) since the day diagnosis was confirmed by HbA1c. Prediction targets is TTI within year 1, 3 or 5 since time of diagnosis. The best performing ANN models with all data except genetics most accurately predicts T2D patients with fast progression. The authors also discuss inclusion of genetic variants in the machine learning models as well as further longitudinal work with the phenotype.

In neurodegeneration, knowledge on etiologies and underlying mechanisms is still sparse, resulting in late diagnosis and a lack of effective therapies. Until longitudinal studies deliver sufficient data, mining and integrating complementary clinical routine data appears promising. In *Longitudinal visualization of heterogeneous data from neurodegenerative patients for clinical hypothesis generation,* **Sebastian Schaaf, Mischa Uebachs, Vyara Tonkova, Kilian Krockauer, Lisa Langnickel, Philipp Koppen and Juliane Fluck** identify a variety of data sources and create an extraction strategy involving text mining, collecting diagnoses, cognitive test scores, biomarker lab measurements as well as medications. The integration into their longitudinal clinical data model allows a semantic access to normalized data from both routine and study contexts, using standards like FHIR, OMOP and adequate public terminologies. Besides programmatic access, they set up an interactive visualization interface, providing views on aggregated data for exploratory settings, but also a custom longitudinal patient viewer, depicting events and measurements for individuals on a timeline. Beyond supporting principal data exchange and review, they regard the recent developments to be crucial for efficient hypotheses generation, stratification and recruitment.

In *MultiPLIER: a transfer learning framework reveals systemic features of rare autoimmune disease,* **Jaclyn Taroni, Peter Grayson, Qiwen Hu, Sean Eddy, Matthias Kretzler, Peter Merkel, and Casey Greene** present a feature-representation-transfer approach, MultiPLIER, which consists of training Pathway Level Information ExtractoR (PLIER) models on large compendia comprised of multiple experiments, tissues, and biological conditions and transferring this information to small rare disease datasets. They demonstrate that MultiPLIER better describes biological processes related to more active or severe disease in a rare autoimmune disorder than models trained on individual datasets.

Yuping Zhang, Zhengqing Ouyang, and Hongyu Zhao in *A statistical framework for data integration through graphical models with application to cancer genomics,* building on a previous study[11], present the problem of discovering regulatory relationships among heterogeneous genomic variables from biological conditions with potentially shared regulatory mechanisms. The genomic variables can be genetic variants, epigenetic states, and gene expression profiles, etc. The heterogeneous genomic variable types may be binary, categorical, or continuous. The biological conditions can be different tissue types or disease types, etc. They may have both shared and tissue- or disease-specific regulations. The authors develop a new general network estimation framework, named DIG, to jointly learn conditional independence among a set of heterogeneous types of variables across a set of distinct but related conditions. They illustrate the method by integrating mutations and copy number variations, and apply it to COAD and BRCA using TCGA data. Their study identify both common and distinct network modules in COAD and BRCA, which shows that the modules are biologically meaningful.

References

1. Collins FS, Varmus H. A New Initiative on Precision Medicine. *N Engl J Med*. 2015;372(9):793-795. doi:10.1056/NEJMp1500523.
2. https://www.nih.gov/sites/default/files/research-training/initiatives/pmi/pmi-working-group-report-20150917-2.pdf Accessed October 1, 2018.
3. Rajpurkar P, Irvin J, Zhu K, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv:171105225 2017.18
4. Rajkomar A, Oren E, Chen K, et al. Scalable and accurate deep learning with electronic health records. npj Digital Medicine 2018;1:18.
5. Babak Ehteshami Bejnordi, MS1; Mitko Veta, PhD2; Paul Johannes van Diest, MD, PhD3; et al, Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer, JAMA. 2017;318(22):2199-2210. doi:10.1001/jama.2017.14585
6. Allot et al. LitVar: a semantic search engine for linking genomic variant data in PubMed and PMC *Nucleic Acids Research*, 2018
7. Singhal et al. Text Mining Genotype-Phenotype Relationships from Biomedical Literature for Database Curation and Precision Medicine. *PLoS Comput Biol*, 2016
8. Lee et al. Scaling up data curation using deep learning: An application to literature triage in genomic variation resources *PLoS Comp Biol*, 2018
9. Wei et al. tmVar 2.0: Integrating genomic variant information from literature with dbSNP and ClinVar for precision medicine *Bioinformatics*, 2017.
10 Wei et al tmVar: a text mining approach for extracting sequence variants in biomedical literature. *Bioinformatics*, 2013.
11. Zhang, Y., Ouyang, Z. and Zhao, H., 2017. A statistical framework for data integration through graphical models with application to cancer genomics. *The Annals of Applied Statistics*, 11(1), pp.161-184.

Translational informatics of population health: How large biomolecular and clinical datasets unite

Yves A. Lussier, MD
Center for Biomedical Informatics & Biostatistics, Department of Medicine, BIO5 Institute, Cancer Center, The University of Arizona
1230 North Cherry Avenue, Tucson, AZ 85721
Yves@email.arizona.edu

Atul J. Butte, MD, PhD
Bakar Computational Health Sciences Institute, University of California, San Francisco
550 16th Street, San Francisco, CA 94158
Atul.Butte@ucsf.edu

Haiquan Li, PhD
College of Agriculture and Life Sciences, The University of Arizona
1177 E 4th Street, Shantz 509, Tucson, AZ 85719
Haiquan@email.arizona.edu

Rong Chen, PhD
Sema4 Genomics
333 Ludlow Street, Stamford, CT 06902
Rong.Chen@seme4Genomics.com

Jason H. Moore, PhD
The Perelman School of Medicine, University of Pennsylvania
D202 Richards, 3700 Hamilton Walk, Philadelphia, PA 19104
Jhmoore@upenn.edu

This paper summarizes the workshop content on how the integration of large biomolecular and clinical datasets can enhance the field of population health via translational informatics. Large volumes of data present diverse challenges for existing informatics technology, in terms of computational efficiency, modeling effectiveness, statistical computing, discovery algorithms, and heterogeneous data integration. While accumulating large 'omics measurements on subjects linked with their electronic record remains a challenge, this workshop focuses on non-trivial linkages between large clinical and biomolecular datasets. For example, exposures and clinical datasets can relate through zip codes, while comorbidities and shared molecular mechanisms can relate diseases. Workshop presenters will discuss various methods developed in their respective labs/organizations to overcome the difficulties of combining together such large complex datasets and knowledge to enable the translation to clinical practice for improving health outcomes.

Keywords: Translational informatics, biomolecular, clinical, population health, big data, workshop

1. Introduction, Background, and Motivation

The field of population health is rapidly moving to the forefront of research, with the advancement of biotechnologies and growth of international collaborations enabling the vast

accumulation of population health data. The availability of such data crossing multiple dimensions, from electronic health records, lifestyles, environmental factors, genetics, to genomics, is promising for further advancing the field via translational bioinformatics. A growing trend is the integrative data collection that encompasses all aspects (both genetic and non-genetic factors) of the same participants, exemplified by eMERGE[1], UK Biobank[2], and All of US program[3], among many others in specific domain and specialties.[4,5]

However, large volumes of data present diverse challenges for existing informatics technology, in terms of computational efficiency, modeling effectiveness, statistical computing, discovery algorithms, and heterogeneous data integration. These new demands also call for bridging the gap between disciplines among statistical genetics, health informatics, and bioinformatics. Successful endeavors in these areas will dramatically enhance the understanding of the genetic/epigenetic mechanisms of complex diseases and their interplay with the environment and lifestyles as well as foster the translation of these findings to clinical practice to improve health outcomes.[6,7]

In this era of Big Data science, the number of opportunities to study large-scale molecular and population datasets together is flourishing. The developers of PheWAS[8] were among the pioneers to transform heterogeneous and sparsely-annotated clinical data for systematic analysis with densely-annotated SNP arrays. Combining this knowledge with publicly-available data from sources, such as UK Biobank[2] that offers health information on over 500,000 participants, not only promotes Big Data analytics, but also demonstrates the feasibility of such studies.

This paper summarizes how the integration of large datasets, such as biomolecular and clinical data, can advance the field of population health via translational informatics as well as focuses on current approaches to overcome the challenges of combining these complex data.

2. Workshop Presenters

The three-hour workshop is organized in the form of six presentations, including two keynote speakers, followed by a discussion session, which will be moderated by Dr. Yves A. Lussier.

Keynote speakers are:

- Atul Butte, MD, PhD (Priscilla Chan and Mark Zuckerberg Distinguished Professor, University of California, San Francisco)
- Jason H. Moore, PhD, FACMI (Edward Rose Professor of Informatics, University of Pennsylvania)

Additional speakers include:

- Francesca Vitali, PhD (Research Assistant Professor, The University of Arizona)
- Lara M. Mangravite, PhD (President, Sage Bionetworks)
- Serghei Mangul, PhD (QCB Postdoctoral Fellow, University of California, Los Angeles)
- Marina Sirota, PhD (Assistant Professor, University of California, San Francisco)

3. Presenters' Abstracts

Translating a trillion points of data into therapies, diagnostics, and new precision medicine
Atul Butte, MD, PhD (University of California, San Francisco)
There is an urgent need to take what we have learned in our new "genome era" and use it to create a new system of precision medicine, delivering the best preventative or therapeutic intervention at the right time, for the right patients. Dr. Butte's lab at the University of California, San Francisco builds and applies tools that convert trillions of points of molecular, clinical, and epidemiological data -- measured by researchers and clinicians over the past decade and now commonly termed "big data" -- into diagnostics, therapeutics, and new insights into disease. Several of these methods or findings have been spun out into new biotechnology companies. Dr. Butte, a computer scientist and pediatrician, will highlight his lab's recent work, including the use of publicly-available molecular measurements to find new uses for drugs including new therapies for autoimmune diseases and cancer, discovering new druggable targets in disease, the evaluation of patients and populations presenting with whole genomes sequenced, integrating and reusing the clinical and genomic data that result from clinical trials, discovering new diagnostics include blood tests for complications during pregnancy, and how the next generation of biotech companies might even start in your garage.

Enabling translational bioinformatics with accessible artificial intelligence
Jason H. Moore, PhD (University of Pennsylvania)
Artificial intelligence (AI) is a rapidly maturing technology that has the potential to accelerate translational bioinformatics and precision medicine using both basic science and clinical data. While AI has become widespread, many commercial AI systems are not yet accessible to individual researchers nor the general public due to the deep knowledge of the systems required to use them. We believe that AI has matured to the point where it should be an accessible technology for everyone. We present an ongoing project whose goal is to deliver an open-source, user-friendly AI system that is specialized for machine learning analysis of complex data in the biomedical and health care domains.

Novel and emerging data fusion strategies for integrating health and biomolecular data
Francesca Vitali, PhD (The University of Arizona)
Over the last few years, biomedical research and clinical practice have experienced incredible growth in terms of both the amount and variety of data being collected and leveraged for different types of analysis. This represents a great opportunity to increase our knowledge about many biological mechanisms as well as improve the medical process. However, not all big data is created equal, complicating the integration and analysis of such large datasets. For example, clinical record data is highly heterogeneous, sparsely annotated, and contains several measurement types and unstructured text fields comprised of ambiguous statements as well as varying levels of certainty, whereas genomic and imaging data are crisp, homogeneous, densely annotated data with a low cardinality of distinct variables. Nowadays, the development of novel methodologies capable of integrating population health data with biomolecular data is crucial,

not only for enabling translational and clinical research, but for developing more effective patient care. However, integrating these data are particularly challenging when the molecular measurements are not conducted on individual subjects. In order to take full advantage of the wide spectrum of biomedical data available, advanced data integration tools need to be developed. In this context, we will discuss novel and emerging data fusion strategies for integrating health and biomolecular data to develop new research hypotheses and conduct predictive and data interpolation operations. These methods include approaches that (i) take into account comprehensive drug-exposure histories of individuals derived from healthcare data, while also including genetic, environmental, and lifestyle variabilities for each individual; (ii) integrate electronic medical records with biobank data to identify new disease pathways; (iii) combine multi-omic profiling with clinical factors from large cohorts; and (iv) perform crisp integration of bimolecular data whilst leveraging population measurements (e.g., counties, medication, diseases).

Open practices to advance biomedicine through data-intensive science

Lara M Mangravite, PhD (Sage Bionetworks)

Open science practices in bio-computing have been promoted over the past 10 years under the premise that these approaches can improve confidence and, therefore, speed advancement of biomedical hypotheses stemming from computational research. In that time, we have observed wide adoption of open practices including those focused on open data, open commons(es), open source software, and open access publishing. Although many of these efforts help to establish confidence in research observations amongst computationally-savvy researchers, they often fail to support the wider acceptance necessary to inform trajectories of biological inquiry and/or to promote adoption for use in clinical care. Here, we discuss complementary mechanisms to further support the advancement of biocomputational hypotheses, including those developed using emerging digital health technologies, through the transfer and translation of knowledge across research domains.

Seeing Beyond the Target: Constructing germline research cohorts from clinical tumor sequencing

Serghei Mangul, PhD (University of California, Los Angeles)

Tens of thousands of cancer patients have had their tumors sequenced to identify clinically actionable mutations. In addition to saving lives, this activity has produced valuable research data sets leading to significant discoveries in basic and translational domains. However, the targeted nature of clinical tumor sequencing has a limited research scope, especially with respect to germline genetics. In this work, we address this problem by developing a software platform (SBT: Seeing Beyond the Target) that mines discarded tumor sequences to produce rich research level data including genome-wide germline genotypes, T and B cell receptor sequences, rDNA and mtDNA copy number, and HLA types. These features have been demonstrated as potential prognostic indicators in research studies, and our methods now make them available in large-scale clinical cohorts. We validate the accuracy of our tool, by comparison, to deeply sequenced cohorts and show its utility through replication of known genetic associations. We provide a free downloadable cloud implementation and demonstrate its efficiency by constructing the largest

germline-somatic cohort produced to date (n>20,000), more than doubling the size of The Cancer Genome Atlas. We believe that SBT will greatly increase the research potential of clinical tumor data sets and provide a bridge between the germline and somatic research communities. SBT is freely available at https://github.com/smangul1/seeing.beyond.target/wiki

Leveraging population level molecular, environmental and clinical data to study adverse pregnancy outcomes

Marina Sirota, PhD (University of California, San Francisco)

Given the wealth and availability of genomic, clinical and environmental exposure data, computational integrative methods provide a powerful opportunity to identify population-specific determinants of disease. In this talk, I will discuss our efforts to develop computational methods and integrate large-scale genomic, transcriptomic and environmental exposure datasets to elucidate factors that affect preterm birth (PTB). Preterm birth, or the delivery of an infant prior to 37 weeks of gestation, is a major health concern. Infants born prematurely, comprising of about 12% of the US newborns, have elevated risks of neonatal mortality and a wide array of health problems. In our work, we leverage the rich multi-omic, clinical and environmental variation data to advance our understanding of biology of preterm birth as it relates to all populations. Our findings further inform precise population-specific diagnostic and therapeutic strategies bringing us closer to applying precision medicine to this important biomedical problem.

4. Conclusion

This workshop will highlight a number of methods, strategies, and tools currently being developed for integrating population health data with biomolecular data to mitigate the diverse challenges of existing informatics technology. The ability to combine these big data across various domains to conduct meaningful and interpretable analysis is critical for improving overall population health outcomes.

Acknowledgements

We would like to thank Dr. Colleen Kenost for her organizational contributions with this workshop and proceedings paper.

References

1. O. Gottesman, H. Kuivaniemi, et al., *Genetics In Medicine*, 2013, **15**, 761.
2. R. Collins, *The Lancet*, 2012, **379**, 1173-1174.
3. N. I. o. Health, 2018.
4. R. J. Hodes and N. Buckholtz, *Expert Opinion on Therapeutic Targets*, 2016, **20**, 389-391.
5. L. National Heart, and Blood Institute, 2016.
6. H. Li, I. Achour, et al., *Npj Genomic Medicine*, 2016, **1**, 16006.
7. L. Li, W.-Y. Cheng, et al., *Science Translational Medicine*, 2015, **7**, 311ra174-311ra174.
8. J. C. Denny, L. Bastarache, et al., *Nature Biotechnology*, 2013, **31**, 1102.