COMPUTATIONAL APPROACHES TO UNDERSTANDING THE EVOLUTION OF MOLECULAR FUNCTION

Yana Bromberg
Department of Biochemistry and Microbiology, Rutgers University
New Brunswick, New Jersey, U.S.A.

Matthew W. Hahn†, Predrag Radivojac
†*Department of Biology, Indiana University*
Department of Computer Science and Informatics, Indiana University
Bloomington, Indiana, U.S.A.

1. Introduction

Understanding the function of biological macromolecules and their interactions is a grand challenge of modern biology, and a key foundation for biomedical research.[1,2] It is now evident that the function of these molecules, in isolation or in groups, can be productively studied in the context of evolution.[3,4] Therefore, understanding how these molecules and their functions evolve is an important step in understanding the specific events that lead to observable changes in molecular and biological processes.

With the advent of high-throughput technologies and the rapid accumulation of molecular data over the past several decades, the evolution of molecular function can be systematically studied at multiple levels. This includes the evolution of protein structure, 3D organization and dynamics, protein and gene expression, as well as the higher-level organization of function contained within pathways.[5–11] New experiments using the latest gene-editing technologies (such as CRISPR-Cas9) have also made it possible to directly test hypotheses about function in almost any organism.[12] Combining these data with theory and computational tools taken from evolutionary biology and related fields has led to an explosion in the study of how function evolves.

2. Overview of Contributions

Our session includes four accepted papers covering a variety of the subjects in this field. The papers address biological questions from metabolic processes to the evolution of duplicated genes; they use computational methods ranging from learning functions on biological networks to the optimal way to choose clustering parameters to identify homologs. Bowerman et al. investigate a set of about one hundred fully sequenced bacterial species mapped onto a space of metabolic variants via a literature search. They subsequently use these data to learn metabolic signatures among these species, an approach that can ultimately lead to a predictive system of metabolic potential for any bacterial species. Cao and Cowen study protein function transfer within a single species and ask under what conditions it leads to accurate prediction. Several sequence, network, and evolutionary features were examined to conclude that the level of sequence divergence is the major determinant of accurate function transfer

among within-species paralogs in yeast. The paper relates to several earlier studies addressing evolutionary relationships and functional similarity.[13–17] Wang et al. present and evaluate a new approach for protein function prediction. Their method is based on amino acid sequences and protein-protein interaction networks over multiple species, integrated into a single heterogeneous network. Network integration is often challenging to formalize considering practical problems such as missing data, sample selection bias, and noise in available protein-protein interactions. Nevertheless, the approach showed good performance upon data integration and provided the insight that the combination of data sources contributed to increased accuracy. Finally, Wiwie and Röttger study the behavior and performance of several clustering algorithms in the context of detecting protein families in similarity graphs. Protein clustering is difficult owing to the unequal sizes of homologous families and the sensitivity of clusters to the parameters of the algorithm. They show that the original data can, in principle, be used to predict clustering performance but also highlight difficulties in finding optimal clustering parameters.

References

1. R. Rentzsch and C. A. Orengo, *Trends Biotechnol* **27**, 210 (2009).
2. P. Radivojac *et al.*, *Nat Methods* **10**, 221 (2013).
3. J. A. Eisen, *Genome Res* **8**, 163 (1998).
4. M. Pellegrini *et al.*, *Proc Natl Acad Sci U S A* **96**, 4285 (1999).
5. N. V. Grishin, *J Struct Biol* **134**, 167 (2001).
6. E. V. Koonin, *Annu Rev Genet* **39**, 309 (2005).
7. D. A. Drummond *et al.*, *Proc Natl Acad Sci U S A* **102**, 14338 (2005).
8. C. Pal *et al.*, *Nat Rev Genet* **7**, 337 (2006).
9. M. E. Peterson *et al.*, *Protein Sci* **18**, 1306 (2009).
10. W. Qian *et al.*, *Proc Natl Acad Sci U S A* **108**, 8725 (2011).
11. C. Park *et al.*, *Proc Natl Acad Sci U S A* **110**, E678 (2013).
12. J. A. Doudna and E. Charpentier, *Science* **346**, p. 1258096 (2014).
13. S. Mika and B. Rost, *PLoS Comput Biol* **2**, p. e79 (2006).
14. R. A. Studer and M. Robinson-Rechavi, *Trends Genet* **25**, 210 (2009).
15. N. L. Nehrt *et al.*, *PLoS Comput Biol* **7**, p. e1002073 (2011).
16. A. M. Altenhoff *et al.*, *PLoS Comput Biol* **8**, p. e1002514 (2012).
17. G. Plata and D. Vitkup, *Nucleic Acids Res* **42**, 2405 (2014).

IDENTIFICATION AND ANALYSIS OF BACTERIAL GENOMIC METABOLIC SIGNATURES

NATHANIEL BOWERMAN
Department of Biology, Hope College, 35 E 12th St,
Holland, MI 49423 USA
nathaniel.bowerman@hope.edu

NATHAN TINTLE
Department of Mathematics and Statistics, Dordt College, 498 4th Ave NE
Sioux Center, IA 51250, USA
nathan.tintle@dordt.edu

MATTHEW DEJONGH
Department of Computer Science, Hope College, 27 Graves Place,
Holland, MI 49423 USA
dejongh@hope.edu

AARON A. BEST*
Department of Biology, Hope College, 35 E 12th St,
Holland, MI 49423 USA
best@hope.edu

With continued rapid growth in the number and quality of fully sequenced and accurately annotated bacterial genomes, we have unprecedented opportunities to understand metabolic diversity. We selected 101 diverse and representative completely sequenced bacteria and implemented a manual curation effort to identify 846 unique metabolic variants present in these bacteria. The presence or absence of these variants act as a metabolic signature for each of the bacteria, which can then be used to understand similarities and differences between and across bacterial groups. We propose a novel and robust method of summarizing metabolic diversity using metabolic signatures and use this method to generate a metabolic tree, clustering metabolically similar organisms. Resulting analysis of the metabolic tree confirms strong associations with well-established biological results along with direct insight into particular metabolic variants which are most predictive of metabolic diversity. The positive results of this manual curation effort and novel method development suggest that future work is needed to further expand the set of bacteria to which this approach is applied and use the resulting tree to test broad questions about metabolic diversity and complexity across the bacterial tree of life.

* To whom correspondence should be addressed.

1. Introduction

The metabolism of an organism relies on thousands of biochemical reactions, which comprise a network that allows the cell to grow, reproduce, and respond to changing environmental conditions. The set of metabolic reactions are defined by the genes the organism carries and dictate the metabolic properties of the organism. Developing an understanding of the metabolic reactions possible by an organism begins to coalesce into a coherent picture of the metabolic capability of the cell. With thousands of annotated genome sequences of microbial organisms available, it is now possible to analyze not only the metabolic properties of individual organisms, but also the patterns that are seen in metabolic networks across organisms. This includes analyses of the evolution of specific metabolic pathways [*e.g.*, 1,2], analyses based on network topology and properties [*e.g.*, 3–6], analyses of simulated metabolic networks [*e.g.*, 7,8], and combinations of flux balance analysis based modeling of metabolic networks within the context of phylogenies [9–11]. Such analyses can lead to a deeper understanding of the metabolic landscape represented by microbial diversity. Further, sequence-based taxonomic surveys and metagenomic analyses of diverse environments are beginning to allow the systematic exploration of relationships between microbial diversity, functional diversity and environment [12–16].

Accurate annotation of sequenced genomes is foundational to downstream analyses of genomes and metagenome communities. We have reviewed [17] the rapid and accurate subsystem approach to genome annotation implemented in the SEED [18] and RAST [19] frameworks. Achieving highly accurate automated annotations of genomes in RAST is predicated upon a core set of manually curated subsystems in which an expert has catalogued the functional elements of a biological process (*e.g.*, a metabolic pathway) and assigned genes to those functional elements for a large set of sequenced microbes. This ensures high quality annotation of each subsystem and the propagation of knowledge captured in the subsystem to all existing and newly sequenced genomes. One outcome of the subsystems approach is the declaration and discovery of metabolic variants, which are defined as different forms or combinations of forms of a functioning metabolic process [17,20,21]. By identifying patterns of genes comprising a variant, one can quickly assign an organism to a particular variant based on the pattern of genes found during the annotation process. Thus, an organism is assigned a variant code for each subsystem, which yields an abstraction of the metabolic capabilities and the forms of those metabolic functions. Further, a catalogue of functional variants that exist for a particular subsystem captures the diversity with which that biological process is performed among sequenced microbes. Such a catalogue represents a rich data set through which we can gain insight into the complexity and diversity of microbial metabolism.

To enable these types of inquiries and to provide consistent descriptions of metabolic variants among sequenced microbes, we selected a representative set of 101 microbial genomes that were used to manually define and annotate metabolic variants in 139 distinct subsystems covering much of known metabolism. We used this resource to (i.) generate a metabolic signature for each of the 101 organisms comprised of assigned variants for each of the 139 subsystems and (ii.) conduct comparative analyses of metabolic signatures of this diverse set of microbes. These variants and their definitions yield a set of high-confidence metabolic subsystems that have been used to aid the automated generation of genome-scale metabolic reconstructions [22], provide a framework

for automated recognition and propagation of variants to newly sequenced genomes, and allow for comparative studies of metabolic variation observed in sequenced microbes.

2. Results

2.1 *Defining Metabolic Variants for Sequenced Bacteria*

A metabolic variant can be described as a particular version of a metabolic process performed by an organism [21]. We will use the synthesis of isoprenoids (terpenoids) to illustrate the concept of metabolic variants and how particular variants are assigned to an organism. Isoprenoids (*e.g.,* chlorophyll and cholesterol) are found in all organisms and are essential to survival. Key isoprenoid precursors, isopentenyl diphosphate (IPP) and dimethylallyl diphosphate (DMAPP) are produced via two known biosynthetic pathways, the so-called mevalonate and non-mevalonate (DOXP) pathways [1]. The reactions in each pathway are catalyzed by non-homologous proteins, and represent two distinct routes to IPP and DMAPP for organisms. In considering the simplest case of defining metabolic variants for this metabolic process, each of these routes represent separate variants – alternative ways to accomplish the same function of producing precursors to isoprenoids. A third variant exists in the case of an organism containing the necessary genes for both of these pathways. A fourth variant indicates absence of this function through known metabolic pathways in an organism. For each variant (defined in this case as A, B, C and -1, respectively), the possible patterns of metabolic steps involved in each variant is generated, and a brief verbal description of the variant is given. Assignment of any one organism to a known variant of the pathway is accomplished by identifying genes in the organism's genome that encode functions corresponding to the area of metabolism and matching the pattern of metabolic steps the organism is predicted to be capable of to one of the defined variants (see Supplemental Figure 1 for additional details).

We have implemented the approach of identifying variants, defining variants, and assigning variants to organisms in the framework of SEED subsystems [18]. This represents a significant, multi-year manual curation effort on the part of SEED annotators through the capture of known metabolic diversity described in the literature and the analysis of patterns seen in sequenced microbial genomes. We chose a set of 101 bacterial genomes, representing 14 bacterial divisions, and 139 subsystems in the SEED that maximized our coverage of metabolism represented in major metabolic databases (*e.g.*, KEGG) and that facilitated the automated generation of metabolic models for bacteria [22]. We characterized a total of 846 metabolic variants in these subsystems that our set of organisms are capable of based on known information of each subsystem and the annotated function of genes in each genome. The outcome of this curation effort is a metabolic variant catalogue comprising descriptions of naturally occurring variations of central and intermediate metabolism for a phylogenetically diverse group of bacteria. Supplemental Figure 2 and Supplemental Files 1-4 give detailed information on the organisms and variants selected and defined.

2.2 *Analyses of Bacterial Metabolic Signatures*

In order to gain a more thorough understanding of metabolic diversity and how metabolic functions are distributed throughout Bacteria, we devised a measure of the metabolic distance, D_{FM}, between two organisms based on the curated metabolic variant catalogue. For a given organism *i*, it is possible to summarize the metabolic capabilities as a binary vector, v_i, of 846 0's and 1's, representing the absence and presence, respectively, of each of the 846 metabolic variants. In effect, *v* is a metabolic signature (or barcode) describing the metabolic capabilities of an organism. D_{FM} measures the metabolic distance between two given organisms *i* and *j* by comparing the similarity of v_i and v_j to the likelihood of observing the similarity between the two vectors by chance. We utilized complete linkage hierarchical clustering of all pairwise D_{FM} of organisms in our dataset to produce a dendrogram summarizing the relationships of the organisms based on metabolic distances (Figure 1). We used a false discovery rate (FDR) of 1×10^{-15} to identify 5 distinct clusters of organisms (Clusters A through E in Figure 1). Each cluster represents a group of organisms with highly similar metabolic signatures. To assess the face validity of the resulting metabolic signature tree, we sought to confirm that the ordering seen in the tree met reasonable biological expectations. For instance, one would expect that closely related organism pairs are likely to be closely paired on the dendrogram – *E. coli* and *Salmonella* are nearest neighbors in the tree as are two representatives of the genus *Shewanella*. Furthermore, the four oxygenic photosynthetic organisms in the set form a tight cluster (FDR $<1 \times 10^{-60}$, Supplemental Figure 3a, organism names colored green). These observations, and many others not detailed here (for example, Supplemental Figures 3b and 3c), indicate that the metabolic distance metric reveals biologically meaningful patterns and gave us confidence that we could use the tree to address additional biological questions of interest.

2.3 *Contribution of Organism Characteristics to Bacterial Metabolic Signatures*

To provide a quantitative estimate of the ability of organism characteristics to explain the clustering observed in the metabolic signature tree, we produced a data set capturing 19 characteristics for each of the 101 organisms, covering attributes such as phylogenetic grouping, environment classification, and oxygen utilization (Supplemental File 1). We performed a multiple regression analysis, using the 19 phenotypic characteristics to predict metabolic distance. The variables in our data set were able to explain 50% of the variance of metabolic distance ($r^2 = 0.50$). The top four characteristics contributing to the clustering are genome size, metabolic mode, host association, and ability to survive in an intracellular environment, uniquely explaining 19.7%, 9.6%, 7.5% and 7.2% of the overall variation in metabolic distance, respectively. All other characteristics contribute to ~5% or less of the overall r^2. Phylogenetic distance ranked 11th of the 19 characteristics, indicating that only a small fraction of the metabolic distance variance could be attributed to phylogeny. A phylogenetic tree of the organisms in this study annotated with metabolic signature cluster membership shows the clear mixing of related organisms throughout the 5 clusters (Supplemental Figure 2). A follow-up analysis which removed 14 organisms with small genomes (Cluster B), showed that there is a slight decrease in the ability to explain the overall variation in metabolic distance ($r^2 = 0.48$) with the 19 phenotypes combined, and less predictive

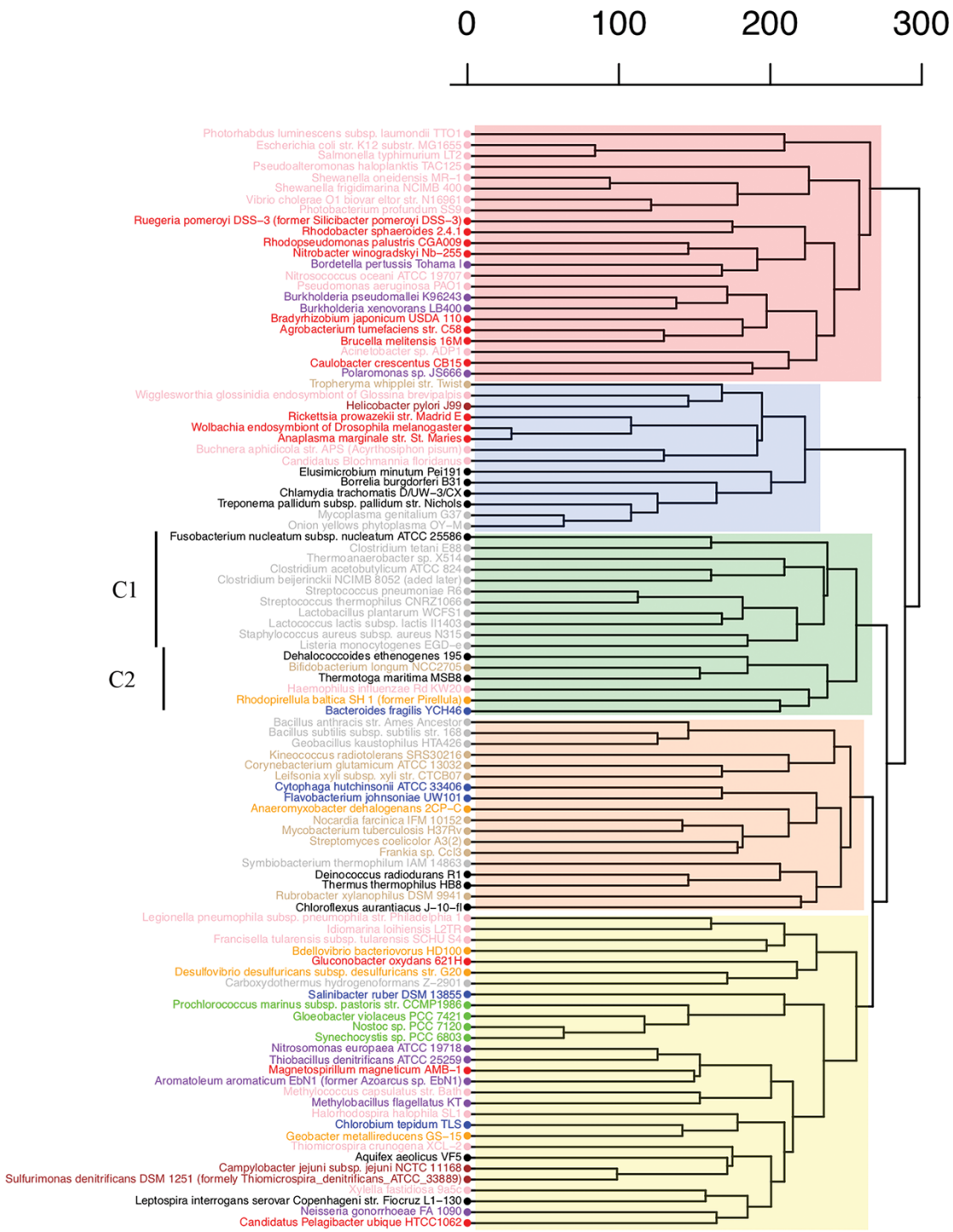

Figure 1. Metabolic Signature Tree from complete linkage hierarchical clustering of D_{FM} of organisms. Five clusters corresponding to an FDR of 1 x 10^{-15} are highlighted by shading – Clusters A-E; pink, blue, green, orange and yellow, respectively. Subclusters C1 and C2 are indicated by black bars. Organism names are colored according to phylogenetic classification: Actionomycetes, Tan; Firmicutes, Gray; Cyanobacteria, Green; Bacteroides/Chloribi, Blue; Other, Black; Proteobacteria: Alpha, Red; Beta, Purple; Delta, Orange; Epsilon, Brown; Gamma, Pink.

ability of genome size (from 19.7% to 6.7%). Full results are provided in Supplemental Figures 4a and 4b.

2.4 *Specific Phenotypes Associated with Individual Clusters*

In addition to characterizing the influence of phenotype on the global topology of the metabolic tree, it is possible to associate specific phenotypic characteristics with individual clusters of organisms in the metabolic tree. We assessed the distribution of each phenotypic characteristic within a cluster and compared this to the distribution of that phenotypic character in the other clusters to yield a statistical measure of the differential distribution of any one phenotypic trait among clusters (see Methods). Each of the clusters is characterized by a particular set of phenotypes as summarized in Table 1 that are over or underrepresented at a conservative measure of statistical confidence ($p < 0.0006$). As expected, the phenotypic characters with the lowest p-values for each cluster correspond to initial observations seen with the overlay of phenotypic characters on the metabolic tree, while providing more specificity to the observations and highlighting characters that may not be otherwise apparent. Cluster A consists completely of Gram negative organisms that also tend to have large genomes (5.2 Mb vs 3.1 Mb average for entire dataset). All organisms are phylogenetically related, being members of the α, β, and γ Proteobacteria. However, these taxonomic groups are not identified as statistically significant due to the broad distribution of other members of these taxonomic groups throughout the clusters (*i.e*, B and E). This result is consistent with the diverse habitats and lifestyles associated with Proteobacteria. Cluster B contains organisms that tend to have small genome sizes, are classified as intracellular and obligate host associated, and have a low GC%. Obligate intracellular parasites tend to have smaller genomes as they require fewer genes due to obtaining resources from the host cell and smaller genomes tend to have lower GC content to facilitate evolution through an increased mutation rate. Cluster C consists of organisms that tend to be in the phylum Firmicutes, families Bacillales or Lactobacillales, are Gram positive, and are anaerobic. Cluster D contains an over-representation of Actinomycetes, Gram positive bacteria, and sporulating bacteria. Cluster E contains many phylogenetically unrelated organisms, a majority of organisms that have preferred metabolic modes other than chemoheterotrophy, and also contains a disproportionate number of Gram negative bacteria.

2.5 *Metabolic Variants Associated with Specific Clusters*

As a complementary approach to exploring organism characteristics associated with specific clusters, it is also possible to explore whether particular metabolic variants are over- or under-represented in the specific metabolic clusters. As an example, we observed that Cluster C could be divided into two subgroups, C1 and C2. The organisms in Cluster C1 are low-GC Gram positive organisms in the phylum Firmicutes with the exception of *Fusobacterium*; the subcluster can be further divided by the oxygen requirement characteristic – the organisms in class Clostridia and *Fusobacterium* are all obligate anaerobes, whereas the organisms in class Bacilli are facultative (Figure 1, Supplemental File 1). We hypothesized that there should be specific metabolic variants (likely related to respiratory systems) that would distinguish these two groups. To investigate this and similar hypotheses, we used an approach that compared the frequencies of metabolic variants

in two groups of organisms (*e.g.,* subgroups of Cluster C1) to highlight those variants that were the most different between the groups (see Methods for details). In the case of Cluster C1, there

Table 1. Over- and under-representation of characteristics by cluster

Cluster	Characteristic*	Present Inside Cluster	Present Outside Cluster	p-value
A	Genome Size	Mean = 5.2	Mean = 3.1	5.88×10^{-6}
	Gram Stain Negative	23/23 (100%)	43/78 (55%)	1.22×10^{-5}
	Gram Stain Positive	0/23 (0%)	26/78 (33%)	6.79×10^{-4}
B	Genome Size	Mean = 1.1	Mean = 4	3.71×10^{-24}
	Intracellular Survival - Obligate Intracellular	8/14 (57%)	0/87 (0%)	1.49×10^{-8}
	Free Living/Host Associated - Obligate Host Association	12/14 (86%)	10/87 (11%)	3.96×10^{-8}
	Host Type - Arthropod/Insect	8/14 (57%)	2/87 (2%)	5.94×10^{-7}
	GC Content	35.5	51.6	1.47×10^{-5}
	Free Living/Host Associated - Free Living	0/14 (0%)	50/87 (57%)	4.35×10^{-5}
	Intracellular Survival - Not Applicable	0/14 (0%)	50/87 (57%)	4.35×10^{-5}
	Habitat Outside Host - Soil	0/14 (0%)	38/87 (44%)	8.39×10^{-4}
C	Taxonomic Class - Mixed Firmicutes	10/17 (59%)	7/84 (8%)	1.17×10^{-5}
	Gram Stain - Positive	12/17 (71%)	14/84 (17%)	2.25×10^{-5}
	Oxygen Requirement - Aerobe	1/17 (6%)	47/84 (56%)	1.09×10^{-4}
	GC Content	Mean = 39.5	Mean = 51.4	1.24×10^{-4}
	Oxygen Requirement - Anaerobe	9/17 (53%)	8/84 (1%)	1.44×10^{-4}
	Gram Stain - Negative	4/17 (24%)	62/84 (74%)	1.47×10^{-4}
	Bacillales, Lactobacillales	6/17 (35%)	3/84 (4%)	5.98×10^{-4}
D	Taxonomic Class - Actinomycetes	8/18 (44%)	2/83 (2%)	7.96×10^{-6}
	Gram Stain - Positive	12/18 (67%)	14/83 (17%)	5.73×10^{-5}
	Sporulation - Sporulating	8/18 (44%)	5/83 (6%)	1.67×10^{-4}
	Gram Stain - Negative	5/18 (28%)	61/83 (73%)	5.86×10^{-4}
	Sporulation - Nonsporulating	10/18 (56%)	76/83 (92%)	6.77×10^{-4}
E	Prefered Metabolic Mode - Chemoorganoheterotroph	14/29 (48%)	69/72 (96%)	1.29×10^{-7}
	Prefered Metabolic Mode - Photolithoautotroph	6/29 (21%)	0/72 (0%)	3.75×10^{-4}
	Gram Stain - Positive	1/29 (3%)	25/72 (35%)	7.92×10^{-4}

*Genome Size given as average number of megabases in group; GC Content given as average percentage in group

are 12 metabolic variants that are unequally distributed between anaerobic and facultative organisms (p-value ≤ 0.05) within the cluster (Supplemental File 5). These 12 variants represent 7 unique subsystems associated with the synthesis of cofactors, vitamins, and isoprenoids. Three of these subsystems are associated with respiratory functions (heme and siroheme biosynthesis, sulfur related anaerobic respiratory reductases, and sodium translocating oxidoreductases). There is differential distribution between the anaerobic (4 of 5) and facultative (0 of 6) cluster members for the presence of sulfur reductases. Likewise, 4 of 5 anaerobic cluster members have an operon of *rnf* like genes encoding putative electron transport complexes associated with nitrogen fixation,

whereas none of the facultative cluster members have this operon, which is consistent with the classical differentiation of *Clostridia* from *Bacilli* organisms in the low-GC Gram positive group.

3. Discussion

We have described a novel approach to examining the metabolic relationships among bacterial genomes that focuses on the collection of metabolic variants associated with an organism. The vector of metabolic variants succinctly describes the organism's metabolic capabilities and allows for statistical comparison of vectors between organisms that is scalable to thousands of genomes. In the current study, we have provided a proof of concept with a phylogenetically diverse set of 101 bacterial genomes, comprising 846 variants and covering much of known metabolism. The variant definitions are the result of a targeted manual curation effort in the framework of the SEED database [18], which breaks down bacterial metabolism into subsystems (defined as collections of functional roles necessary to perform a cellular function). In this study, 139 subsystems were individually examined to define the possible metabolic variants. The outcome of the manual curation effort is a set of curated metabolic variants that can be rapidly assigned to bacterial genomes and used to compare the metabolic capabilities present in the genomes.

Many of the approaches to understanding the breadth, conservation and evolution of metabolic networks found in the bacterial domain have focused on properties of network architecture such as scale, network path length, network motifs, centrality, modularity and connectedness [3,4,23]. Common themes are observed in that metabolic networks have been shown to be scale-free and highly modular for most organisms. It has been shown that the complexity of a metabolic network can be associated with particular lifestyles/habitats. For example, obligate symbionts that experience relatively stable environments have less complex networks than organisms that are free-living and exposed to many environments. These approaches are highly granular in that they connect networks on the level of individual reactions, compounds and enzymes. An extension to network based approaches was introduced by Mazurie *et al.* [5] that compares higher level functional units called networks of interacting pathways. These were used to classify organisms into phenotypic categories. They observed similar trends with respect to the nature of the networks as seen with other network-based approaches and were able to assign functional pathways to organisms of particular phenotypes. For instance, free-living and host-associated organisms differed with respect to frequency of observed carbohydrate and energy metabolism pathways; motile and non-motile organisms differed with respect to xenobiotic degradation pathways. More recently, Pearcy *et al.* [6] introduced a method that produced vectors for an organism whose elements described individual network motifs. They analyzed 3 and 4 node motifs that are abstractions of specific compound and reaction connections and identified network motifs that were enriched for organisms with different habitats/lifestyles, such as aerobic/facultative vs. anaerobic. By looking at the reactions and compounds that made up the enriched motifs, it was possible to identify specific metabolites associated with the different lifestyles. Patterns such as these supported the assertion that environmental conditions shape the properties of metabolic networks that occur in organisms. In a departure from analyzing network properties, Poot-Hernandez *et al.* [24] calculated linear enzymatic step sequences (ESS) found in metabolic maps in KEGG and defined core and peripheral metabolic pathways for 40 gamma proteobacteria

species. An analysis of the relationships of ESS vectors among organisms was not conducted. Mithani *et al.* [4] analyzed the presence/absence of enzymatic reactions in pathways of *Pseudomonas* species based on KEGG map reaction mining. They found interesting patterns of gains and losses associated with niche specific adaptations to host association. Their approach is limited by the restriction to KEGG maps and boundary effects (reactions that appear in more than one map do not get connected). Further, the authors noted that other information such as genome context could improve understanding of evolutionary processes. The approach that we describe here is fundamentally different than those employed to date in that the unit being analyzed – variants associated with an organism – is non-network based; implicitly incorporates genome context, paralogs, isoenzymes and non-orthologous replacements through manual curation; and allows for coverage of metabolic capabilities across the modular nature of networks and their representation as disconnected metabolic maps. Further, each variant represents a functioning biological process, allowing the succinct assertion of organism capabilities (both positive and negative attribution). The analysis of variant vectors and the patterns observed therein give rise to clusters of metabolic forms comprised of the organisms and their individual variants. It is then possible to attribute the influence of phenotypic characters and phylogenetic relationships to these clusters through standard statistical approaches. It would be instructive to map individual variants to data types analyzed previously (e.g., networks of interacting pathways, individual network motifs, and ESS) to enable systematic comparison of each of these approaches to the variant approach.

We identified five main clusters of metabolically related organisms in our analysis (A-E in Fig. 1), each of which share some phenotypic traits (Table 1). We also described a complementary approach to evaluate which variants are most differentially distributed between clusters on the tree. These analyses yield patterns that are consistent with the approaches mentioned above. For instance, Cluster B is comprised of organisms that are host-associated and found in relatively stable environments; the 144 variants that are significantly differentially distributed ($p < 0.05$) include the absence of functions in amino acid, purine and pyrimidine, and vitamin/co-factor biosynthesis pathways (Supplemental File 5). There are other cases where there are hints at what drives the members of a cluster together in metabolic space (*e.g.*, *Neisseria*, *Pelagibacter*, *Xylella*, *Leptospira* – amino acid usage; *Gluconobacter*, *Desulfovibrio*, *Carboxydothermus* – extreme environments), but the current sampling of 101 organisms limits the statistical analysis of small clusters such as these.

These types of problems will become tractable with the inclusion of new genomes that begin to fill out metabolic signature clusters. Importantly, the fundamental structure of the dendrogram will not change as new genomes are added given a constant set of defined variants (*e.g.*, Cluster B will continue to contain organisms with small genomes/symbionts, Cluster C will contain most low GC Gram positive organisms, Cluster D will contain high GC organisms, and Cluster E will likely expand and subdivide as representation of metabolically diverse organisms increases). Organisms that do not follow these expectations may yield insight into novel combinations of metabolic capabilities; the current metabolic clusters represent a framework of hypotheses about relationships between suites of metabolic variants associated with any one organism. In contrast, as additional metabolic variants are identified, curated and assigned, the nature of the metabolic clusters may change. In short, as more well-annotated genomes are included, the statistical power

for this type of analysis increases, enhancing our ability to examine the metabolic relationships between organisms and what factors impact these metabolic commonalities.

The proof of concept described in this work serves as a foundation for identifying metabolic signatures for all sequenced bacteria and associating those signatures with specific organism characteristics and metabolic variants. Analyses of correlations between metabolic variants observed across bacterial life will enhance our understanding of the nature of the metabolic space occupied by diverse organisms.

4. Methods

4.1 *Organisms and Features*

The 101 organisms chosen were representatives of 14 phylogenetic divisions of eubacteria (Supp Fig. 2), which provides a reasonable coverage of sequenced microbial diversity with complete genomes. Each of the 101 organisms were classified on 19 different phenotypic features based on information already present in the SEED and via literature review. The features considered here and summary statistics are provided as Supplemental File 1. In order to generate maximum likelihood phylogenetic distances for each pair of organisms, we selected a representative 16S rRNA sequence of each organism from the Silva SSU Reference Set Release 106 using the ARB environment [25]. RaxML 7.0.4 [26] was then used to generate a set of maximum likelihood pairwise distances. Pairwise phylogenetic distances are included as Supplemental File 2.

4.2 *Creating a Metabolic Distance Measure*

We calculated a measure of metabolic distance, D_{FM}, between organisms based on the vector v_i, where i is the i^{th} organism, of 0's and 1's, indicating the presence/absence of the 846 subsystem variants. In general, the metabolic distance between organisms i and j, will be a function that measures the dissimilarity of vectors v_i and v_j. While there are numerous options for measuring dissimilarity or similarity between two vectors (e.g., Euclidean distance, Pearson correlation), we chose to use a novel method based on Fisher's exact test because of its robustness to the widely varying numbers of 0's and 1's observed in vector v, along with its ability to directly integrate a measure of statistical confidence into the distance measure, making D_{FM} an indirect measurement of the likelihood of two organisms possessing the observed degree of overlap in metabolism 'by chance.' To generate D_{FM}, first, for each of the 5050 ($101 * 100/2$) pairs of organisms, a 2x2 cross tabulation table was created and a Fisher's exact test p-value was generated. The Fisher's exact test p-value (that is, the likelihood of observing pattern of metabolic consistency by chance) acts as a measure of metabolic similarity and is available for all pairs of organisms in Supplemental File 6. We transformed p-values using: D_{FM}=300+ln(p) to yield a metric of metabolic distance, D_{FM}, which is always greater than 0 in our dataset.

4.3 *Statistical Analyses*

Four main statistical analyses were performed on D_{FM}. First, hierarchical clustering with complete linkage was conducted on the 101 organisms using D_{FM} as computed between all 5050 pairs of organisms. A dendrogram was created and phenotypic features were overlaid on it to aid in

interpretation of subsequent analyses. Clusters of interest on the dendrogram were determined using a false discovery rate (FDR) based on the Fisher's exact test p-values. Second, a multiple regression analysis was conducted to investigate the extent to which the metabolic distance, D_{FM}, could be explained by the 19 phenotype features. We used a dataset comprising 19 phenotype features and metabolic distance for each of the 5050 pairs of organisms (supplemental file #2). Models regressed metabolic distance on each of the 19 phenotype features. Third, we conducted analyses designed to answer the question "Which phenotypes explain why this cluster (on the dendrogram) exists?" After 'cutting' the dendrogram by looking at all of the mutually exclusive clusters for which all pairs of organisms within the cluster have a certain level of association, we wish to compare two mutually exclusive clusters of organisms to attempt to identify phenotypic differences in the clusters which are likely candidates for why the organisms separated into two mutually exclusive clusters. For categorical phenotypes, a Fisher's exact test is conducted which compares the proportion of organisms in cluster #1 with the phenotypic characteristic to the proportion of organisms in cluster #2 with the characteristic. For quantitative phenotypes, a two-sample t-test is used. Full results for all phenotypes and clusters A, B, C, D, E1 and E2 are provided in Supplemental File 7. Lastly, we conducted the same analysis as just described to answer the question "Which metabolic variants associate with specific clusters?" by using the Fisher's exact test approach on mutually exclusive clusters, evaluating association between metabolic variants and cluster memberships. Unless otherwise indicated, all analyses were conducted using R (www.r-project.org).

Supplemental Files

All supplemental files are available online at the following URL: http://homepages.dordt.edu/ntintle/metsig.zip

5. Acknowledgments

This work is supported by NSF MCB-1330734. We gratefully acknowledge discussions with Andrei Osterman, Ross Overbeek and other members of the Fellowship for the Interpretation of the Genome in early phases of this project.

References

[1] Y. Boucher and W. F. Doolittle, *Mol. Microbiol.* **37**, 703 (2000).
[2] G. Xie, C. A. Bonner, T. Brettin, R. Gottardo, N. O. Keyhani, and R. A. Jensen, *Genome Biol.* **4**, R14 (2003).
[3] A. Kreimer, E. Borenstein, U. Gophna, and E. Ruppin, *Proc. Natl. Acad. Sci.* **105**, 6976 (2008).
[4] A. Mithani, G. M. Preston, and J. Hein, *PLoS Comput. Biol.* **6**, (2010).
[5] A. Mazurie, D. Bonchev, B. Schwikowski, and G. A. Buck, *BMC Syst. Biol.* **4**, 59 (2010).
[6] N. Pearcy, J. J. Crofts, and N. Chuzhanova, *Mol. BioSyst.* **11**, 77 (2015).
[7] A. Barve and A. Wagner, *Nature* **500**, 203 (2013).
[8] J. Raymond, *Science* **311**, 1764 (2006).
[9] R. Braakman and E. Smith, *PLoS Comput. Biol.* **8**, e1002455 (2012).
[10] R. Braakman and E. Smith, *Phys. Biol.* **10**, 11001 (2013).

[11] R. Braakman and E. Smith, *PLoS One* **9**, e87950 (2014).
[12] J. Raes, I. Letunic, T. Yamada, L. J. Jensen, and P. Bork, *Mol. Syst. Biol.* **7**, 473 (2014).
[13] C. von Mering, P. Hugenholtz, J. Raes, S. G. Tringe, T. Doerks, L. J. Jensen, N. Ward, and P. Bork, *Science* **315**, 1126 (2007).
[14] T. A. Gianoulis, J. Raes, P. V. Patel, R. Bjornson, J. O. Korbel, I. Letunic, T. Yamada, A. Paccanaro, L. J. Jensen, M. Snyder, P. Bork, and M. B. Gerstein, *Proc. Natl. Acad. Sci.* **106**, 1374 (2009).
[15] S. Chaffron, H. Rehrauer, J. Pernthaler, and C. von Mering, *Genome Res.* **20**, 947 (2010).
[16] N. Fierer, J. W. Leff, B. J. Adams, U. N. Nielsen, S. T. Bates, C. L. Lauber, S. Owens, J. A. Gilbert, D. H. Wall, and J. G. Caporaso, *Proc. Natl. Acad. Sci.* **109**, 21390 (2012).
[17] C. S. Henry, R. Overbeek, F. Xia, A. A. Best, E. Glass, J. Gilbert, P. Larsen, R. Edwards, T. Disz, F. Meyer, V. Vonstein, M. Dejongh, D. Bartels, N. Desai, M. D'Souza, S. Devoid, K. P. Keegan, R. Olson, A. Wilke, J. Wilkening, and R. L. Stevens, *Biochim. Biophys. Acta* **1810**, 967 (2011).
[18] R. Overbeek, T. Begley, R. M. Butler, J. V Choudhuri, H. Y. Chuang, M. Cohoon, V. de Crecy-Lagard, N. Diaz, T. Disz, R. Edwards, M. Fonstein, E. D. Frank, S. Gerdes, E. M. Glass, A. Goesmann, A. Hanson, D. Iwata-Reuyl, R. Jensen, N. Jamshidi, L. Krause, M. Kubal, N. Larsen, B. Linke, A. C. McHardy, F. Meyer, H. Neuweger, G. Olsen, R. Olson, A. Osterman, V. Portnoy, G. D. Pusch, D. A. Rodionov, C. Ruckert, J. Steiner, R. Stevens, I. Thiele, O. Vassieva, Y. Ye, O. Zagnitko, and V. Vonstein, *Nucleic Acids Res.* **33**, 5691 (2005).
[19] R. K. Aziz, D. Bartels, A. A. Best, M. DeJongh, T. Disz, R. a Edwards, K. Formsma, S. Gerdes, E. M. Glass, M. Kubal, F. Meyer, G. J. Olsen, R. Olson, A. L. Osterman, R. a Overbeek, L. K. McNeil, D. Paarmann, T. Paczian, B. Parrello, G. D. Pusch, C. Reich, R. Stevens, O. Vassieva, V. Vonstein, A. Wilke, and O. Zagnitko, *BMC Genomics* **9**, 75 (2008).
[20] A. Osterman and R. Overbeek, Curr. Opin. Chem. Biol. **7**, 238 (2003).
[21] Y. Ye, A. Osterman, R. Overbeek, and A. Godzik, Bioinformatics **21**, i478 (2005).
[22] C. S. Henry, M. DeJongh, A. A. Best, P. M. Frybarger, B. Linsay, and R. L. Stevens, *Nat. Biotechnol.* **28**, 977 (2010).
[23] A.-L. Barabasi and Z. N. Oltvai, Nat Rev Genet **5**, 101 (2004).
[24] A. C. Poot-Hernandez, K. Rodriguez-Vazquez, and E. Perez-Rueda, *BMC Genomics* **16**, 957 (2015).
[25] W. Ludwig, O. Strunk, R. Westram, L. Richter, H. Meier, Yadhukumar, A. Buchner, T. Lai, S. Steppi, G. Jobb, W. Förster, I. Brettske, S. Gerber, A. W. Ginhart, O. Gross, S. Grumann, S. Hermann, R. Jost, A. König, T. Liss, R. Lüssmann, M. May, B. Nonhoff, B. Reichel, R. Strehlow, A. Stamatakis, N. Stuckmann, A. Vilbig, M. Lenke, T. Ludwig, A. Bode, and K.-H. Schleifer, *Nucleic Acids Res.* **32**, 1363 (2004).
[26] A. Stamatakis, *Bioinformatics* **22**, 2688 (2006).

WHEN SHOULD WE *NOT* TRANSFER FUNCTIONAL ANNOTATION BETWEEN SEQUENCE PARALOGS?

MENGFEI CAO and LENORE J. COWEN
Department of Computer Science, Tufts University,
Medford, MA 02155, USA
Email: mengfei.cao@tufts.edu and lenore.cowen@tufts.edu

Current automated computational methods to assign functional labels to unstudied genes often involve transferring annotation from orthologous or paralogous genes, however such genes can evolve divergent functions, making such transfer inappropriate. We consider the problem of determining when it is correct to make such an assignment between paralogs. We construct a benchmark dataset of two types of similar paralogous pairs of genes in the well-studied model organism *S. cerevisiae:* one set of pairs where single deletion mutants have very similar phenotypes (implying similar functions), and another set of pairs where single deletion mutants have very divergent phenotypes (implying different functions). State of the art methods for this problem will determine the evolutionary history of the paralogs with references to multiple related species. Here, we ask a first and simpler question: we explore to what extent any computational method with access only to data from a single species can solve this problem.

We consider divergence data (at both the amino acid and nucleotide levels), and network data (based on the yeast protein-protein interaction network, as captured in BioGRID), and ask if we can extract features from these data that can distinguish between these sets of paralogous gene pairs. We find that the best features come from measures of sequence divergence, however, simple network measures based on degree or centrality or shortest path or diffusion state distance (DSD), or shared neighborhood in the yeast protein-protein interaction (PPI) network also contain some signal. One should, in general, not transfer function if sequence divergence is too high. Further improvements in classification will need to come from more computationally expensive but much more powerful evolutionary methods that incorporate ancestral states and measure evolutionary divergence over multiple species based on evolutionary trees.

Keywords: protein function prediction, paralogs

1. Introduction

When new genes are sequenced and deposited into databases, a variety of manual and automated curation is involved in associating functional annotation to these genes. One of the most common practices is to transfer functions based on some threshold of sequence similarity.[1] However, when this sequence similarity threshold results in automatically transferring functional annotation between all pairs of orthologous and paralogous genes, this is deeply problematic because there are cases when the functions of the genes have diverged.[2]

In this paper, we consider the question of transfer of functional annotation *solely for paralogous genes.* It was widely believed that paralogs were more likely to acquire divergent functions than orthologs (the so-called *ortholog conjecture*),[3,4] but in recent years, this assumption has been the subject of spirited debate.[3–6] The present study requires neither a positive nor negative resolution of the ortholog conjecture, nor does it directly shed light on the conjecture itself, since it focuses only on a practical problem in the field of automatic function prediction artificially restricted to a single species: we ask whether any computational method *with access*

to information based only on the single species in which the paralogs reside, can distinguish the pairs whose functional roles are similar from those where functional roles are diverged.

We construct a benchmark dataset of two types of paralogous pairs of genes in the well-studied model organism *S. cerevisiae:* one set of pairs where single deletion mutants have very similar phenotypes (implying similar function), and another set of pairs where single deletion mutants have very divergent phenotypes (implying different function). We are fortunate in that there exist data in *S. cerevisiae* where the similarity of phenotypes of deletion mutants has been categorized: in particular, the extensive phenotype data from Hillenmeyer et al.[7] who look at the phenotypes of homozygous single gene deletion knockouts under 418 different conditions such as depletion of certain amino acids or nutrients.

The Hillenmeyer et al data[7] allows us to construct a gold-standard benchmark dataset of paralogous yeast gene pairs, some with highly similar and some with highly dissimilar functions, as follows. We consider two different datasets of paralogous gene pairs in *S. cerevisiae*. The first dataset we construct from scratch by taking pairs of yeast genes with high sequence similarity. The second dataset is derived from the study of the putative whole genome duplication event for *S. cerevisiae* by Kellis et al.[8] who identify 450 paralogous gene pairs. For each of the paralog pairs in the two datasets we compute a co-fitness score[7] to represent to what extent the two gene deletion knockouts have similar phenotypes. We choose a subset of these paralogs with very high co-fitness score and a subset of these paralogs with very low co-fitness score. The subset with the high co-fitness score are our *same* or *conserved function* paralogs, and the subset with the low co-fitness score are our *divergent function* paralogs. Note that Hillenmeyer et al.[7] has already shown that when genes are clustered using such a co-fitness score, they find clusters that are consistent with shared Gene Ontology annotations for biological process and molecular functions. Gu et al.[9] also uses fitness effect data to study functional compensation among gene duplicates.

We note that a recent study of Plata and Vitkup[10] also considered the genetic robustness and functional evolution of gene duplicates in yeast, based on the same gene deletion knockout set of Hillenmeyer et al. However, they considered a measure that is different than our co-fitness score over the collection of gene deletion mutants. In particular, under the assumption that paralogs with similar function could mutually compensate for each other whereas paralogs with divergent function could not, they considered the average number of "sensitive" conditions (i.e. conditions where a growth defect was observed with a P value cutoff of 0.01) between paralog pairs. Paralogs with a small average number of conditions where there was a growth defect (also alternatively, with a small average fraction of conditions where there was data, to deal with missing data), they assumed meant that the paralogs were mutually able to compensate for one another in the deletion mutant. We discuss how well this measure correlates with our "similar function" co-fitness score below.

In addition to nucleotide and amino acid sequence similarity, we sought to investigate whether simple features of the PPI network would also help distinguish same function from divergent function paralogs. Mika and Rost[11] showed that PPI interactions were better conserved within species than across species: a sort of anti-ortholog conjecture for interlogs. Thus it is reasonable to think that the interaction partners of a gene will be more similar for genes

with similar functions; the problem is of course complicated by the fact that existing PPI data is both noisy and also extremely incomplete. We consider some simple well-studied parameters of this network, namely degree, shortest path distance, shared neighborhood, as well as our diffusion-based DSD measure,[12,13] which has been shown to be especially robust to noise and missing data,[14] to find out to what extent these are informative features for our problem.

2. Related work

The most related paper to the current one is the previously mentioned work of Plata and Vitkup.[10] In addition, there have been some previous studies that have tried to place paralogous gene pairs into different functional categories based on a variety of information sources, including the recent SIFTER[2] which performed quite well in the past two Critical Assessment of protein Functional Annotation algorithm (CAFA) experiments[15] for automated function prediction. Unlike the present study, SIFTER assumes access to information from ancestral states, not just the species in which the paralogous gene pairs themselves reside, so they are able to leverage the power of evolutionary information. Other work[16–19] has used gene expression levels, the number of shared interacting partners, and shared Gene Ontology annotations, in order to predict or assess which pairs are instances of conserved function, subfunctionalization, and neofunctionalization. In each of these papers, ground truth for the predictions are assessed in different ways. Zeng and Hannenhalli[16] compare tissue specific gene expression levels from an ancestral gene (a single-copy gene from a closely related species) and the duplicated genes, where in the neofunctionalization case, for example, they assume the ancestral gene's expression level should be lower than that of both duplicates. In addition to this being a somewhat controversial assumption, noise in measuring expression levels can impact their conclusions. Nakhleh's group[17] uses the yeast PPI network to study the problem of categorizing different evolutionary fates of duplicate genes, but in their case, instead of using the structure of the PPI network to assist in predicting the categories, they used the network to *define* their ground truth gold standard for the categories. In particular, they define gene pairs as similar and divergent in function based on comparing the number of known interacting partners of the ancestral gene and the duplicated genes, a measure that will be very sensitive to noise and incomplete data even in the relatively well-studied yeast interactome.[20]

Our method of determining ground truth for same and divergent functions is less noise sensitive than either of these two other methods, but it is much more restrictive than the methods of previous studies. First, it presents only two categories of functional similarity and divergence. More importantly, it makes use of extensive phenotype data from single deletion mutants: a dataset available for yeast but unavailable for most other species at this time. Thus these other measures may be the only ones available in other species; conversely, if one accepts that the single deletion phenotype data is the best measure of ground truth when available for this problem, then the subject of this paper, namely, determining which *other* more easily obtainable sequence and network measures best correlate with this standard, might be the most important application to studying computational transfer of functional annotation standards in other organisms of interest.

Finally, the most common way in the field to determine if paralogs share the same function

is simply to look up the curated functional annotations in a database based on a human-created ontology structure such as MIPS[21] and GO[22] to see if they are annotated with the same functional labels. However, we note that in many databases, paralogs with nearly identical or identical sequence are often annotated with the same functional labels, even if that annotation comes from experiments with only one of the paralogs.

3. Materials and Methods

3.1. *Physical interaction network*

We download all the 141,327 physical interactions compiled from 7601 publications by BioGRID, version 3.3.122 (date March 3rd, 2015), each interaction of which is experimentally verified and associated with one of the following experimental evidence codes: "Affinity Capture-Luminescence", "Affinity Capture-MS", "Affinity Capture-RNA", "Affinity Capture-Western", "Biochemical Activity", "Co-crystal Structure", "Co-fractionation", "Co-localization", "Co-purification", "Far Western", "FRET", "PCA", "Protein-peptide", "Protein-RNA", "Proximity Label-MS", "Reconstituted Complex", "Two-hybrid". While collecting these physical interactions, we adopt the scoring scheme from Cao et al.[13] and assign real value confidence scores in $(0, 1)$ as weights to interactions, where the scoring scheme weights interactions as higher confidence when they are verified by experiments from multiple publications, plus low-throughput experiments are deemed more reliable than high-throughput experiments. Since we only consider interactions associated for genes in the list of 5091 verified ORFs (open reading frames) from the *Saccharomyces* Genome Database (download date: April 11th, 2014), we exclude the interactions that are associated with non-verified ORFs. Using the data above, we build an undirected weighted graph where a node is a protein, a weighted edge between two nodes exists if and only if there is a physical interaction between the two nodes and the weight on each edge is calculated as the confidence score. As a result, we obtain a connected simple graph, involving 5043 nodes and 79594 edges, with diameter 5.

3.2. *Duplicated gene pairs*

We collect two sets of duplicated gene pairs in two ways. We construct the first set that we call the "SequenceCover" or "SC" set based on sequence similarity using the following process: We first collect the result from all against all BLAST searches over the 5043 proteins and then build a sequence similarity graph where a node is a protein and an edge between two proteins exists if and only if the sequence identity is at least 80% and the BLAST E-value is below 10^{-5}. We then find the maximal independent edge set using a naive heuristic algorithm from the graph which satisfies two conditions: (1) if an edge (a, b) is chosen, none of a or b's neighbors will be chosen; (2) no more edges can be added to the set without violating (1). Because these conditions together with the sequence similarity threshold settings are so strict, we will generate edges for protein pairs that have very high sequence similarity and any two of them in the set will not share a common node. Note that in order to analyze the gene pairs using their fitness data, we exclude from the edge set the edges that are incident to nodes that do not have fitness data available from Hillenmeyer et al.;[7] as a result, we are restricted to

the 3732 genes out of 5043 genes have fitness data available. However, we may find different maximal independent edge sets if we choose different random seeds.

We randomly pick one of these independent maximal edge sets, which then defines our first duplicated gene pair set. For the second set of duplicated gene pairs, we download all 450 WGD gene pairs from Kellis et al.[8] The Kellis et al[8]'s gene pair set consists of gene pairs that are believed to be paralogs derived from the whole genome wide duplication event, inferred by both sequence mapping and gene locus. This data set has also been widely used by many other groups studying function of paralogous genes.[16–19,23] Again we restrict ourselves to the subset of the WGD gene pair set where both nodes have fitness data from Hillenmeyer et al.[7]

Note that by restricting the SC set to gene pairs with a relatively high degree of sequence similarity, it will miss some pairs of distant paralogs. However, since our focus is not on the behavior of the landscape of all paralogs, but rather on the scenario where one might computationally decide to transfer functional annotation based on sequence similarity, this is a reasonable threshold.

3.3. *Fitness profile*

We download all the 1,982,156 fitness defect log ratio scores (where 188,642 scores are missing) derived from the homozygous gene deletion experiments from Hillenmeyer et al.[7] Each log ratio score indicates the fitness defect for one of the 4769 homozygous gene deletion strains involving 4742 genes under one of the 418 different testing environments such as depletion of amino acids or nutrients. With respect to a strain for one gene, Hillenmeyer et al.[7] defines a fitness profile as the 418-dimensional vector where each entry is the log ratio score corresponding to each testing environment. Using the fitness profile, we then follow Hillenmeyer et al's[7] analysis and calculate a co-fitness score for each pair of genes that captures the phenotype similarity based on the growth defect under different testing environments. As a preprocessing step accounting for the missing entries, we impute the missing value as follows: 1) when only one gene has fitness values missing for a given environment, we use the same fitness value for both. 2) when both genes have their fitness values missing for a given environment, we use the mean value over all strains under that environment for both. We calculate the co-fitness scores between any two genes as the cosine distance between the fitness profile vectors as defined in Hillenmeyer et al.[7] In total, we obtain co-fitness scores for $4769 \times (4769-1)/2 = 11,369,296$ unique pairs.

In order to provide statistical analysis on the co-fitness scores for our targeted duplicated gene pairs, we define a z-score z_{cfs} as a normalized co-fitness score: $z_{cfs} = \frac{c_{ij}-\mu}{\sigma}$, where c_{ij} is the co-fitness score between gene i and gene j, μ is the mean co-fitness score over all pairs and σ is the standard deviation over all co-fitness scores. Therefore our empirical $p-value$ is computed as the probability that we will see a result at the normalized co-fitness score using the $t-distribution$ with $n-1$ degrees of freedom where n is the number of distinct pairs of strains, namely 11,369,296. We report for each of our targeted duplicated gene pairs the co-fitness score and their $p-values$, of which a higher value will indicate that the pair of genes is less likely to share phenotype similarity and thus less likely to carry out the same biological function. Since the problem we are trying to solve is to distinguish between paralogous gene pairs with divergent functions and that with shared functions, we need to have two separate

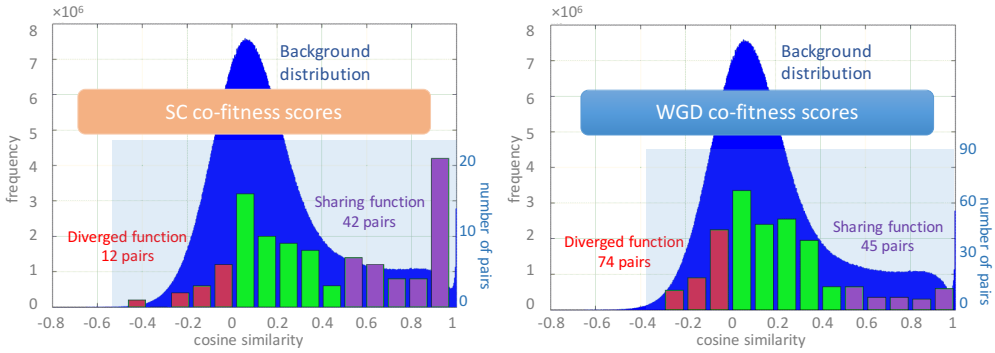

Fig. 1. Defining gene pairs with diverged functions and that with shared functions for the SC and WGD sets using the co-fitness scores.

sets of gene pairs, one set that with high confidence includes gene pairs with divergent functions and the other that with high confidence includes gene pairs with shared functions. We set the thresholds {0.00, 0.50} to define the gene pairs as with divergent functions if the co-fitness score is below 0.00 and the pairs to be with shared functions if the co-fitness score is above 0.50 as shown in Figure 1. Among the SC set where all genes in the 100 gene pairs have the fitness data available, we define 12 gene pairs with diverged functions and 42 gene pairs with shared functions; among the 450 WGD gene pairs where only 337 gene pairs have both genes' fitness data available, we define 74 gene pairs with diverged functions and 45 gene pairs with shared functions. The remaining unclassified gene pairs with co-fitness scores in the middle range, we decline to classify as "shared" or "divergent". Among the 54 classified gene pairs in the SC set, 78% pairs are considered as shared function pairs, while among the 119 classified gene pairs in the WGD set, only 38% pairs are considered as sharing function– this is not surprising as the WGD set includes gene pairs whose duplication event was in the very far past with a lot of evolutionary time to evolve mutations that could affect function.

Finally, we note that among the 119 pairs of paralogs that make up our WGD set, a total of 7 pairs lie on the same yeast chromosome. Among the 54 pairs of paralogs that make up our SC set, also a total of 7 pairs lie on the same yeast chromosome.

3.4. *Sequence similarity*

To measure amino acid similarity, for each of the classified gene pairs, we collect the BLAST bit-score, BLAST alignment length and the percentage identity as the protein sequence similarity measurements.

For nucleotide sequence similarity, for each of the classified gene pairs, we estimate the Ka score, the non-synonymous substitution rate, and the Ks score, synonymous substitution rate, as the nucleotide sequence divergence measurements.[24] More specifically, we compute the pairwise alignment for each gene pair using clustalw2,[25] then we translate the protein

alignment to a codon alignment and estimate Ka and Ks scores using the KaKs_Calculator of Zhang et al.[26] with default parameters. In addition, we also compute the Ka/Ks ratio, which is commonly considered as an indicator of selective pressure acting on a protein-coding gene. We note that for the more distant pairs, these statistics do not give reliable indications of expected evolutionary divergence, however, we can still calculate the values: we just need to assume their correlation with true evolutionary divergence is weaker.

3.5. *PPI network based measures*

For each of the classified gene pairs, we compute a set of network based similarities (or distances): the number of shared interacting neighbors, the normalized shared neighborhood size, the normalized degree difference, the normalized betweenness-centrality score difference, the shortest-path distance and the diffusion state distance (as defined by Cao et al.[13]). In the case of the normalized shared neighborhood size, degree difference and betweenness-centrality score difference, we simply divide by the maximum of the quantities for each paralog to normalize: i.e. to compute the normalized degree difference of paralogs A and B, we simply take $|deg(A) - deg(B)|/\max(deg(A), deg(B))$.

3.6. *Problem formulation*

For each of the measures defined above, we can rank the paralogous gene pairs according to each measure. However, in order to appropriately set a cutoff for each measure, beyond which we predict "conserved function" or "divergent function," we need a training set of labeled examples. First we report the predictive power of each measure described above using a leave-one-out cross validation paradigm. Namely, we learn the optimal cutoffs for classifying pairs as conserved or divergent based on all the data except the held out pair, and then classify the held-out pair according to those thresholds, and report percentage accuracy.

Then we look at the power of some standard machine learning methods when given access to all the features. In particular, we consider: decision trees, naive Bayes, support vector machines (with linear kernel), K-nearest-neighbor (with K=1), logistic regression, random forest, multilayer perceptron, one rule method, and AdaBoost with decision tree, all implemented in WEKA,[27] and see their power on the task of distinguishing the same-function from the divergent-function pairs also in leave-one-out cross validation.

4. Results

4.1. *Classification using each individual similarity measurement*

We assess the predictive power of each individual similarity/divergence measurement using the leave-one-out cross validation paradigm. Specifically, per each measurement, for each paralogous gene pair, we learn a classification threshold based on all the other gene pairs where we will classify pairs above (or below, as appropriate) the threshold as "similar function" and below the threshold (or above) as "diverged function". We then count the percent of pairs that we classify correctly. This list is somewhat deceptive in measuring true performance because of the unbalanced class sizes: but we find the nucleotide sequence-based scores uniformly more

informative than the protein sequence-based scores that we measure. Moreover the Ka and Ks scores remain good classifiers even if the thresholds are trained across the two datasets (see Table 2): for example when the Ks threshold for best classification is trained on WGD and tested on SC, and when the Ks threshold for best classification is trained on SC and tested on WGD, the percent accuracies become 88.89% and 80.67%, respectively. Figure 2 presents the scatter plot of the Ks score versus our co-fitness score.

None of the network measures perform as well as the sequence similarity measures, but the best performing network measures were related to shared neighborhood size.

Table 1. Accuracies for individual measurements			
Performance for leave-one-out-cross-validation (% Accuracy)		SC	WGD
Protein sequence measurements	AA percent identity	74.07%	78.99%
	AA BLAST alignment length	87.04%	69.75%
	AA BLAST bit-score	81.48%	63.03%
	AA ClustalW length	83.33%	76.47%
Sequence measurements	Ka	90.74%	76.47%
	Ks	88.89%	79.83%
	Ka/Ks	79.63%	71.43%
Network measurements	degree-difference (normalized)	70.37%	61.34%
	bc-difference (normalized)	72.22%	63.03%
	shared neighborhood size (SNH)	75.93%	73.11%
	normalized SNH	87.04%	72.27%
	shortest path distance	81.48%	62.18%
	DSD	74.07%	69.75%

Table 2. Cross-training performance accuracy		
	TrainOnWGDTestOnSC	trainOnSCTestOnWGD
Ka/Ks	64.81%	57.98%
Ka	87.04%	79.83%
Ks	88.89%	80.67%

4.2. *Common supervised learning methods for using all measurements*

Motivated by the observation above, we place all the 13 measurements into one feature vector for each gene pair and then try several common supervised learning methods. However, as shown in Table 3, none of the learning algorithms obtain better performance than the best performing individual measurement. Thus it remains an open question how to develop better algorithms that can separate the same-function from divergent function yeast paralogs in our set.

We also wondered whether our method of filling in missing data from the Hillenmeyer et al.[7] experiments contributed to the misclassification rates we saw. Recall that when data

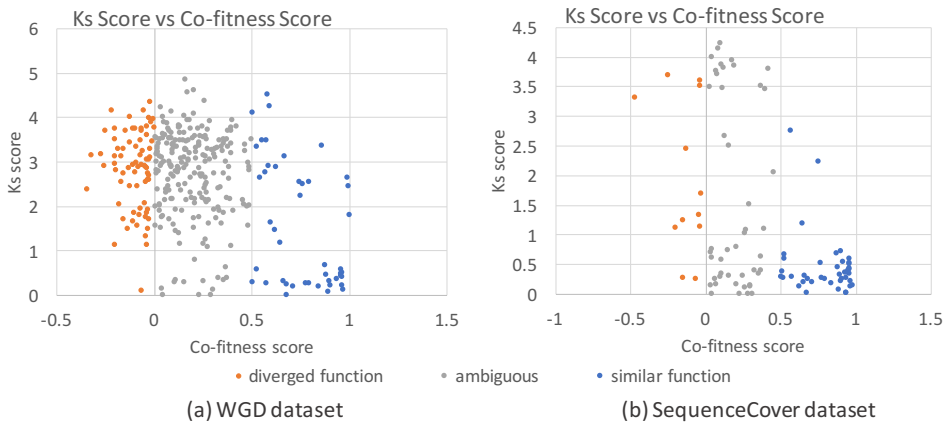

Fig. 2. Scatter plot of Ks score v.s. co-fitness scores for paralog gene pairs

Table 3. Accuracies for different learning algorithms

Performance for leave-one-out-cross-validation (% Accuracy)	SC	WGD
Decision Tree	79.63%	78.15%
Naïve Bayes	90.74%	73.95%
Support Vector Machine (linear kernel)	83.33%	78.15%
k-nearest-neighbor	90.74%	68.07%
Logistic regression	85.19%	75.63%
Random forest	87.04%	78.15%
Multilayer perceptron	87.04%	68.91%
One-Rule	83.33%	75.63%
AdaBoost + Decision Tree	85.19%	70.59%

was missing from a phenotype experiment, we filled in artificial fitness values: if the value was missing in only one of the paralogs, we matched the other paralog; if it was missing for both paralogs, we utilized the mean fitness value over all the deletion experiments for that phenotype for both paralogs. This would make yeast paralogs that are in fact divergent be more likely to have co-fitness scores that would result in our classification as "same function", if at least one had many missing values.

This did, in fact, seem to underlie some of the bad classification results for the WGD dataset in particular. For example, for the SC dataset, among the 48/54 pairs for which the Ks feature results in the correct classification, the average missing ratio is .41, whereas among the 6/54 pairs where the Ks feature results in incorrect classification, the average missing ratio is .37, whereas, for the WGD dataset, among the 95/119 examples where the Ks feature is correct, the average missing ratio is .12, whereas for the 24/119 examples where the Ks feature is wrong, the average missing ratio is .45. Removing all examples with missing data will result in too small a benchmark set; it thus remains an open question to find better ways to deal with missing values in construction of the benchmark datasets.

5. Some example paralog pairs

We looked in more detail at some of the pairs we classified as paralogs with divergent function. Because *S. cerevisiae* is so well-studied, we thought that some of the paralog pairs that we classified as divergent function, might have support from functional annotations in the SGD database, or in the literature. We found the situation quite heterogeneous– for some of the pairs we found support for functional divergence in the literature, for others, there seems to be no annotation indicating that anyone has noted any functional divergence in the two paralogs.

For example, GPP1 and GPP2 are an example of a paralog pair where some functional divergence is known. In particular, GPP1 and GPP2 seem to behave very similarly under aerobic conditions but very differently under anerobic conditions.[28] Another paralog pair where functional divergence is documented is OAF1 and PIP2. Both genes are involved in fatty acid induction of the peroxisomal β-oxidation machinery involving regulation by the oleate response element, and form a heterodimer. But OAF1 binds fatty acids and PIP2 does not.[29] GDH1 and GDH3 are both involved in glutamate biosynthesis, but their regulation indicates that they are utilized under different growth conditions: expression of GDH3 is induced by ethanol and repressed by glucose, whereas GDH1 expression is high in either carbon source.[30] TPK1 and TPK3, together with a third gene, TPK2, are functionally redundant for cell viability, but they have differing protein targets, and also recognize and affect the transcription of different sets of gene targets.[31] In each of these cases, manual inspection implies that functional labels, at least at the top levels of MIPS or GO, would correctly transfer between paralogs, despite these pairs having documented different roles within these broad functional categories.

On the other hand, the majority of the paralog pairs which we mark as "functionally divergent" have no indication in the literature or SGD database that any functional divergence between the paralogs is yet documented. ALK1 and ALK2 is a typical case. Despite a low co-fitness score, this gene pair is currently annotated in a fashion very similar to "same function" pairs: the summary description of ALK2 reads: *Protein kinase; along with its paralog, ALK1, required for proper spindle positioning and nuclear segregation following mitotic arrest, proper organization of cell polarity factors in mitosis, proper localization of formins and polarity factors, and survival in cells that activate spindle assembly checkpoint; phosphorylated in response to DNA damage; ALK2 has a paralog, ALK1, that arose from the whole genome duplication; similar to mammalian haspins.* The description for ALK1 is identical except with the names interchanged.

6. Discussion

The problem of predicting when functional annotation terms should transfer between sequence homologs and paralogs is a difficult but urgent one in the field of automatic prediction of protein function. Here, we have done a very simple study in a single, well-studied species without leveraging the wealth of evolutionary information that is available in sequences. Clearly any reasonable solution will have to leverage this evolutionary information in order to make more accurate predictions.

One clue as to the difficulty of the task might come from the alternative definitions of Plata and Vitkup.[10] We sought to measure how our co-fitness score correlated with the genetic robustness measures of Plata and Vitkup:[10] for each duplicate pair, following their paper, the

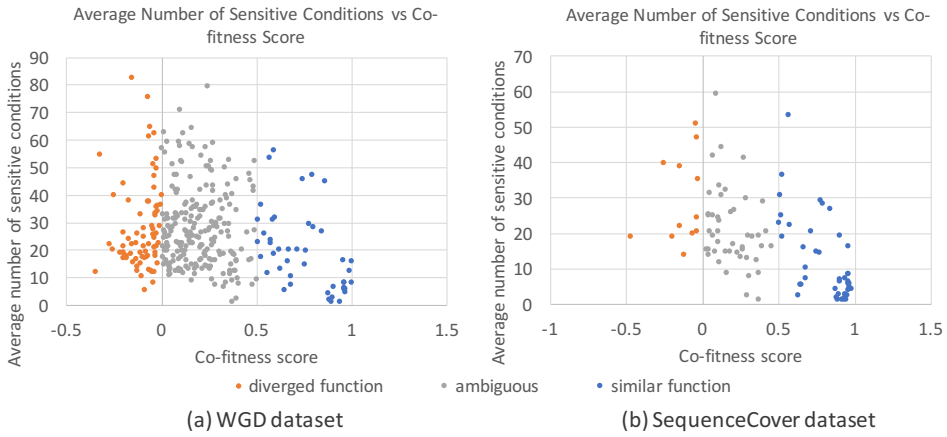

Fig. 3. Scatter plot of average number of sensitive conditions v.s. co-fitness scores for paralog gene pairs.

number of "sensitive" conditions is measured for each deletion mutant, where a "sensitive" condition is defined as a growth defect with $p < 0.01$. We report the number of sensitive conditions, averaged over the two paralogs in the pair, and plot its correlation with our co-fitness score. We find a negative correlation of -0.2046 ($P < 1.49E^{-04}$) (WGD pairs) and a negative correlation of -0.5484 ($P < 2.74E^{-09}$) (SC pairs). A scatter plot appears in Figure 3.

We notice that in both datasets, there are small but distinct set of pairs with both a very low number of average sensitive conditions, and a very high co-fitness score. For these pairs, it is hard to tell if the paralogs are similar function, or if no phenotype is frequently observed because there are third or fourth copy duplicate genes that can buffer both deletion mutants. Thus full understanding of both our co-fitness and their sensitive condition scores may require the consideration of higher order duplicates.

The SC and WGD benchmark datasets are available at bcb.cs.tufts.edu/paralogs

7. Acknowledgements

We thank the entire Tufts BCB group for helpful discussions.

References

1. I. Friedberg, *Briefings in Bioinformatics* **7**, 225 (2006).
2. S. M. Sahraeian, K. R. Luo and S. E. Brenner, *Nucleic Acids Research* , p. gkv461 (2015).
3. N. L. Nehrt, W. T. Clark, P. Radivojac and M. W. Hahn, *PLOS Comput Biol* **7**, p. e1002073 (2011).
4. R. A. Studer and M. Robinson-Rechavi, *Trends in Genetics* **25**, 210 (2009).
5. A. M. Altenhoff, R. A. Studer, M. Robinson-Rechavi and C. Dessimoz, *PLOS Comput Biol* **8**, p. e1002514 (2012).
6. X. Chen and J. Zhang, *PLOS Comput Biol* **8**, p. e1002784 (2012).
7. M. E. Hillenmeyer, E. Fung *et al.*, *Science* **320**, 362 (2008).
8. M. Kellis, B. W. Birren and E. S. Lander, *Nature* **428**, 617 (2004).
9. Z. Gu, L. M. Steinmetz, X. Gu, C. Scharfe, R. W. Davis and W.-H. Li, *Nature* **421**, 63 (2003).
10. G. Plata and D. Vitkup, *Nucleic Acids Research* **42**, 2405 (2014).

11. S. Mika and B. Rost, *PLOS Comput Biol* **2**, p. e79 (2006).
12. M. Cao, H. Zhang, J. Park, N. M. Daniels, M. E. Crovella, L. J. Cowen and B. Hescott, *PLOS One* **8**, p. e76339 (2013).
13. M. Cao, C. M. Pietras, X. Feng, K. J. Doroschak, T. Schaffner, J. Park, H. Zhang, L. J. Cowen and B. Hescott, *Bioinformatics* **30**, i219 (2014).
14. I. Fried, A. Cannistra, C. Casey, A. Piel, M. Crovella and B. Hescott, *ISMB Late Breaking Research* (2015).
15. P. Radivojac, W. T. Clark, T. R. Oron, A. M. Schnoes, T. Wittkop, A. Sokolov, K. Graim, C. Funk, K. Verspoor, A. Ben-Hur *et al.*, *Nature Methods* **10**, 221 (2013).
16. J. Zeng and S. Hannenhalli, *BMC Genomics* **14**, p. 1 (2013).
17. Y. Zhu, Z. Lin and L. Nakhleh, *G3: Genes— Genomes— Genetics* **3**, 2049 (2013).
18. A. Baudot, B. Jacq and C. Brun, *Genome Biology* **5**, p. 1 (2004).
19. L. Hakes, J. Pinney, S. Lovell, S. Oliver and D. Robertson, *Genome Biology* **8**, p. 1 (2007).
20. J.-F. Rual, K. Venkatesan *et al.*, *Nature* **437**, 1173 (2005).
21. A. Ruepp, A. Zollner, D. Maier *et al.*, *Nucleic Acids Research* **32**, 5539 (2004).
22. M. Ashburner, C. A. Ball *et al.*, *Nature Genetics* **25**, 25 (2000).
23. C. Roth, S. Rastogi, L. Arvestad, K. Dittmar, S. Light, D. Ekman and D. A. Liberles, *Journal of Experimental Zoology Part B: Molecular and Developmental Evolution* **308**, 58 (2007).
24. Z. Yang and R. Nielsen, *Molecular Biology and Evolution* **17**, 32 (2000).
25. M. A. Larkin, G. Blackshields *et al.*, *Bioinformatics* **23**, 2947 (2007).
26. Z. Zhang, J. Li, X.-Q. Zhao, J. Wang, G. K.-S. Wong and J. Yu, *Genomics, Proteomics & Bioinformatics* **4**, 259 (2006).
27. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, *ACM SIGKDD Explorations Newsletter* **11**, 10 (2009).
28. A.-K. Påhlman, K. Granath, R. Ansell, S. Hohmann and L. Adler, *Journal of Biological Chemistry* **276**, 3555 (2001).
29. A. Gurvitz and H. Rottensteiner, *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research* **1763**, 1392 (2006).
30. A. DeLuna, A. Avendaño, L. Riego and A. González, *J. of Biol. Chem.* **276**, 43775 (2001).
31. L. S. Robertson and G. R. Fink, *PNAS* **95**, 13783 (1998).

PROSNET: INTEGRATING HOMOLOGY WITH MOLECULAR NETWORKS FOR PROTEIN FUNCTION PREDICTION

SHENG WANG, MENG QU, AND JIAN PENG

Department of Computer Science,
University of Illinois at Urbana-Champaign,
Champaign, IL, USA
**E-mail: jianpeng@illinois.edu*

Automated annotation of protein function has become a critical task in the post-genomic era. Network-based approaches and homology-based approaches have been widely used and recently tested in large-scale community-wide assessment experiments. It is natural to integrate network data with homology information to further improve the predictive performance. However, integrating these two heterogeneous, high-dimensional and noisy datasets is non-trivial. In this work, we introduce a novel protein function prediction algorithm ProSNet. An integrated heterogeneous network is first built to include molecular networks of multiple species and link together homologous proteins across multiple species. Based on this integrated network, a dimensionality reduction algorithm is introduced to obtain compact low-dimensional vectors to encode proteins in the network. Finally, we develop machine learning classification algorithms that take the vectors as input and make predictions by transferring annotations both within each species and across different species. Extensive experiments on five major species demonstrate that our integration of homology with molecular networks substantially improves the predictive performance over existing approaches.

Keywords: protein function prediction, homology, molecular networks, dimensionality reduction, data integration

1. Introduction

Comprehensively annotating protein function is crucial in illustrating activities of millions of proteins at molecular level, which can further advance basic biological research and biomedical sciences.[1] Although massive annotations have been curated, such as popular Gene Ontology (GO) annotations,[2] current experimental approaches are infeasible to fully exploring protein function annotations. As a result, computational approaches have become a more accessible way to annotate protein function[3,4] and help biologists prioritize their experiments.

Computational prediction of protein function has been extensively studied in the context of molecular evolution. Homologous proteins have most likely evolved from a common ancestor. They often carry out similar protein functions, because functions are generally conserved during molecular evolution. Consequently, computational approaches can predict the function of query proteins by transferring those of their annotated homologs. In addition to automatic annotations based on orthology or domain information or pre-existing cross-references and keywords,[5] a variety of machine learning algorithms[6–12] have been proposed to extract annotations based on sequence similarity-detection tools such as BLAST, PSI-BLAST,[13] and phylogenetic analysis.[14,15] Despite the success of homology-based approaches, their major constraint arises from a lack of annotated sequences.[16] In fact, among over 65 million protein sequences in publicly accessible databases,[17] only 2 million of them are manually curated.[18] Consequently, the predictive power of homology-based methods has been limited due to the scarcity of an-

notations. Furthermore, reliable homology relationships are sparse between distantly related species, thus posing computational and statistical challenges when making faithful predictions.

Fortunately, the rapidly growing interactome data from high-throughput experimental techniques allows us to extract patterns from neighbors in molecular networks[19–21] in addition to homologous proteins. This idea is supported by the established "guilt-by-association" principle, which states that proteins that are associated or interacting in the network are more likely to be functionally related.[22] Recently, this "guilt-by-association" principle has become the foundation of many network-based function prediction algorithms.[23–30] Among them, GeneMANIA[31] and clusDCA[32] are state-of-the-art network-based function prediction approaches. In addition to incorporating network topology, clusDCA also leverages the similarity between GO labels and obtains substantial improvement on sparsely annotated functions. GeneMANIA uses a label propagation algorithm on an integrated network specifically constructed for each functional label, and is currently available as a state-of-the-art web interface for gene function prediction for many organisms.

Intuitively, integrating homology data with molecular networks can synergistically improve function prediction results. On one hand, it enables us to transfer annotations from functionally well-characterized neighbors in the molecular network as well as from homologous proteins with conserved similar functions. On the other hand, homology data can further mitigate the incomplete and noisy nature of molecular networks through interologs,[33] which states that a conserved interaction occurs between a pair of proteins that have interacting homologs in another organism.[34]

Nevertheless, integrating homology data with molecular networks is both computationally and statistically challenging. Since they are heterogeneous data sources, it is likely sub-optimal to integrate them in an additive way which simply averages the prediction results of either of these two data sources. Moreover, we also need an efficient algorithm that scales to hundreds of thousands of proteins from multiple species. One way to integrate these two heterogeneous data sources seamlessly is to construct a multiple species heterogeneous network in which both nodes and edges are associated with different types. With this network, we can predict functions for query proteins based on annotations extracted from both their homologs and their neighbors in molecular networks. Furthermore, information can also be transferred between two proteins that are neither homologs nor neighbors in molecular networks. Notably, the only previous attempt to integrate these two heterogeneous data sources is using multi-view learning.[35] However, it does not scale to multiple species. In addition, they formulated protein function prediction as a structured-output hierarchical classification problem whose performance for sparsely annotated functional labels is far from satisfactory.[32]

In this work, we introduce **ProSNet**, a novel **Pro**tein function prediction algorithm which efficiently integrates **S**equence data with molecular **Net**work data across multiple species. Specifically, an integrated heterogeneous network is first constructed to include all molecular networks of multiple species, in which homologous proteins across multiple species are also linked together. Based on this integrated network, a novel dimensionality reduction algorithm is applied to obtain compact low-dimensional vectors for proteins in the network. Proteins that are topologically close in the molecular networks and/or have similar sequences are co-localized

in this low-dimensional space based on their vectors. These low-dimensional vectors are then used as input features to two classifiers which utilize annotations from molecular networks and homologous proteins, respectively. In addition, ProSNet is inherently parallelized, which further promises scalability. When compared to the state-of-the-art methods that only use homology data or molecular networks, ProSNet substantially improves the function prediction performance on five major species.

2. Methods

As an overview, ProSNet first constructs a heterogeneous biological network by integrating homology data with molecular network data of multiple species. It then performs a novel dimensionality reduction algorithm on this heterogeneous network to optimize a low-dimensional vector representation for each protein. The vectors of two proteins will be co-localized in the low-dimensional space if the proteins are close to each other in the heterogeneous biological network. A key computational contribution is that ProSNet obtains low-dimensional vectors through a fast online learning algorithm instead of the batch learning algorithm used by previous work.[23,32] In each iteration, ProSNet samples a path from the heterogeneous network and optimizes low-dimensional vectors based on this path instead of all pairs of nodes. Therefore, it can easily scale to large networks containing hundreds of thousands or even millions of edges and nodes. After finding low-dimensional vector representation for each node, ProSNet calculates an intra-species affinity score and an inter-species affinity score by transfering annotations within the same species and across different species, respectively. Finally, ProSNet predicts functions for a query protein by averaging these scores and picking the function(s) with the highest score(s).

2.1. *Heterogeneous biological network*

Definition 1. Heterogeneous Biological Networks (HBNs) are biological networks where both nodes and edges are associated with different types. In an HBN $G = (V, E, R)$, V is the set of typed nodes (i.e., each node has its own type), R is the set of edge types in the network, and E is the set of typed edges. An **edge** $e \in E$ in a heterogeneous biological network is an ordered triplet $e = \langle u, v, r \rangle$, where $u \in V$ and $v \in V$ are two typed nodes associated with this edge and $r \in R$ is the edge type.

Definition 2. In an HBN $G = (V, E, R)$, a **heterogeneous path** is a sequence of compatible edge types $\mathcal{M} = \langle r_1, r_2, \ldots, r_L \rangle$, $\forall i$, $r_i \in R$. The outgoing node type of r_i should match the incoming node type of r_{i+1}. Any path $\mathcal{P}_{e_1 \rightsquigarrow e_L} = \langle e_1, e_2, \ldots, e_L \rangle$ connecting node u_1 and u_{L+1} is a **heterogeneous path instance** following $\mathcal{M}$, iff $\forall i$, e_i is of type r_i.

In particular, any edge type r is a length-1 heterogeneous path $\mathcal{M} = \langle r \rangle$. We show a toy example of an HBN under our function prediction framework in Fig. 1.

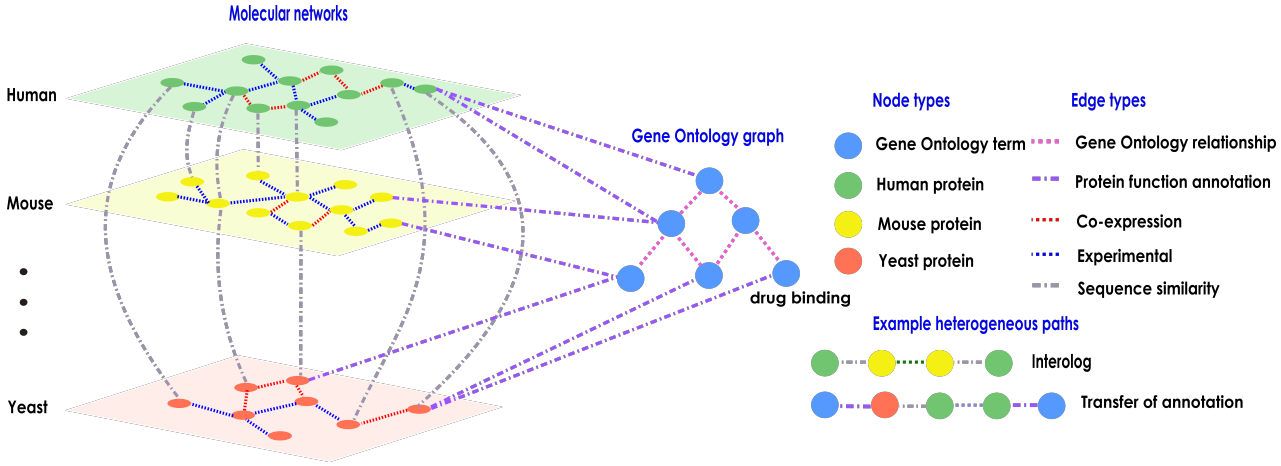

Fig. 1. An example of the heterogeneous biological network under our function prediction framework. The node set V consists of four types, {"*Human protein*", "*Yeast protein*", "*Mouse protein*", and "*Gene Ontology term*"}. The edge type set R consists of five types, {"*Sequence similarity*", "*Protein function annotation*","*Gene Ontology relationship*","*Experimental*", and "*Co-expression*" }. This HBN explicitly captures *interolog* and *transfer of annotation* through heterogeneous paths across different species.

2.2. *Low-dimensional vector learning in the heterogeneous biological network*

ProSNet finds the low-dimensional vector for each node through first sampling a large number of heterogeneous path instances according to the HBN. It then finds the optimal low-dimensional vector so that nodes that appear together in many instances turn to have similar vector representations. We first define the conditional probability of node v connected to node u by a heterogeneous path $\mathcal{M}$ as:

$$Pr(v|u,\mathcal{M}) = \frac{\exp(f(u,v,\mathcal{M}))}{\sum_{v' \in V} \exp(f(u,v',\mathcal{M}))}, \tag{1}$$

where f is a scoring function modeling the relevance between u and v conditioned on $\mathcal{M}$. Inspired from the previous work,[36] we define the following scoring function:

$$f(u,v,\mathcal{M}) = \mu_{\mathcal{M}} + \mathbf{p}_{\mathcal{M}}{}^{T}\mathbf{x_u} + \mathbf{q}_{\mathcal{M}}{}^{T}\mathbf{x_v} + \mathbf{x_u}{}^{T}\mathbf{x_v}. \tag{2}$$

Here, $\mu_{\mathcal{M}} \in \mathbb{R}$ is the global bias of the heterogeneous path $\mathcal{M}$. $\mathbf{p}_{\mathcal{M}}$ and $\mathbf{q}_{\mathcal{M}} \in \mathbb{R}^d$ are local bias d dimensional vectors of the heterogeneous path $\mathcal{M}$. $\mathbf{x_u}$ and $\mathbf{x_v} \in \mathbb{R}^d$ are low-dimensional vectors for nodes u and v respectively. Our framework models different heterogeneous paths differently by using $\mathbf{p}_{\mathcal{M}}$ and $\mathbf{q}_{\mathcal{M}}$ to weight different dimensions of node vectors according to the heterogeneous path $\mathcal{M}$.

For a heterogeneous path instance $\mathcal{P}_{e_1 \rightsquigarrow e_L} = \langle e_1 = \langle u_1, v_1, r_1\rangle, \ldots, e_L = \langle u_L, v_L, r_L\rangle\rangle$ following $\mathcal{M} = \langle r_1, r_2, \ldots, r_L\rangle$, we propose the following approximation.

$$Pr(\mathcal{P}_{e_1 \rightsquigarrow e_L}|\mathcal{M}) \propto C(u_1, 1|\mathcal{M})^{\gamma} \times Pr(\mathcal{P}_{e_1 \rightsquigarrow e_L}|u_1, \mathcal{M}), \tag{3}$$

where $C(u,i|\mathcal{M})$ represents the count of path instances following $\mathcal{M}$ with the i^{th} node being u. $C(u,i|\mathcal{M})$ can be efficiently computed through a dynamic programming algorithm. γ is a widely used parameter to control the effect of overly-popular nodes, which is set to 0.75 in

previous work.[37] We assume that each node on the path only depends on its previous node. Then we have

$$Pr(\mathcal{P}_{e_1 \rightsquigarrow e_L} | u_1, \mathcal{M}) = \prod_{i=1}^{L} Pr(v_i | u_i, r_i). \tag{4}$$

Given the conditional distribution defined in Eq. (1) and (3), the maximum likelihood training is tractable but expensive because computing the gradient of the log-likelihood takes time linear in the number of nodes. Following the noise-contrastive estimation (NCE),[38] we reduce the problem of density estimation to a binary classification, discriminating between samples from path instances following the heterogeneous path and samples from a known noise distribution. In particular, we assume these samples come from the following mixture.

$$\frac{1}{\theta + 1} Pr^{+}(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M}) + \frac{\theta}{\theta + 1} Pr^{-}(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M}), \tag{5}$$

where θ is the negative sampling weight and $Pr^{+}(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M})$ denotes the distribution of path instances in the HBN following the heterogeneous path $\mathcal{M}$. $Pr^{-}(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M})$ is a noise distribution, and for simplicity we set

$$Pr^{-}(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M}) \propto \prod_{i=1}^{L+1} C(u_i, i \,| \mathcal{M})^{\gamma}. \tag{6}$$

We further assume noise samples are θ times more frequent than positive path instance samples. The posterior probability that a given sample D came from positive path instance samples following the given heterogeneous path is

$$Pr(D = 1 | \mathcal{P}_{e_1 \rightsquigarrow e_L}, \mathcal{M}) = \frac{Pr^{+}(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M})}{Pr^{+}(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M}) + \theta \cdot Pr^{-}(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M})}, \tag{7}$$

where $D \in \{0, 1\}$ is the label of the binary classification. Since we would like to fit $Pr(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M})$ to $Pr^{+}(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M})$, we simply maximize the following expectation.

$$\begin{aligned} \mathcal{L}_{\mathcal{M}} = & \mathbb{E}_{Pr^{+}} \left[\log \frac{Pr(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M})}{Pr(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M}) + \theta \cdot Pr^{-}(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M})} \right] \\ & + \theta \cdot \mathbb{E}_{Pr^{-}} \left[\log \frac{\theta \cdot Pr^{-}(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M})}{Pr(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M}) + \theta \cdot Pr^{-}(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M})} \right]. \end{aligned} \tag{8}$$

The loss function can be derived as

$$\begin{aligned} \mathcal{L}_{\mathcal{M}} \approx & \sum_{\mathcal{P}_{e_1 \rightsquigarrow e_L} \text{ following } \mathcal{M}} \log \sigma(\sum_{i=1}^{L} f(u_i, v_i, r_i)) + \\ & \textstyle\sum_{j=1}^{\theta} \mathbb{E}_{\mathcal{P}^{j}_{e_1 \rightsquigarrow e_L} \sim Pr^{-} | u_1, \mathcal{M}} \left[\log \left(1 - \sigma(\sum_{i=1}^{L} f(u_i^j, v_i^j, r_i)) \right) \right], \end{aligned} \tag{9}$$

where $\sigma(\cdot)$ is the sigmoid function. Note that when deriving the above equation we used $\exp(f(u, v, \mathcal{M}))$ in place of $Pr(v | u, \mathcal{M})$, ignoring the normalization term in Eq. (1). We can do this because the NCE objective encourages the model to be approximately normalized and recovers a perfectly normalized model if the model class contains the data distribution.[38] Following the idea of negative sampling,[37] we also replaced $\sum_{i=1}^{L} f(u_i, v_i, r_i) - \log \left(\theta \cdot Pr^{-}(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M}) \right)$ with $\sum_{i=1}^{L} f(u_i, v_i, r_i)$ for ease of computation. We optimize parameters $\mathbf{x_u}, \mathbf{x_v}, \mathbf{p_r}, \mathbf{q_r}$, and $\mu_{\mathbf{r}}$ based on Eq. (9).

2.3. *Runtime improvements through online learning*

Like diffusion component analysis,[23] the number of pairs of nodes $\langle u, v \rangle$ that are connected by some path instances following at least one of the paths is $O(|V|^2)$ in the worst case. This is too large for storage or processing when $|V|$ is at the order of hundreds of thousands. Therefore, sampling a subset of path instances according to their distribution is the most feasible choice when optimizing, instead of going through every path instance per iteration. Thus, our method is still very efficient for networks containing large numbers of edges. Based on Eq. (3), we can sample a path instance by sampling the nodes on the heterogeneous path one by one. Once a path instance has been sampled, we use gradient descent to update the parameters $\mathbf{x_u}, \mathbf{x_v}, \mathbf{p_r}, \mathbf{q_r}$, and $\mu_\mathbf{r}$ based on Eq. (9). As a result, our sampling-based framework becomes a stochastic gradient descent framework. The derivations of these gradients are trivial and thus are omitted. Moreover, since stochastic gradient descent can generally be parallelized without locks, we can further optimize via multi-threading. Decomposing a heterogeneous network with more than sixty thousand nodes and ten million edges into a 500-dimensional vector space takes less than 30 minutes on a 12-core 3.07GZ Intel Xeon CPU through this online learning framework.

2.4. *Function prediction*

After using the above framework to find the low-dimensional vector for each protein in the HBN, ProSNet transfers annotations both within the same species and across different species to predict for a query protein.

To transfer annotations within the same species, ProSNet first uses diffusion component analysis[23] on the Gene Ontology graph[2] to find low-dimensional vector $\mathbf{y_i}$ for each functional label i. It then uses a transformation matrix $\mathbf{W}$ to project proteins from the protein vector space to the function vector space, which allows us to match proteins to functions based on geometric proximity. Let $\mathbf{y_i'}$ be the projection of the protein vector $\mathbf{x_i}$:

$$\mathbf{y_i'} = \mathbf{x_i W}. \tag{10}$$

We define the intra-species affinity score z_{ij} between gene i and function j to be used for function prediction as:

$$z_{ij} = \mathbf{x_i W y_j^T}. \tag{11}$$

A larger z_{ij} indicates that gene i is more likely to be annotated with function j. We follow clusDCA[32] to find the optimal $\mathbf{W}$.

Since proteins from different species are located in the same low-dimensional vector space, ProSNet is able to use the annotations across different species as well. Instead of using the annotations from all the other proteins, ProSNet only considers the k most similar proteins based on the cosine similarity between their low-dimensional vectors. It then calculates the inter-species affinity score s_{ij} between gene i and function j as:

$$s_{ij} = \sum_{g \in B_i} \cos(\mathbf{x_i}, \mathbf{x_g}) \cdot \mathbb{1}(g \in T_j), \tag{12}$$

where B_i is the set of k most similar proteins of i and T_j is the set of genes that are annotated to function j in the training data.

After obtaining the intra-species affinity score $\mathbf{z}$ and inter-species affinity score $\mathbf{s}$, ProSNet normalizes them by z-scores. It predicts functions for a query protein by averaging these two normalized affinity scores and picking the function(s) with the highest score(s)

3. Experimental results

3.1. *Construction of heterogeneous biological network for function prediction*

To construct the heterogeneous biological network (HBN), we obtained six molecular networks for each of five species, including human (*Homo sapiens*), mouse (*Mus musculus*), yeast (*Saccharomyces cerevisiae*), fruit fly (*Drosophila melanogaster*), and worm (*Caenorhabditis elegans*) from the STRING database v10.[20] These six molecular networks are built from heterogeneous data sources, including high-throughput interaction assays, curated protein-protein interaction databases, and conserved co-expression data. We excluded text mining-based networks to avoid potential confounding. Each edge in the molecular networks has been associated with a weight between 0 and 1 representing the confidence of interaction. Next, we obtained protein-function annotations and the ontology of functional labels from the GO Consortium.[2] We only used annotations that have experimental evidence codes including EXP, IDA, IPI, IMP, IGI, and IEP. As a result, annotations that are based on an *in silico* analysis of the gene sequence and/or other data are removed to avoid potential leakage of labels. We built a directed acyclic graph of GO labels from all three categories [biological process (BP), molecular function (MF) and cellular component (CC)] based on "*is a*" and "*part of*" relationships. This graph has 13,708 functions and 19,206 edges. We set all edge weights of protein-function links to 1 and all edge weights between GO labels to 1. Finally, we extracted amino acid sequences of all proteins in our five-species network from the STRING database and the Universal Protein Resource (Uniprot).[17] To construct homology edges, we performed all-vs-all BLAST[13] and excluded edges with E-value larger than 1e-8. We then used the negative logarithm of the E-values as the edge weights and rescaled them into $[0, 1]$. We showed the statistics of our HBN in Tab. 1. For simplicity, all edges are undirected. Note that we excluded the protein-function annotation edges that are in the hold-out test set in the following experiments for rigorous comparisons. Our heterogeneous network is similar to the example network in Fig. 1, except that our network has five species and six different types of molecular networks.

3.2. *Experimental setting*

We used 3-fold cross-validation to evaluate the methods of interest. For a given species for evaluation, we randomly split proteins of the species into three equal-size subsets. Each time, the GO annotations of proteins in one subset were held out for testing, and the annotations of the other two subsets were used for intra-species classification training. For inter-species training, we used all experimental GO annotations from the other four species, ensuring no leakage of label information in the training data. To evaluate the predictive performance, we

Table 1. Statistics of our heterogeneous network

	Human	Mouse	Yeast	Fruit fly	Worm
#proteins	16,544	16,649	6,307	11,261	13,469
#co-expression edges	1,319,562	1,406,572	628,014	2,466,234	2,774,840
#co-occurrence edges	28,334	29,472	5,328	17,962	14,678
#database edges	275,860	347,406	66,972	116,748	69,948
#experimental edges	492,548	672,326	439,956	380,046	298,684
#fusion edges	2,678	3,994	2,722	4,026	4,336
#neighborhood edges	78,440	77,962	91,220	69,934	49,890
#human homology edges	0	525,221	55,884	202,993	159,481
#mouse homology edges	525,221	0	52,916	188,729	151,408
#yeast homology edges	55,884	52,916	0	26,950	28,269
#fruit fly homology edges	202,993	188,729	26,950	0	75,831
#worm homology edges	159,481	151,408	28,269	75,831	0
#annotations	77,950	66,238	28,668	32,259	21,655

measured the extent to which the predicted ranked list was consistent with the ground truth ranked list by computing the receiver operating characteristic curve (AUROC). We used the macro-AUROC as the evaluation metric following previous work.[31,32] The macro-AUROC is calculated by separately averaging the area under the curves for each label. We set the vector dimension $d = 500$, the number of nearest neighbors $k = 2000$, and the negative sampling weight $\theta = 5$ in our experiment. We observed that the performance of our algorithm is quite stable with different d, k, and θ values. We included all edge types in the predefined heterogeneous path set. Additionally, we added "*transfer of annotation*" to the predefined heterogeneous path set (Fig. 1).

To show the improvement from integrating homology data with molecular networks of multiple species, we compared our method with three existing state-of-the-art function prediction methods: GeneMANIA,[31] clusDCA,[32] and BLAST.[13] GeneMANIA and clusDCA integrate protein molecular networks within a given species. Neither of them is able to integrate information across different species. We used the latest released code and the suggested parameter settings for these two methods. BLAST uses bit score to rank annotations from significant hits by BLAST. We used the same datasets (i.e. annotations, proteins, and networks) and the same evaluation scheme for every method we tested.

3.3. *Molecular network data and homology data are complementary in function prediction*

We first studied whether information extracted from homology and from molecular networks are complementary. We compared the predictive performance of three different data sources: 1) molecular networks, 2) homology, 3) both molecular network and homology (integrated). We used clusDCA to predict function annotations based on molecular networks. We used BLAST to make predictions of function annotation based on homology. We summarized how many functions can be accurately annotated (AUROC>0.9) by each data source (Fig. 2). We notice that there are many functions that can only be accurately predicted by homology or network. For example, on mouse MF with 3-10 labels, 9% of functions (difference between

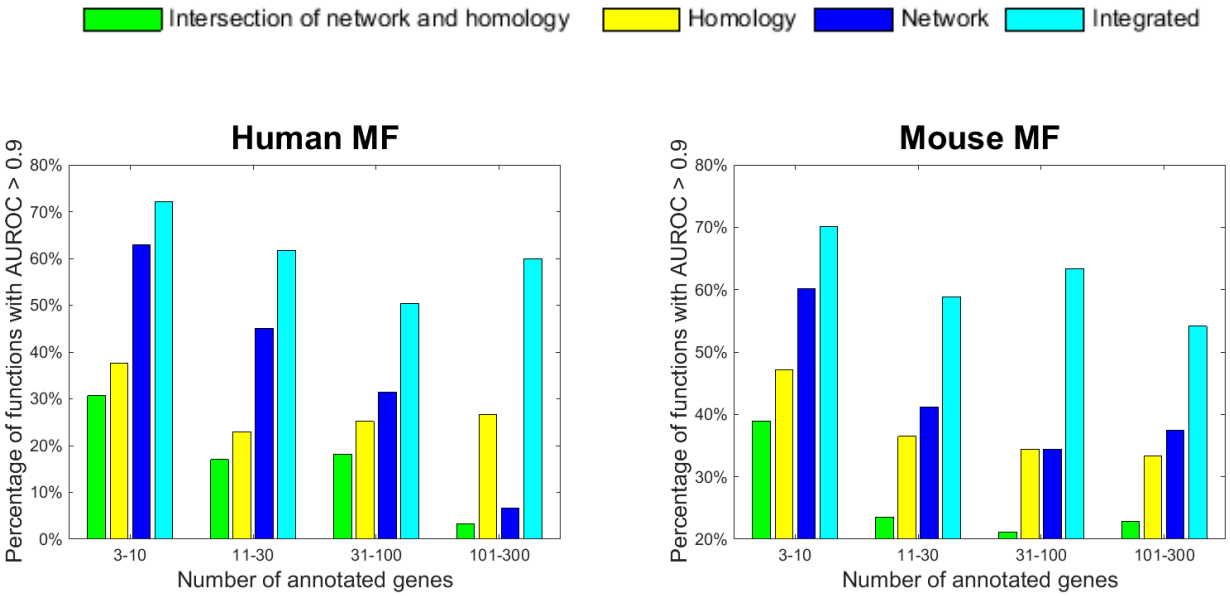

Fig. 2. Comparison of using different data sources for function prediction

yellow bar and green bar) can be accurately predicted only by homology but not by network. In the same category, another 21% of functions (difference between blue bar and green bar) can be accurately predicted only by network but not by homology. This suggests that these two data sources are complementary, and integrating them can synergistically improve the function prediction results. To this end, we integrated homology and network data by simply taking average of the z-scores of predicted annotations from these two data sources. We found that the predictive performance using both molecular network data and homology data is significantly better than only using one in all categories on both human and mouse. For example, on human MF with 101-300 labels, using both network data and homology data accurately annotates 60% of functions, which is much higher than 4% of only using network data and 26% of only using homology data. Notably, we only use the homology data from five species here. When including homology data from more species in the future, homology data may further boost the function prediction performance.

3.4. *ProSNet substantially improves function prediction performance*

We performed large-scale function prediction on all five species to compare our method to other state-of-the-art function prediction approaches. The results are summarized in Fig. 3 and Supplementary Fig. 1 (Supplementary Data). It is clear that our approach achieved the best overall results in all five species. When comparing with homology-based methods, we found that ProSNet significantly outperforms BLAST on both sparsely annotated and densely annotated labels (data not shown). For example, ProSNet achieves 0.8690 AUROC on human BP labels with 3-10 annotations, which is much higher than the 0.6326 AUROC by BLAST.

Furthermore, we compared ProSNet to existing state-of-the-art network-based methods, in-

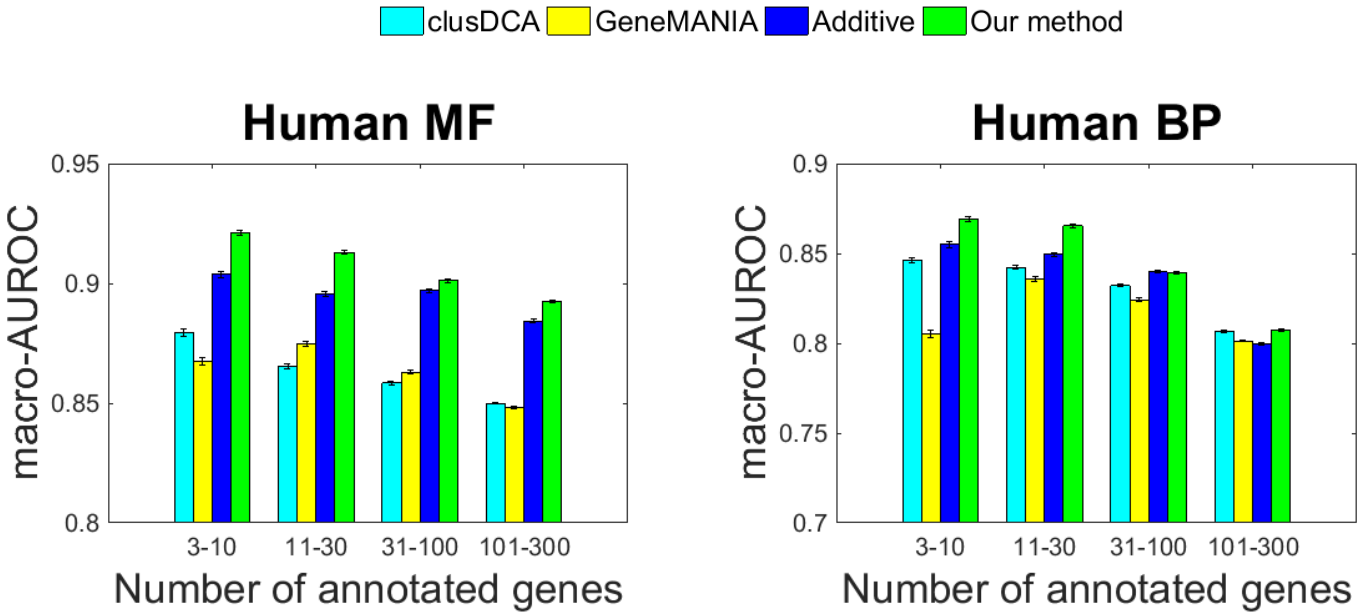

Fig. 3. Comparison of different methods

cluding clusDCA and GeneMANIA, which only integrate molecular networks of single species. We found that the overall performance of our approach is substantially higher than that of both of these methods. For instance, in human, our method achieved 0.9211 AUROC on MF labels with 3-10 annotations, which is much higher than 0.8673 by GeneMANIA and 0.8794 by clusDCA. In mouse, our method achieved 0.8523 AUROC on BP labels with 31-100 annotations, which is much higher than 0.8078 AUROC by GeneMANIA and 0.8299 AUROC by clusDCA.

To evaluate the integration of homology and network data, we developed a baseline approach that simply merges predictions made from homology data and sequence data, separately. This additive approach takes the average z-scores of the annotation score of clusDCA and BLAST to rank functional labels for each protein. We note that this baseline approach outperformes both GeneMANIA and clusDCA, indicating that integrating homology with molecular networks can substantially improve the function prediction performance. We then compared this additive approach to our method. We found that ProSNet also outperforms the additive approach. For instance, in human, our method achieves 0.9129 AUROC on MF labels with 11-30 labels, which is higher than 0.8956 AUROC by the additive approach. The improvement of our method in comparison to the additive approach demonstrates a better data integration by constructing a heterogeneous network and finding low-dimensional vector representations for each node in this network.

The improvement of ProSNet over existing network-based approaches is more pronounced on sparsely annotated functions. Since very few proteins are annotated to these functions, it is very easy to overfit any classification algorithm if we only use the data from a single

species. With the integrated heterogeneous biological network, ProSNet successfully transfers annotations from other species to have a more robust and improved predictive performance on sparsely annotated functions.

4. Conclusion

In this paper, we have presented ProSNet, a novel protein function prediction method which seamlessly integrates homology data and molecular network data. ProSNet constructs a heterogeneous network to include molecular networks from all species and homology links across different species. We have designed an efficient dimensionality reduction approach which only takes 30 minutes to decompose a heterogeneous network containing hundreds of thousands of proteins. We have demonstrated that ProSNet outperforms state-of-the-art network-based approaches and homology-based approaches on five major species. Furthermore, ProSNet has achieved improved performance over an additive integration approach that simply adds predictions from network and homology data. This result supports our hypothesis that constructing a heterogeneous network and then finding low-dimensional vector representations for each node in this network is a better data integration approach. In the future, we plan to study how to annotate proteins of species that have very sparse molecular networks or even no molecular network. In addition, we plan to pursue further improvement by integrating networks and homology data from a complete spectrum of reference species.

Supplementary Data:
`http://web.engr.illinois.edu/~swang141/PSB/ProSNetSupp.pdf`

Funding
Jian Peng is supported by Sloan Research Fellowship. This research was partially supported by grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge initiative. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

1. B. Rost, P. Radivojac and Y. Bromberg, *FEBS Letters* (2016).
2. M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin and G. Sherlock, *Nat. Genet.* **25**, 25 (May 2000).
3. P. Radivojac, W. T. Clark, T. R. Oron, A. M. Schnoes, T. Wittkop, A. Sokolov, K. Graim, C. Funk, K. Verspoor, A. Ben-Hur *et al.*, *Nature methods* **10**, 221 (2013).
4. Y. Jiang, T. R. Oron, W. T. Clark, A. R. Bankapur, D. D'Andrea, R. Lepore, C. S. Funk, I. Kahanda, K. M. Verspoor, A. Ben-Hur *et al.*, *arXiv preprint arXiv:1601.00891* (2016).
5. S. Burge, E. Kelly, D. Lonsdale, P. Mutowo-Muellenet, C. McAnulla, A. Mitchell, A. Sangrador-Vegas, S.-Y. Yong, N. Mulder and S. Hunter, *Database* **2012**, p. bar068 (2012).
6. Y. Loewenstein, L. Yaniv, R. Domenico, O. C. Redfern, W. James, F. Dmitrij, L. Michal, O. Christine, T. Janet and T. Anna, *Genome Biol.* **10**, p. 207 (2009).
7. W. T. Clark and P. Radivojac, *Proteins: Structure, Function, and Bioinformatics* **79**, 2086 (2011).

8. J. Gillis and P. Pavlidis, *BMC bioinformatics* **14**, p. 1 (2013).
9. R. Rentzsch and C. A. Orengo, *BMC bioinformatics* **14**, p. 1 (2013).
10. D. Cozzetto, D. W. Buchan, K. Bryson and D. T. Jones, *BMC bioinformatics* **14**, p. S1 (2013).
11. D. Lee, O. Redfern and C. Orengo, *Nature Reviews Molecular Cell Biology* **8**, 995 (2007).
12. G. Yachdav, E. Kloppmann, L. Kajan, M. Hecht, T. Goldberg, T. Hamp, P. Hönigschmid, A. Schafferhans, M. Roos, M. Bernhofer *et al.*, *Nucleic acids research* , p. gku366 (2014).
13. S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller and D. J. Lipman, *Nucleic acids research* **25**, 3389 (1997).
14. B. E. Engelhardt, M. I. Jordan, K. E. Muratore and S. E. Brenner, *PLoS Comput Biol* **1**, p. e45 (2005).
15. B. E. Engelhardt, M. I. Jordan, J. R. Srouji and S. E. Brenner, *Genome research* **21**, 1969 (2011).
16. Y. Jiang, W. T. Clark, I. Friedberg and P. Radivojac, *Bioinformatics* **30**, i609 (2014).
17. U. Consortium *et al.*, *Nucleic acids research* , p. gku989 (2014).
18. R. P. Huntley, T. Sawford, P. Mutowo-Meullenet, A. Shypitsyna, C. Bonilla, M. J. Martin and C. O'Donovan, *Nucleic acids research* **43**, D1057 (2015).
19. A. Chatr-Aryamontri, B.-J. Breitkreutz, S. Heinicke, L. Boucher, A. Winter, C. Stark, J. Nixon, L. Ramage, N. Kolas, L. ODonnell *et al.*, *Nucleic acids research* **41**, D816 (2013).
20. D. Szklarczyk, A. Franceschini, S. Wyder, K. Forslund, D. Heller, J. Huerta-Cepas, M. Simonovic, A. Roth, A. Santos, K. P. Tsafou, M. Kuhn, P. Bork, L. J. Jensen and C. von Mering, *Nucleic Acids Res.* **43**, D447 (January 2015).
21. T. Rolland, M. Taşan, B. Charloteaux, S. J. Pevzner, Q. Zhong, N. Sahni, S. Yi, I. Lemmens, C. Fontanillo, R. Mosca *et al.*, *Cell* **159**, 1212 (2014).
22. S. Oliver, *Nature* **403**, 601 (10 February 2000).
23. H. Cho, B. Berger and J. Peng, Diffusion component analysis: unraveling functional topology in biological networks, in *RECOMB*, 2015.
24. E. Sefer, S. Emre and K. Carl, Metric labeling and semi-metric embedding for protein annotation prediction, in *Lecture Notes in Computer Science*, 2011 pp. 392–407.
25. T. Milenkovic, V. Memisevic, A. K. Ganesan and N. Przulj, *J. R. Soc. Interface* **7**, 423 (6 March 2010).
26. M. Cao, C. M. Pietras, X. Feng, K. J. Doroschak, T. Schaffner, J. Park, H. Zhang, L. J. Cowen and B. J. Hescott, *Bioinformatics* **30**, i219 (2014).
27. A. K. Wong, A. Krishnan, V. Yao, A. Tadych and O. G. Troyanskaya, *Nucleic acids research* **43**, W128 (2015).
28. R. Sharan, I. Ulitsky and R. Shamir, *Molecular systems biology* **3**, p. 88 (2007).
29. E. Nabieva, K. Jim, A. Agarwal, B. Chazelle and M. Singh, *Bioinformatics* **21**, i302 (2005).
30. S. Navlakha and C. Kingsford, *Bioinformatics* **26**, 1057 (2010).
31. S. Mostafavi and Q. Morris, *Bioinformatics* **26**, 1759 (2010).
32. S. Wang, H. Cho, C. Zhai, B. Berger and J. Peng, *Bioinformatics* **31**, i357 (2015).
33. H. Yu, N. M. Luscombe, H. X. Lu, X. Zhu, Y. Xia, J.-D. J. Han, N. Bertin, S. Chung, M. Vidal and M. Gerstein, *Genome Res.* **14**, 1107 (June 2004).
34. A. J. Walhout, *Science* **287**, 116 (2000).
35. A. Sokolov, S. Artem and B.-H. Asa, Multi-view prediction of protein function, in *BCB '11*, 2011.
36. J. Pennington, R. Socher and C. D. Manning, Glove: Global vectors for word representation., in *EMNLP*, 2014.
37. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality, in *NIPS*, 2013.
38. M. U. Gutmann and A. Hyvärinen, *The Journal of Machine Learning Research* **13**, 307 (2012).

ON THE POWER AND LIMITS OF SEQUENCE SIMILARITY BASED CLUSTERING OF PROTEINS INTO FAMILIES

CHRISTIAN WIWIE and RICHARD RÖTTGER

Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Fyn, Denmark

E-mail: {wiwiec, roettger}@imada.sdu.dk

Over the last decades, we have observed an ongoing tremendous growth of available sequencing data fueled by the advancements in wet-lab technology. The sequencing information is only the beginning of the actual understanding of how organisms survive and prosper. It is, for instance, equally important to also unravel the proteomic repertoire of an organism. A classical computational approach for detecting protein families is a sequence-based similarity calculation coupled with a subsequent cluster analysis. In this work we have intensively analyzed various clustering tools on a large scale. We used the data to investigate the behavior of the tools' parameters underlining the diversity of the protein families. Furthermore, we trained regression models for predicting the expected performance of a clustering tool for an unknown data set and aimed to also suggest optimal parameters in an automated fashion. Our analysis demonstrates the benefits and limitations of the clustering of proteins with low sequence similarity indicating that each protein family requires its own distinct set of tools and parameters. All results, a tool prediction service, and additional supporting material is also available online under `http://proteinclustering.compbio.sdu.dk`.

Keywords: Protein Classification, Protein Evolution, Clustering

1. Introduction

With current wet-lab technology, we are producing a vast amount of genomic data at an ever increasing pace.[1] The knowledge of the very sequence of the organism is only one part of the complex puzzle of how organisms survive, reproduce and adopt to changing environmental conditions.[2] In order to benefit from the genomic data of an organism the data needs to be analyzed in an efficient and automated manner.

Of fundamental importance is the identification and classification of protein families fostering insights in the functional diversity of homologous proteins allowing to investigate the evolutionary history of the proteins.[3,4] Several, hand-curated databases exist providing information on protein family classification, e.g., SCOP[5] or PFAM.[6] Even though these databases are impressive in size, the number of known protein families is still growing with every sequenced organism.[7] Therefore, it is of importance to have reliable and automated means of classifying proteins in families, which can generally be separated into three groups:[8,9] pairwise alignment algorithms, generative models, and discriminative classifiers. Here, we are focusing on the common approach of pairwise alignments using NCBI BLAST[10] followed by a cluster analysis. There exists a myriad of clustering tools, all of them require different parameters and can only be used efficiently with a profound understanding of the underlying algorithm. Furthermore, as every clustering approach uses a different way of determining its optimal clustering, there is no universal best performer suiting all data sets equally well.[11]

There have been several studies comparing the performance of various clustering approaches for this task, discussing the problem from various points of view. For example, the

study of Chan *et al.*[12] compares the performance of two clustering tools on three different genomes in order to assess the sensitivity of these tools towards the C+G content. The main limitation of this study is the small number of data sets and tools utilized. In a different study by Bernardes *et al.*[3] a larger-scale attempt was taken to compare the general performance of four different clustering approaches on data sets similar to our setting. The main focus of the paper was to demonstrate the limitations of sequence-based similarity functions compared to their novel profile based similarity function. Nevertheless, this work applied the tools in question only to the entire SCOP data set (with various levels of sequence identities) and clustered them into families and superfamilies. This approach neglects the variety within the protein families but gives a good overview of the general performance of the tested tools.

In contrast to previous works, we create several hundred data sets comprising smaller subsets of the SCOP data set in order to strategically assess the variance of the different protein families and their consequences to the different clustering tools. Further, we clustered each of our hundreds of data sets with extensive parameter training (1,000 parameters per data set per tool) using seven popular clustering approaches which have already demonstrated to work well on protein data sets.[11] This approach allows for a more detailed evaluation of the performances and limitations of the clustering tools. We further use the massive database of 100 of thousands clustering results generated during this work in order to conduct a meta learning approach, comparable to the work of De Souto *et al.*,[13] for the prediction of the expected clustering performance and thus a tool ranking. We also suggest the parameter settings for the tools, as we can identify the most similar data set in our database together with the best parameters.

To summarize, we present an in-depth analysis of protein clustering and the inherent variability of the data sets. We intensively investigated the performance of the tools on 202 different data sets with 1,000 different parameter settings each. We investigated the behavior of the tools and their parameters, reflecting the diversity of the different protein families. With a meta-learning approach we aim to predict the expected performance of the clustering tools on unseen data sets. We utilized intrinsic properties of the data sets (e.g., matrix rank or the cluster coefficient) and used them as features of a regression model for the prediction. We also provide the performance predictor as a web-service together with all results, the source code of the predictor, and additional information at `http://proteinclustering.compbio.sdu.dk`.

2. Materials

2.1. *Data sets*

We based our work on the Astral SCOPe 2.06 data set with less than 40% sequence identity.[5] This scenario is very challenging for clustering tools as the alignment scores fall into the so-called twilight zone when the sequence identity drops below 35%.[14] The data set provides a gold standard classification derived from the SCOP database which we utilize in order to assess the cluster quality. The Astral data set classifies each protein into a hierarchy of *class*, *fold*, *superfamily* and eventually *family*.

For our goal of predicting the expected performance of the clustering tools we require a multitude of data sets. Therefore, we have created sub-samples of the Astral data set by

splitting it into classes and folds, i.e., we have created a single data set for each class, containing only the sequences of the one class and one data set for each fold in the same fashion. In the remainder we will refer to them as the *class data sets* and the *fold data sets*. This serves two purposes: (1) we received a sufficient number of data sets and (2) we were able to assess the diversity of the protein families and their impact on the clustering tools. We calculated pairwise BLAST[10] hits (E-value cut-off 100) between all protein sequences and converted them into similarities using the "Coverage BeH" method by Wittkop *et al.*[15] (coverage factor $f = 20$, cut-off $100,000$).

Given these data sets, we cluster each of them into the corresponding families, leading to the following two *scenarios*: *Class* $\rightarrow$ *Families* and *Fold* $\rightarrow$ *Families*. We performed a final filtering process by excluding all those data sets containing only one cluster, e.g., a fold containing only one family. We excluded them because they are trivial to cluster and would hugely distort the parameter prediction. After this final step we created seven class data sets and 195 fold data sets.

2.2. *Clustering Tools*

Table 1. Overview of the chosen clustering methods. We assign an abbreviation to each of the tools. We optimized the denoted parameters for each of the tools.

Abbreviation	Name	Optimized Parameter(s)
CDP	Clusterdp[16]	Kernel radius $dc \in [\wedge, \vee]$
HC(linkage)	Hierarchical Clustering[17]	Number of clusters $k \in [2, n]$
MCL	Markov Clustering[18]	Inflation $I \in [1.1, 10]$
PAM	Partitioning Around Medoids[19]	Number of clusters $k \in [2, n-1]$
TC	Transitivity Clustering[20]	$T \in [\wedge, \vee]$

We based our tool selection on the top performers (using the F1-score[21]) of a previous large-scale performance comparison of various clustering approaches,[11] summarized in Table 1. The F1-score is defined as the harmonic mean of precision and recall when comparing a cluster result with a gold standard. Generally, external validity indices (i.e., measures comparing against a gold standard) evaluate a result with regard to the purity of individual clusters and the completeness of the clusters.[11,21] In that context, the F1-score is a comprehensive measure that takes both of these into account by combining two external measures (precision and recall). The F1-score is the quasi-standard in clustering evaluation and has already proved useful in many biomedical contexts.[11,21] All considered clustering tools performed very well with an average F1-score of over 0.7 in the original study. We excluded tools which return overlapping clusters, as the F1-Score is undefined for such clusterings. We treat hierarchical clustering as three tools, depending on the linkage function used (single, complete, average).

3. Methods

3.1. *Data Statistics & Clustering*

For each data set, we calculated 25 data statistics (see Table 2). We selected these statistics to reflect a wide variety of properties of the data sets. Note, that some of the statistics are

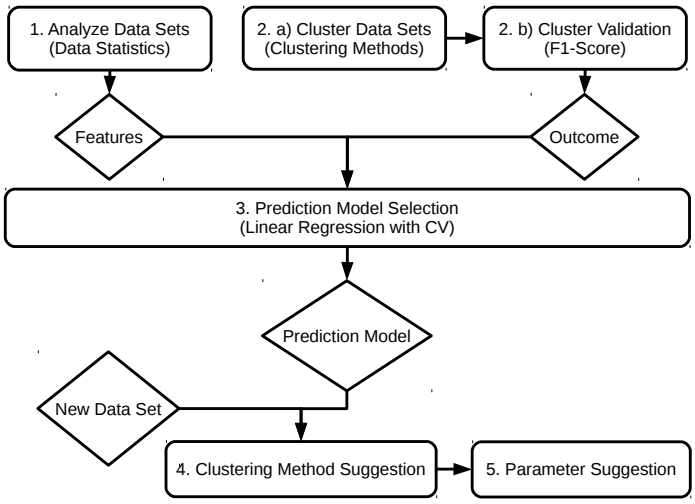

Fig. 1. Overview of the workflow of the presented method. **(1)** We calculate the features for the models, **(2a)** perform a clustering of all data sets and **(2b)** evaluate their quality. (1) and (2) are used to **(3)** train a regression model. **(4)** This model is used to predict the expected performance of each tool and suggests **(5)** the parameters.

correlated; this fact and the influence on the models is discussed in Section 3.2. The ranges of all statistics except *Minimal Similarity*, *Maximal Similarity* and *Number Samples* were normalized to $[0, 1]$ to avoid biases in the trained regression models due to differences in the value ranges.

We utilized ClustEval[11] to execute each clustering tool with 1,000 different parameter sets as indicated in Table 1 and validated the results using the F1-Score. The maximal execution time of any tool per clustering was limited to 15 minutes as we occasionally observed degenerated execution times depending on the used parameters.

3.2. *Regression & Feature Selection*

For each clustering tool we selected an ordinary, Lasso and Ridge regression model. We used the R functions *lm*, *glmnet* ($\alpha = 1$) and *glmnet* ($\alpha = 0$) to train ordinary, Lasso and Ridge regression models respectively. The data set statistics used as features for the regression models are potentially correlated and thus might be troublesome for regression models. For this reason, we perform a feature selection for the ordinary linear regression. Lasso and Ridge regression already have an intrinsic feature selection, thus they were not subject to an additional feature selection.

We trained each of the three regression models per tool using the data statistics as feature variables. The outcome variables are either the best achieved F1-Score of each tool on each data set, or the parameter leading to the best result; depending on whether we want to predict the F1-Scores or the parameters. To assess the quality of the prediction, we used the mean absolute error (MAE) to measure error rates: $\text{MAE}(\hat{y}, y) = \frac{1}{N} \sum_{i=1}^{N} |\hat{y}_i - y_i|$ where $\hat{y}_i$ denotes the prediction, y_i the real value for data set i, and N the total number of data sets. Using MAE

Table 2. Overview of the calculated data statistics. The *Absolute Z-Score*, *Assortativity* and *Similarity Percentiles* are parameterized, i.e., we calculate the same statistic multiple times for different parameters. The brackets behind the statistic name denotes the number of parameters used.

Data Statistic Name	**Description**
Absolute Z-Scores (4)	The fraction of all object pairs having a similarity within {1,2,3,4} standard deviations from the mean.
Assortativity, un/weighted[22] (2)	The preference for vertices with same degree to connect to each other in the similarity graph.
Clustering Coefficient, avg.[23]	The ratio of fully connected triplets of nodes to connected triplets of nodes in the similarity graph.
Graph Adhesion[24]	The number of edges to remove such that the similarity graph falls into several connected components.
Graph Density[25]	The ratio of the number of edges and the number of possible edges in the similarity graph.
Graph Diversity, avg.[26]	The average scaled Shannon entropy of the weights of the incident edges on each vertex in the similarity graph.
Graph Min-Cut[25]	The sum of edge weights to remove such that the similarity graph falls into several connected components.
Matrix Rank	The number of independent rows in the similarity matrix.
Maximal Similarity	The largest similarity in the similarity matrix.
Minimal Similarity	The smallest similarity in the similarity matrix.
Number Samples	The number of objects in the input data set.
Similarity Percentiles (10)	The fraction of all object pairs having a similarity within the {[0-10],[10-20],...,[90-100]} similarity percentile.

allows for easy interpretation of the error-rate compared to other measures such as the root mean squared error (RMSE).[27]

3.2.1. *Cross Validation*

In order to estimate prediction errors for a trained model we utilize a 10-fold cross validation. We repeated the cross validations 100 times with different folds to minimize the influence of a single fold. Note that the Astral data set has only seven classes, thus when only using the class data sets, a Leave-one-out cross validation (LOOCV) was performed instead.

3.2.2. *Feature Selection for Ordinary Regression Models*

We utilized a greedy forward feature selection approach coupled with 10-fold cross validations to select features and thus models with small prediction error while trying to avoid overfitting. In each step of the process, we successively added that feature to the model which lead to the smallest cross validation prediction error estimate.

During this feature selection procedure, we generate models of increasing complexity, i.e., using more features. Thus, both training and testing errors of the cross-validation will decrease in the beginning. However, with increasing number of features, the model will overfit the training data which is indicated by a growing prediction error. The moment we observe a growing prediction error, we stop adding features and report the current model as the final model. A similar feature selection procedure was previously published in Pahikkala *et al.*[28]

4. Results & Discussion

4.1. *Data Statistics*

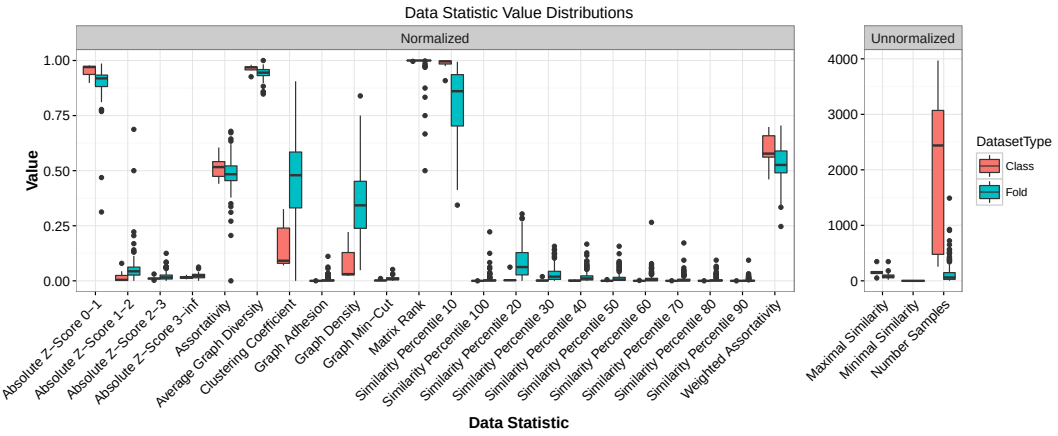

Fig. 2. The distributions of data statistic values for the class and fold data sets. We normalized data statistics using the theoretical maxima where available. The statistics *Minimal Similarity*, *Maximal Similarity* and *Number Samples* are not normalized.

Figure 2 summarizes the calculated data statistics for both class and fold data sets. Generally, some statistics such as *Graph Min-Cut*, *Graph Density* or *Clustering Coefficient* emphasize the sparsity of the pair-wise similarity matrix of the protein sequences. This is due to the fact that the proteins in the Astral data set do not have large sequence similarities resulting in many protein pairs without any significant BLAST hit. Further, we want to highlight two interesting observations:

(1) There is a clear difference in the statistical properties between the class and fold data sets. Again, this is due to the many protein pairs without any BLAST hit. The ratio of these pairs is larger in the class data sets which contain even more distantly related proteins. This is most clearly seen on Statistics such as *Average Graph Diversity*, *Clustering Coefficient*, *Graph Density*, *Similarity Percentile 10/20* and *Absolute Z-Score 0-1/1-2* which are very sensitive to this proportion.

(2) Even data sets of the same type (i.e., fold or class) vary hugely. This demonstrates the variety of the different protein families. This is even more pronounced in the fold data sets as they contain fewer families and thus are more susceptible to "outlier" families whereas in the class data sets, the variety of the different statistics is generally more balanced.

4.2. *Clustering Tool Performances*

We clustered all data sets into protein families using the clustering tools summarized in Table 1 to all previously mentioned class and fold data sets. The resulting F1-Scores are depicted in Figure 3. Generally, the selected clustering methods perform well on the data sets. HC(complete), MCL and PAM perform on average slightly worse than their competitors. The

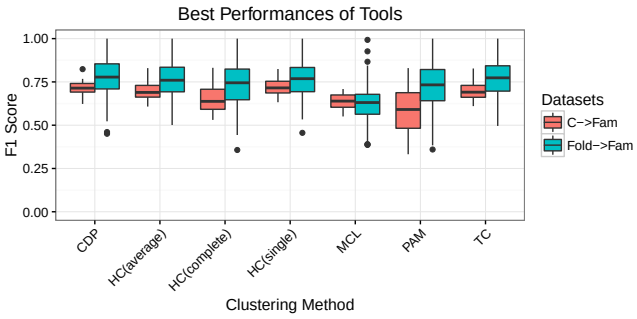

Fig. 3. The best tool performances as F1-Scores for the two scenarios: Clustering class (C) or fold (Fold) into families (Fam).

performance of PAM on class data sets might be due to our execution time limit of 15 minutes per clustering. For k-parameter values close to the real number of clusters in the classes, the algorithm does not finish in time. On the other hand, we only have seven of those data sets in this study, so the effect on the performance should be limited. None of the other methods were affected by the time limit. The general trend is that fold data sets can be clustered better (on average) than class data sets which can be explained by the fact that the class data sets are sparser. When ranking the tools by their F1-Score performance for each data set it shows that there is no best performer across all data sets, as expected. Rather, several tools alternate in taking the top ranks. The lack of a universal best performer and the variance in the rankings emphasize that performances and rankings are highly data set dependent. This further motivates the demand of a predictor based on data statistics.

4.3. *Clustering Tool Parameters*

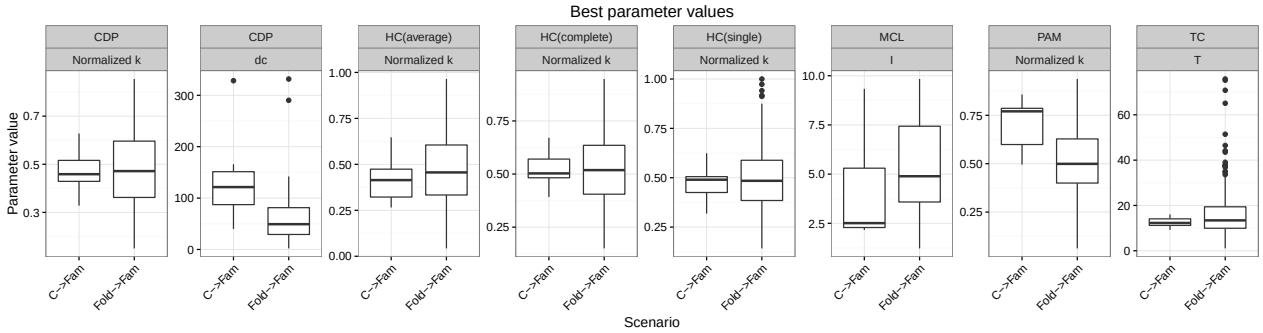

Fig. 4. The best performing parameter values of all tools. If a tool performed best with multiple parameter values we took the mean. Each tool was executed using 1,000 different parameter values. Because CDP has two main parameters we assessed both of them with an equal number of values: $32 * 32 = 1024$. Note that the parameter k is normalized by the number of objects.

We compared the best parameters of each tool for the two scenarios. Figure 4 summarizes our findings. Clearly, when clustering a fold data set we can observe a considerably larger

variety for all tools. Parameters directly reflecting the desired number of clusters, i.e. k, have been normalized with the number of objects in the data set. Please note, that we cannot use the mean k parameter as a general "rule-of-thumb" as this value entirely depends on the average family size in the data set which is determined by the way we created the data sets. Nevertheless, the variance in the k parameters certainly demonstrate the variance in protein families. The only outlier with respect to the k parameter is PAM, again likely due to the runtime restriction.

Interestingly, the parameters of CDP and MCL have different means when clustering classes compared to clustering folds. This has practical implications, as for an unknown data set it is impossible to determine whether it is comprised of a class, a fold or a mixture. The threshold T of TC remains stable regardless of the data set type, with a larger variance for the fold data sets, including some significant outliers. Overall, this indicates that a naive parameter suggestion for arbitrary protein data sets is not feasible at least it does not do justice to the variety present in different protein families.

4.4. *Predicting Tool Performance*

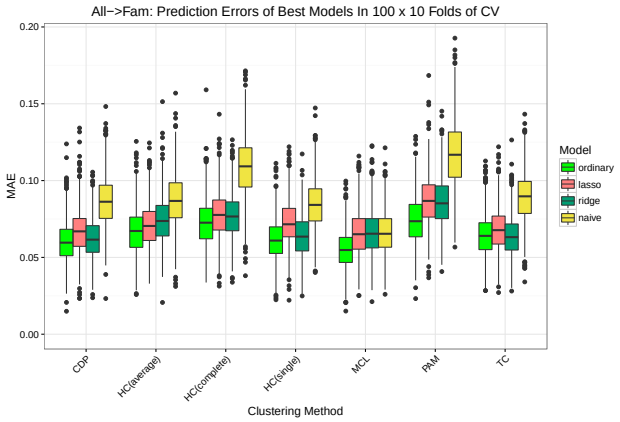

Fig. 5. The tool performance prediction errors for the final models of each tool when trained on all data sets. The prediction errors were estimated with 100×10-fold cross validations. The yellow boxes represent the performance of the naive model.

Figure 5 compares the tool performance prediction errors of the final models for all tools when trained on all data sets. We also calculate a naive predictor serving as baseline which predicts the average performance of each tool over all training data sets.

Generally, our final models outperform the naive models for all clustering tools except MCL (difference in MAE of ≥ 0.025). Note that prediction errors are relatively low for both kinds of models as all clustering tools performed well on the selected data sets. On average, the predictions of the naive models have an MAE ≈ 0.1, while those of our final models show an MAE ≈ 0.075. Ordinary models generally outperform Lasso and Ridge regression models in terms of MAE. The general trend is MAE(ordinary) $<$ MAE(lasso) $<$ MAE(ridge). However, the differences between ordinary, Lasso and Ridge regression are very small.

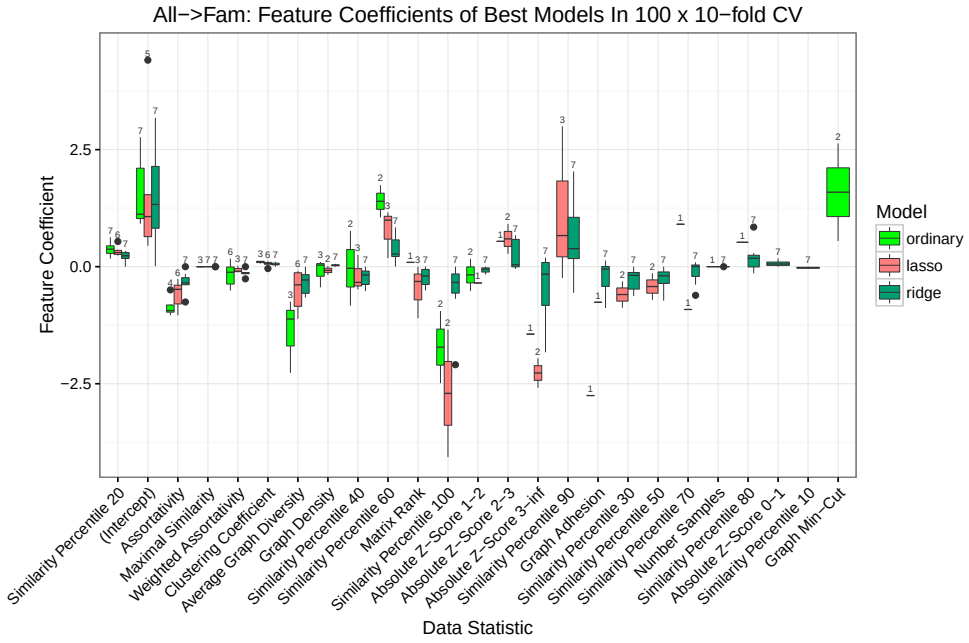

Fig. 6. This figure depicts the features and their average value in the different regression models for tool performance prediction. The features are sorted according to how often they have been selected by all models. We treated features with zero coefficient as not being in the model. The small number above each box indicates how often the feature was selected by the model represented by the box. Please note that the feature *Minimal Similarity* was never selected and thus is omitted from the figure.

Here, we want to point out the limitations of the models presented. A meaningful prediction is only possible in case the features of the unknown data set are in the same range as the features of the training data sets. We have chosen the ASTRAL data set with only up to 40% sequence similarity as we expected to observe here the most extreme feature distributions compared to data sets with higher sequence similarity.

Therefore, we have also tested the performance of the prediction with data sets not used for training. For that we have used the SCOP data set with proteins having 95% or less sequence identity; we proceeded as with the original data set and separated it also into the different classes. The error for the predicted F1 score with 0.083 for Lasso and 0.084 for Ridge regression was still remarkably small. Only the ordinary regression model showed a clear drop in performance with an average error of 0.191. This indicates that the ordinary regression is the most sensitive model with respect to unseen feature values. We will constantly update the model with new clustering results in order to further improve the quality and robustness of the models over time. To this point, the presented models should rather be regarded as a proof-of-concept.

Furthermore, we compared which data statistics have been chosen as features in the different types of models (see Figure 6). Features that have been chosen by all models clearly have predictive power for the tool performances. Examples for such features are the $[10, 20]$-*Similarity Percentile*, *Assortativity*, *Maximal Similarity* and *Weighted Assortativity*. The coefficients of the maximal similarity are very small compared to the other features, as this feature

is not normalized and thus takes large values across the data sets.

The *Graph Diversity* measures whether a node in the similarity graph is very similar to only few other nodes (low diversity) or is equally similar to many nodes (high diversity). All model types chose this statistic as a predictor with negative impact on the tool performance. This might be explained by the fact that a very high diversity implies equal similarities between all nodes, leading to the lack of an actual cluster structure.

Interestingly, the selected *Similarity Percentile* statistics indicate that details of the similarity distribution have a large predictive power for the tool performance. For example, many pairwise similarities between the $[10-20]$-*Similarity Percentile* indicate a better tool performance while fewer pairwise similarities between the $[90-100]$-*Similarity Percentile* have the opposite effect.

Surprisingly, the *Clustering Coefficient* does not enter many models with a large coefficient. Equally surprising, given the performance difference between the class and fold data sets, is that the data set size is only very rarely chosen as a feature.

4.5. *Predicting Tool Parameters*

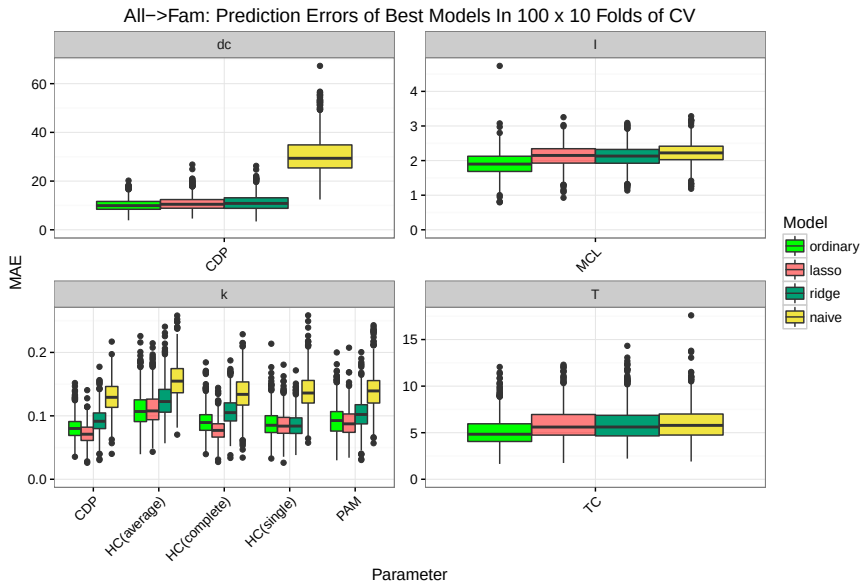

Fig. 7. The prediction performances for clustering tool parameters when trained on all data sets. Note that the various k parameters are summarized in one common plot and are normalized by the data set size.

As already previously discussed, a simple parameter suggestion valid for all data sets is not feasible due to the large variance in the protein families. Therefore, we applied the same pipeline as for the quality prediction to the parameters of the tools as well.

The results are summarized in Figure 7 and show a more mixed quality. We do not outperform the naive predictor for the threshold parameter T of TC and the Inflation parameter of MCL. We clearly outperform the naive predictor in the case of the *dc* parameter of CDP as well as the k parameters of all tools using such a parameter. Nevertheless, as discussed

earlier, the k parameter is highly dependent on the way we have sampled the data sets, thus the predictive power has to be taken into account with care. Overall, the results indicate that an automated parameter prediction is not reliably possible with the presented simple models and may require more test data and more sophisticated models. In practice, the user has to resort to other methods for finding suitable parameters.[29]

5. Conclusion

With this work, we have thoroughly investigated the performance of seven well-known and established clustering tools and have particularly investigated the behavior of the tools' parameters. We have observed that all tools perform quite well on these data sets. Nevertheless, the good performance can only be reached when exhaustive parameter finding by means of a comparison against a gold standard is performed. In practice, such gold standards are not available and consequently the parameters need to be retrieved by different means. When investigating the behavior of the parameters, we cannot suggest the user a single parameter for all data sets due to the high variance of the protein families. Only TC shows a consistent behavior of a parameter which is not directly dependent on the number of clusters. Overall, a single fixed parameter cannot account for the potential variety in the data sets. Even though the k parameter also shows a consistent behavior, it is not suitable for any recommendations as this behavior results from the way we have sampled our data sets which cannot be expected in practice.

Given this massive repository of clustering results at hand, we utilized it for learning regression models for predicting the expected performance of the investigated tools on previously unseen data sets. The presented model does outperform the naive model. Especially when considering that all clustering tools performed constantly well, the achieved prediction accuracy is notable. We also tested the models on data sets which have not been part of the training process. This can be seen as a strong indicator that it is generally possible to identify data sets suitable for a particular tool in an automated fashion. We have created a web-service where the user can upload a data set and receive the expected performance of the different tools. Please be advised that the model might fail when presented with data sets whose feature values are outside of the range of values the model was trained on. The web service also presents the features of the most similar training data set for comparison. The service is available under `http://proteinclustering.compbio.sdu.dk`. We will constantly enhance the model with additional data in order to cover a broader variety of data set features and thus creating more reliable predictions.

More generally speaking, the study shows that state-of-the-art clustering tools, when presented only with sequence similarities, have limitations with capturing the high diversity of protein families and require a specific parameter for every data set which cannot be easily provided in practice. Nevertheless, the performance achieved by the tools is certainly good enough to render this approach a viable one; probably the biggest limitation is due to the rather simple similarity function only using sequence data. Fed with more sophisticated similarity functions, these tools might be able to capture the nature of the data set even better.

References

1. M. L. Metzker, *Nature reviews genetics* **11**, 31 (2010).
2. J. Baumbach, A. Tauch and S. Rahmann, *Briefings in bioinformatics* **10**, 75 (2009).
3. J. S. Bernardes, F. R. Vieira, L. M. Costa and G. Zaverucha, *BMC bioinformatics* **16**, p. 1 (2015).
4. S. Whelan and N. Goldman, *Molecular biology and evolution* **18**, 691 (2001).
5. N. K. Fox, S. E. Brenner and J.-M. Chandonia, *Nucleic Acids Research* **42**, D304 (dec 2013).
6. R. D. Finn, P. Coggill, R. Y. Eberhardt, S. R. Eddy, J. Mistry, A. L. Mitchell, S. C. Potter, M. Punta, M. Qureshi, A. Sangrador-Vegas *et al.*, *Nucleic acids research* **44**, D279 (2016).
7. V. Kunin, I. Cases, A. J. Enright, V. de Lorenzo and C. A. Ouzounis, *Genome biology* **4**, p. 1 (2003).
8. J. Chen, B. Liu and D. Huang, *BioMed Research International* **2016** (2016).
9. L. Liao and W. S. Noble, *Journal of computational biology* **10**, 857 (2003).
10. S. F. Altschul, W. Gish, W. Miller, E. W. Myers and D. J. Lipman, *Journal of Molecular Biology* **215**, 403 (oct 1990).
11. C. Wiwie, J. Baumbach and R. Röttger, *Nature Methods* **12**, 1033 (sep 2015).
12. C. X. Chan, M. Mahbob and M. A. Ragan, *BMC bioinformatics* **14**, p. 1 (2013).
13. M. C. De Souto, R. B. Prudencio, R. G. Soares, D. S. De Araujo, I. G. Costa, T. B. Ludermir and A. Schliep, Ranking and selecting clustering algorithms using a meta-learning approach, in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008.
14. B. Rost, *Protein engineering* **12**, 85 (1999).
15. T. Wittkop, J. Baumbach, F. P. Lobo and S. Rahmann, *BMC Bioinformatics* **8**, p. 396 (2007).
16. A. Rodriguez and A. Laio, *Science* **344**, 1492 (jun 2014).
17. R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, (2014).
18. S. Dongen, *A Cluster Algorithm for Graphs*, tech. rep. (Amsterdam, The Netherlands, The Netherlands, 2000).
19. M. Maechler, P. Rousseeuw, A. Struyf, M. Hubert and K. Hornik, *cluster: Cluster Analysis Basics and Extensions*, (2016). R package version 2.0.4 — For new features, see the 'Changelog' file (in the package source).
20. T. Wittkop, D. Emig, S. Lange, S. Rahmann, M. Albrecht, J. H. Morris, S. Böcker, J. Stoye and J. Baumbach, *Nature Methods* **7**, 419 (jun 2010).
21. J. Handl, J. Knowles and D. B. Kell, *Bioinformatics* **21**, 3201 (may 2005).
22. M. E. J. Newman, *Physical Review E* **67** (feb 2003).
23. A. Barrat, M. Barthlemy, R. Pastor-Satorras and A. Vespignani, *Proc Natl Acad Sci U S A* **101**, 3747 (Mar 2004).
24. D. R. White and F. Harary, *Sociological Methodology* **31**, 305 (2001).
25. R. Diestel, *Graph Theory (Graduate Texts in Mathematics)* (Springer, 2006).
26. N. Eagle, M. Macy and R. Claxton, *Science* **328**, 1029 (2010).
27. C. J. Willmott and K. Matsuura, *Climate research* **30**, 79 (2005).
28. T. Pahikkala, A. Airola and T. Salakoski, Speeding up greedy forward selection for regularized least-squares, in *2010 Ninth International Conference on Machine Learning and Applications*, (Institute of Electrical & Electronics Engineers (IEEE), dec 2010).
29. R. Röttger, P. Kalaghatgi, P. Sun, S. de Castro Soares, V. Azevedo, T. Wittkop and J. Baumbach, *Bioinformatics* , p. bts653 (2012).