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 annotations. 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.

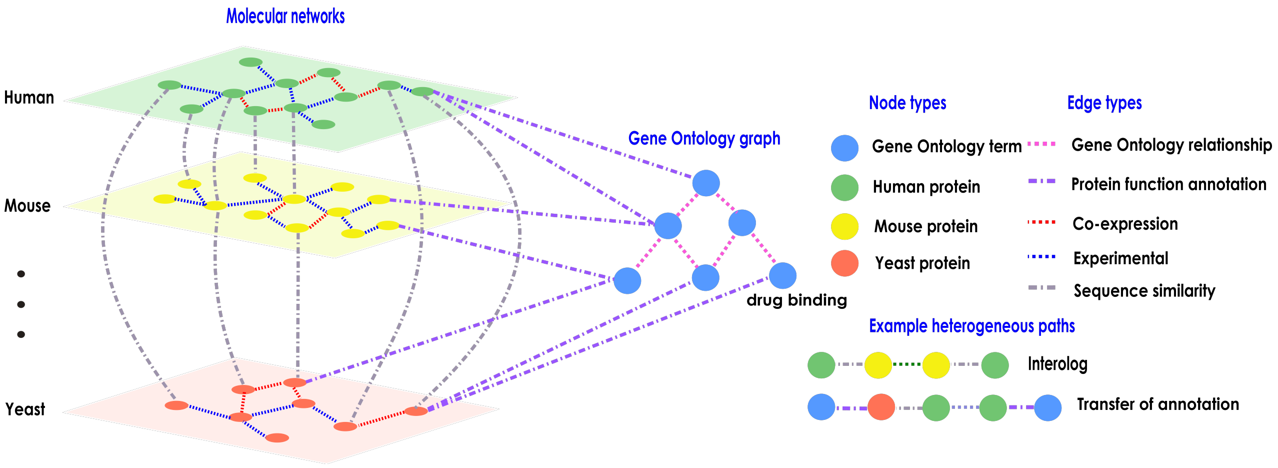

Fig. 1. An example of the heterogeneous biological network under our function prediction framework. The node set V consists of four types, {"*Human protein*", "*Yeast protein*", "*Mouse protein*", and "*Gene Ontology term*"}. The edge type set R consists of five types, {"*Sequence similarity*", "*Protein 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

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

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

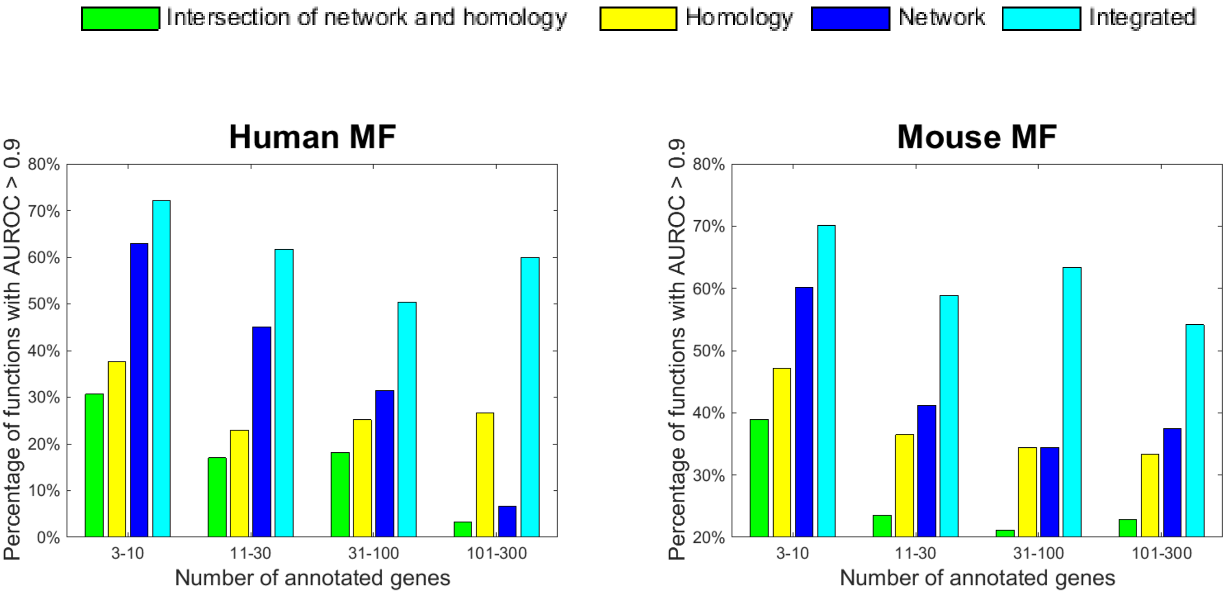

Fig. 2. Comparison of using different data sources for function prediction

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-

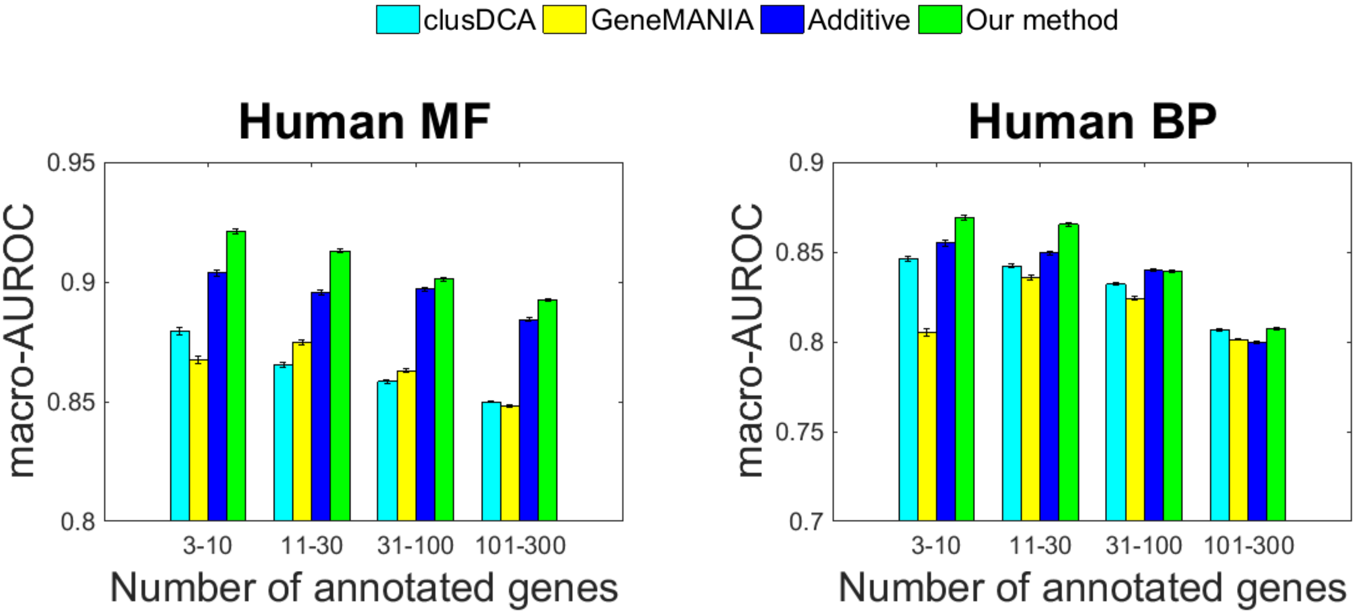

Fig. 3. Comparison of different methods

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).