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

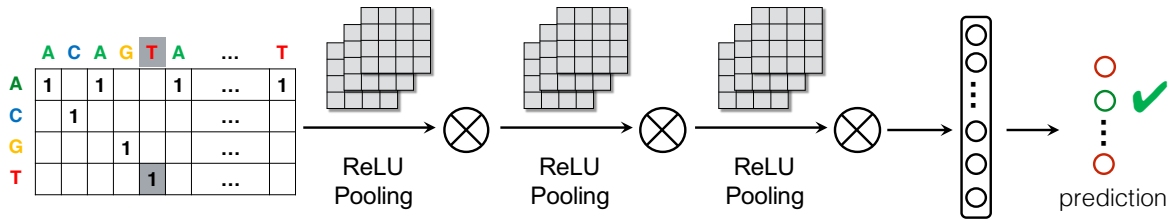

Fig. 1. General CNN models architecture. One-hot-encoded sequence data (left) shown as a $4 \times L$ matrix; ReLU denotes a network unit based on the rectifier function, $(f(x) = \max(0, x))$; the $\otimes$ symbol denotes convolution; the pooling layer selects the stronger signals from previous layer;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

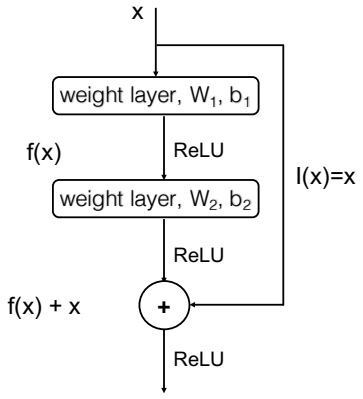

Fig. 2. A building block of Residual Network

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]

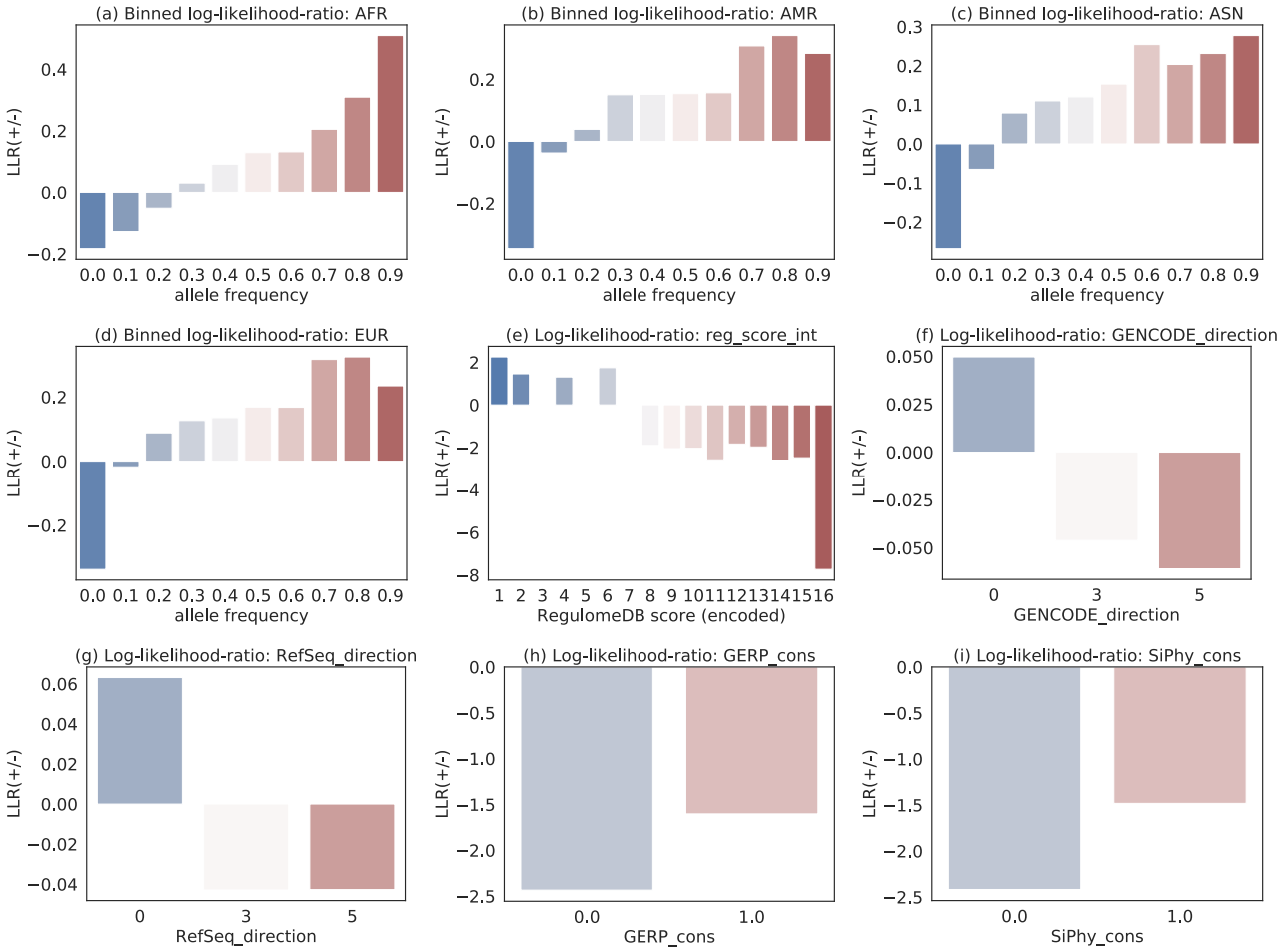

Fig. 3. Estimated log likelihood ratio (LLR) of features. "Direction" means the location of the SNP relative to the nearest gene (0 = within; 3 = downstream, 5 = upstream).

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

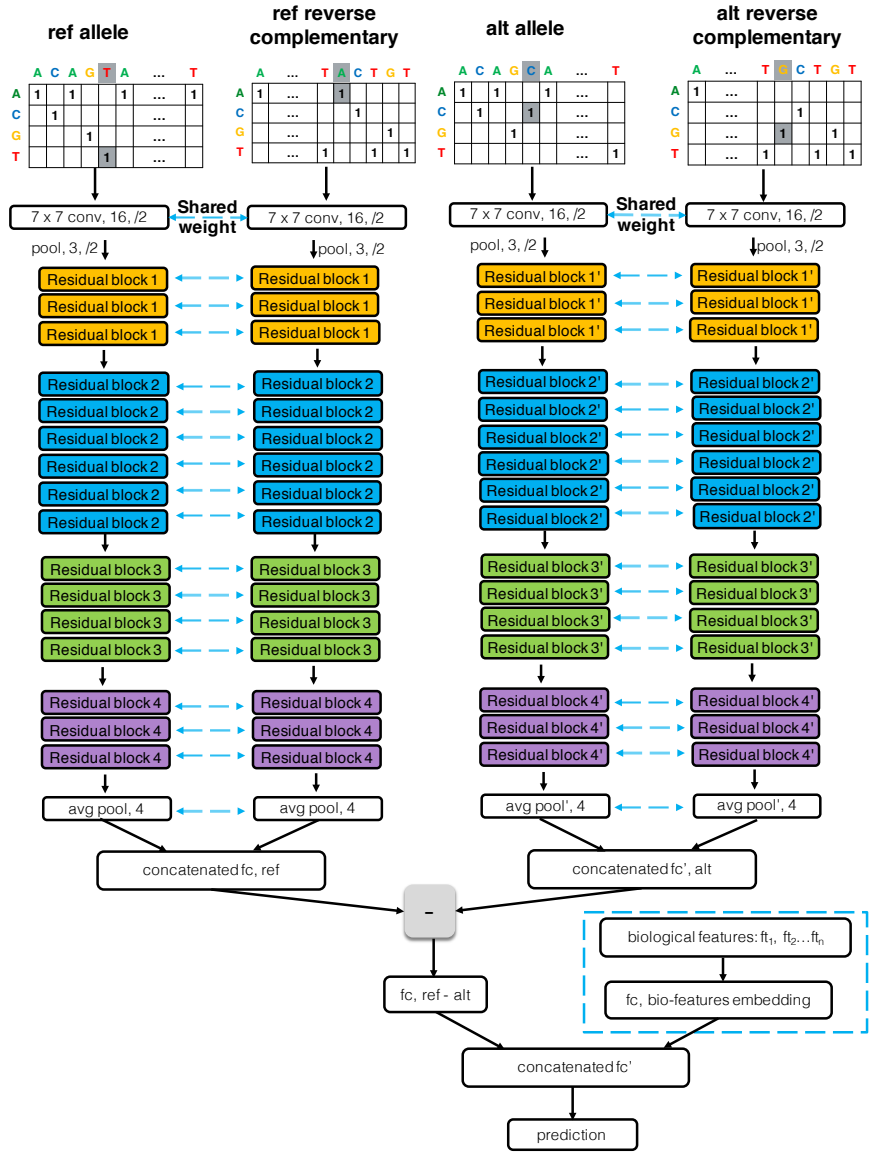

Fig. 4. Architecture of model Res2s2aM. In the Res2s2a model, the portion of the network shown in the dotted rectangle (which is based on SNP annotation data from HaploReg) is not included.

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.

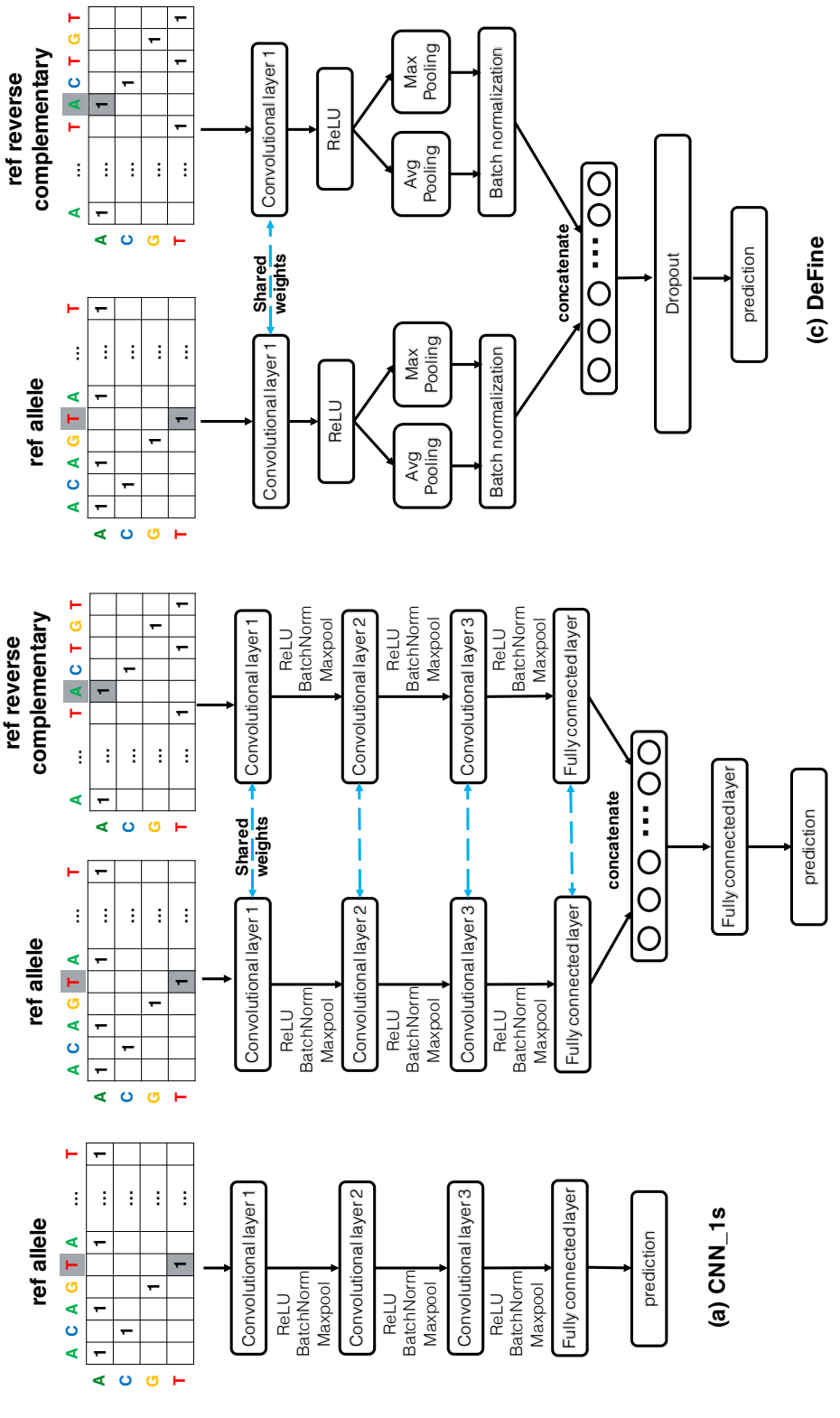

Fig. 5. Deep models to compare with: (a) CNN_1s is the CNN model with "+" strand sequence. (b) CNN_2s is the CNN model with "+" and "-" strands (c) DeFine[11]

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

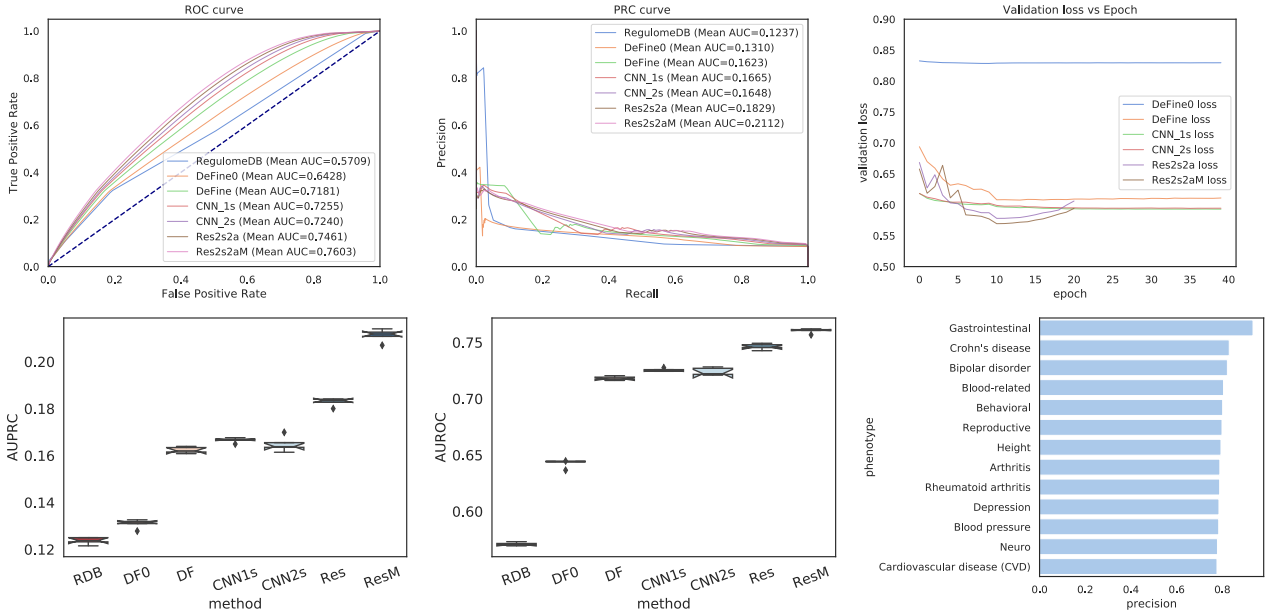

Fig. 6. Performance comparison of seven models: ResM = Res2s2aM , Res = Res2s2a, DF = DeFine, DF0 = DeFine0, CNN2s = CNN_2s, CNN1s = CNN_1s, and RDB = RegulomeDB. Lines, boxes, and marks denote median, interquartile range, and outliers, respectively.

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).