PREDICTION OF PROTEIN FUNCTION FROM CONNECTIVITY OF PROTEIN INTERACTION NETWORKS

L. Shi,* Y.-R. Cho,** and A. Zhang*

Abstract

Determining protein function on a proteomic scale is a major challenge in the post-genomic era. Right now only less than half of the actual functional annotations are available for a typical proteome. The recent high-throughput bio-techniques have provided us large-scale protein–protein interaction (PPI) data, and many studies have shown that function prediction from PPI data is a promising way as proteins are likely to collaborate for a common purpose. However, the protein interaction data is very noisy, which makes the task very challenging.

In this paper, a distance matrix is proposed based on the small-world property and connectivity of the PPI network. It measures the reliability of edges and filters the noise in the network. In addition, we design an ANN (artificial neural network) method to predict protein functions with integration of several protein interaction data sets. Our approach is tested with MIPS functional categories and the experiential results show that our approach has better performance than other existing methods in terms of precision and recall.

Key Words

Protein–protein interaction network, protein function prediction, weighted network, neural network

1. Introduction

The classical way to predict protein functions is to find homologies between an unannotated protein and other proteins using sequence similarity algorithms, such as FASTA [1] and PSI-BLAST [2]. The function of the unannotated protein can then be assigned according to the annotated proteins with similar sequences. In addition, several computational approaches are proposed based on correlated evolution mechanisms of genes. For example, the domain fusion analysis infers that a pair of proteins interacts with each other and thus performs related functions [3].

* Computer Science & Engineering Department, State University of New York at Buffalo, Buffalo, NY 14260, USA; e-mail: {lshi2, azhang}@cse.buffalo.edu

** Department of Computer Science, Baylor University, Waco, TX 76798, USA; e-mail: young-rae_cho@baylor.edu

(paper no. 210-1009)

In recent years, the high-throughput bio-techniques have provided additional opportunities for inference of protein functions. Protein–protein interaction (PPI) data, enriched by high-throughput experiments including yeast two-hybrid analysis [4, 5], mass spectrometry [6, 7] and synthetic lethality screen [8], have provided the important clues of functional associations between proteins. Proteins are likely to collaborate for a common purpose. Therefore, the functions of an unannotated protein can be deduced when the functions of its binding partners are known.

There are several approaches proposed to predict protein functions with protein interaction networks. The neighbour counting method [9] uses the majority-rule to label a protein with the functions that occur most frequently in its interaction partners. Some caveats of this approach are that it can only predict up to three functions and it doesn't take into account any significance value and the full topology of the network. To solve the above problem, Hishigaki *et al.* [10] use a chi-square statistics to calculate the significance of the functions of neighbour proteins. In detail, they examine the n-neighbourhood of a protein. For a protein p, each function f is assigned a score. Those functions with higher score than a threshold will be kept as predicted functions for protein p. A shortcoming of this approach is that within the n-neighbourhood, proteins at different distances from p are treated in the same way. Chua *et al.* [11] try to tackle the problem by investigating the relation between network distance and functional similarity. They focus on the 1- and 2-neighbourhoods of a protein, and devise a functional similarity score that gives different weights to proteins according to their distances from the target protein. In addition, these methods can only predict the proteins which have at least one interaction partner. This means lots of unknown proteins cannot be predicted by these methods. Moreover, the predicted annotations for an unknown protein are limited by the annotations of its interacting partners.

To avoid those limitations, several other approaches are proposed to use the global topology of protein interaction networks. Vazquez *et al.* [12] assign a function f to each unannotated protein p so as to maximize the number of edges that connect proteins assigned with the same function. This optimization problem, which generalizes the

computationally hard problem of minimum multiway cut, is heuristically solved using simulated annealing. Karaoz *et al.* [13] use a similar approach but handle one function at a time. They apply a local search procedure in which for every vertex in turn (until convergence), the state of the vertex is changed according to the majority of the states of its neighbours. This procedure guarantees a solution with value at least half of the optimum. Nabieva *et al.* [14] apply the concept of functional flow which is propagated from an annotated protein to unannotated proteins. After simulating the spread over time of this functional flow through the network, each unannotated protein is assigned a score for having the function based on the amount of flow it received during the simulation. Relying on a Markovian assumption that the function of a protein is independent of all other proteins given the functions of its immediate neighbours, Deng *et al.* [15] adopt the Markov random field (MRF) model to simulate the protein interaction network with functional annotations, which fit the network and got good result. Letovsky and Kasif [16] also use an MRF model but with an assumption that the number of neighbours of a protein that are annotated with a given term is binomially distributed, where that distribution's parameter depends on whether the protein has that function or not. Lee *et al.* [17] develop a kernel logistic regression (KLR) method, which uses diffusion kernels and incorporated all indirect neighbours in the networks. While these approaches demonstrated that using machine learning and statistical methods can improve prediction performance, they bank on the same functional concept that the interaction partners of a protein are likely to share similar functions with it [11].

Although previous methods have proven to be useful to predict protein functions from PPI networks, they still suffer several limitations. Protein interaction data derived from the high-throughput techniques are typically very noisy. The data may include many false negatives (true interactions which remain undetected) and false positives (putative interactions that in fact do not occur). Sprinzak *et al.* [18] reported that the reliability of high-throughput yeast two-hybrid assays is about 50%. So how to filter the PPI data to overcome those noise is a big challenge.

In graph theory, a network can be represented as either a weighted graph or an unweighted graph. Previously a protein interaction network is normally represented as an unweighted graph, as it is easy to use and implement. However, as the PPI data includes a large number of unreliable interactions, the unweighted graph is far from optimal in representing the data. In this paper, we build a weighted graph model of protein interaction networks. Based on that, we propose a topological measurement to reflect our knowledge of small-world network property of the network to filter the protein interaction network and then get a more reliable protein interaction network. In fact, one protein may have multiple functions, which make it a typical multi-label problem. In a weighted graph, it is adequate to use artificial neural network (ANN) model to predict protein functions. In this paper, we investigate the problem of predicting protein functions from protein interaction data and make the following contributions:

- We analyze the reliability of connections in several protein interaction networks.
- We propose a novel topological measurement to calculate the interaction reliability between two proteins and filter the PPI networks.
- We propose an ANN-based method to predict the functions of proteins.

The remainder of the paper is organized as follows. In Section 2, we present our weighted graph model and topological measurement to rebuild protein interaction networks. In Section 3, we present our ANN-based prediction model. Extensive experimental results are reported in Section 4. The paper is concluded in Section 5.

2. Weighted Graph Model of Protein Interaction Networks

Many methods are based on the assumption that interacting proteins should share common functions. Table 1 shows the percentage of *function-relevant* interactions in three PPI data sets, namely, DIP, MIPS and BioGrid (see Result Section for detailed description). An interaction is considered to be *function-relevant* if the two proteins involved in the interaction have at least one function in common. In this test, we adopt FunCat(version 20070316) [19] in the MIPS database as our annotation categories. From Table 1, we can see that only 30–40% observed interactions are relevant in functions. In other words, most of the observed interactions do not share functions. Among those sharing function pairs, some of them share more functions than the others. Table 2 shows the percentage of *function-consistent* protein pairs which are observed to interact in the three data sets. Formally, we define two proteins P1 and P2 to be *funtion-consistent* if $|\frac{F(P_1)\cap F(P_2)}{F(P_1)\cup F(P_2)}| \geq \frac{1}{2}$, where $F(P_1)$ and $F(P_2)$ are functions of P_1 and P_2, respectively. As shown, only a small percentage

Table 1
The Percentage of Function-Relevant Interactions in Three Protein Interaction Data Sets

Data Set	Total Number of Interactions	Number of Functional-Relevant Interactions	Percentage
DIP	14,162	5,216	36.83
MIPS	13,877	4,189	30.18
BioGrids	117,675	36,446	30.97

Table 2
The Percentage of Function-Consistent Protein Pairs which Interact in Protein Interaction Data Sets

Data Set	Total Number of Interactions	Number of Functional-Consistent Interactions	Percentage
DIP	20,099	1,283	6.38
MIPS	21,795	898	4.12
BioGrids	21,499	2,718	12.64

of *function-consistent* protein pairs are observed to interact in the interaction data sets. These observations suggest two things: the protein interaction data may have many false interactions which need to be removed from protein interaction data and a weighted graph needs to be built to show the reliability between two proteins and to show the *functional similarity* between two proteins. As proteins with similar functions are likely to interact with each other in cells, we assume that the more reliable two proteins are, the more chance that they share common proteins.

We define a *weighted protein interaction network* [20] as follows: A weighted protein interaction network is a weighted undirected graph $G=(P,I,W)$, where P is a set of vertices, I is a set of edges between the vertices $(I \subseteq (u,v)|u,v \in P)$ and W is a function making each edge in I to a real value in the range of $[0 \ldots 1]$. Each vertex $v \in P$ in the graph represents a protein. Each edge $(u,v) \in I$ represents an interaction between proteins u and v. For each edge (u,v), $w(u,v)$ is the weight of (u,v) which represents the probability of this interaction being a true positive. Figure 1 shows our weighted protein interaction network model. The nodes represent the proteins, the edges between nodes represent the interactions between proteins and the numbers on the edges represent the weights between interacted proteins.

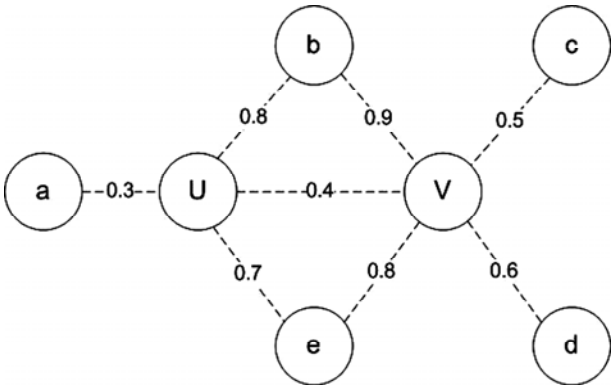


Figure 1. A weighted protein interaction network model. Node U and V are proteins. Node a, b, c, d, and e are the neighbours of U and V. The number on the edge between two nodes is the weight of the edge.

In this paper, we use the following additional terminologies: A *neighbour* of a vertex v is a vertex adjacent to v, also called *direct neighbour*. *Level-k neighbour* of vertex v is a vertex having k edges or steps to reach vertex v. The degree of a vertex v, denoted as $D(v)$, is the sum of weights of the edges connecting v: $D(v) = \sum_{(u,v) \in I} w(u,v)$. A *walk* is an alternating sequence of vertices and edges, with each edge being incident to the vertices immediately preceding and succeeding it in the sequence. A *path* is a walk with no repeated vertices.

Generally, there are two approaches to give a probability estimate for each interactions: We can use either the probability estimates of single interactions or the reliability estimates of interaction data sets.

Reliability estimates for single interactions are often achieved by incorporating known protein properties. These properties include paralogs (PVM) [21], protein domain information (DPV) [22] and the Bayesian integration of several information [23]. The probability estimate for any specific protein interaction is directly based on the domain knowledge of the proteins involved and therefore is intrinsically biased towards those proteins that we know well about.

Reliability of an interaction data set can be estimated by comparing the data set with reliable interaction data sets (usually those from small-scale experiments) [24, 25] or comparing the statistics of the data set with those of known reliable interaction data sets. The statistics include gene expression profile [21] and protein annotation [18]. Comparatively, as the reliability in this approach is estimated using the global statistics of the data set instead of any specific proteins, it is less biased towards any specific interactions in the data set. Therefore, we choose this approach for our initial estimate of probabilities.

We first combine several different protein interaction data sets $S = \{S_1, S_2, \ldots, S_n\}$, where each set S_i includes many interactions. If an interaction (u,v) appears only in one data set, we will set its probability as the reliability of this data set:

$$w(u,v) = r_k \quad \text{for each } (u,v) \in S_k \tag{1}$$

where r_k is the estimated reliability of the protein interaction data set S_k. The interaction (u,v) may appear in several data sets, i.e.:

$$(u,v) \in S_1 \cap S_2 \ldots \cap S_n \tag{2}$$

where $n > 1$. In this case, its probability is set to:

$$w(u,v) = 1 - (1 - r_1) \times (1 - r_2) \ldots \times (1 - r_n) \tag{3}$$

where r_i is the estimated reliability of S_i. This formula reflects the fact that interactions detected in multiple experiments are generally more reliable than those detected by only one experiment. Estimating the prior probability for each interaction in this manner represent our prior knowledge of the probability of interactions.

We then consider the topological features of PPIs. In a small-world protein interaction network [26], high clustering coefficient property predicates that proteins are likely to form dense clusters by interactions. Therefore true positive interactions in protein complexes and tightly coupled networks demonstrate dense interconnections. However, considering that there are many false positives in the data, we decide to measure the significance of two proteins's coexisting in a dense network as an indication of interaction reliability. In this paper, we consider all length k paths between two vertices and try to evaluate the significance of the paths. Then we combine the significance measurements for all different ks into our final topological measurement.

Definition 1. *The PathStrength of a path p, denoted as $PS(p)$, is the product of the weights of all the edges on the path, i.e.*:

$$PS(p) = \prod_{i=1}^{l} w(v_{i-1}, v_i)$$

for path $p = <v_0, v_1, \ldots, v_l>$.

The PathStrength of a path indicates the probability that a walk on the path can reach its ending vertex.

Definition 2. *The k-length PathStrength between two vertices A and B, denoted as $PS^k(A, B)$, is the sum of the PathStrength of all k-length paths between vertices A and B, i.e.*:

$$PS^k(A, B) = \sum_{p=<v_0=A, v_1, \ldots, v_k=B>} PS(p)$$

By summing upon all these paths, the k-length PathStrength between two vertices reflects the strength of connections between these two vertices by a k-step walk.

Definition 3. *The k-length MaxPathStrength between two vertices A and B, denoted as $MaxPS^k(A, B)$, is defined as*:

$$MaxPS^k(A, B) = \begin{cases} \sqrt{D(A) \times D(B)} & \text{if } k = 2 \\ D(A) \times D(B) & \text{if } k = 3 \\ \sum_{P_i \in N(A), P_j \in N(B)} MaxPS^{k-2}(P_i, P_j) & \text{if } k > 3 \end{cases}$$

MaxPathStrength measures the maximum possible PathStrength between two vertices. As we consider only $PS^k(A, B)$ for $k > 1$, we define $MaxPS^k(A, B)$ only for $k > 1$ case. By dividing the PathStrength by this maximum possible value, we get the significance measurement of k-length paths.

Definition 4. *The k-length PathRatio between two vertices A and B, denoted as $PR^k(A, B)$, is the ratio of the k-length PathStrength to the k-length MaxPathStrength between two vertices A and B, i.e.*:

$$PR^k(A, B) = \frac{PS^k(A, B)}{MaxPS^k(A, B)}$$

We sum this measurement on all different lengths and get our final topological measurement:

Definition 5. *The PathRatio between two vertices A and B, denotes as PR(A,B), is the sum of k-length PathRatios between A and B for all possible $k > 1$, i.e.:*

$$PR(A, B) = \sum_{k=2}^{|P|-2} PR^k(A, B)$$

where $|P|$ is the number of vertices in the graph.

The value of PathRatio reflects the reliability between two proteins in the network. In a protein interaction network, the closer two proteins, the more chances they should share common functions. Table 3 shows clearly that there are quite strong functional influence between level-1 and level-2 neighbours, still some influence for level-3 neighbours, but weaker influence for neighbours higher than 3 in the yeast protein interactions. So in this paper, when we calculate the PathRatio between two vertices, we just calculate the PathStrength up to the third level. The bigger the value of PathRatio, the more reliable these two proteins are, i.e., the higher probability that these two

Table 3
Fraction of Annotated Yeast Proteins That Share Function With (1) Level-1 neighbours exclusively; (2) Level-2 neighbours exclusively; (3) Level-3 neighbours exclusively; (4) Level-4 neighbours exclusively

Shared Functions With	Number of Corresponding Neighbours	Number of Sharing Common Functions	Fraction [%]
Level-1 neighbours exclusively	4,812	1,136	23.61
Level-2 neighbours exclusively	203,574	40,275	19.78
Level-3 neighbours exclusively	1,381,525	185,182	13.45
Level-4 neighbours exclusively	913,742	49,068	5.17

proteins share common proteins. In this way, we changed this weighted graph to a *functional similarity* interaction network.

Figure 2 shows that our assumption works well for some simple protein function prediction methods, such as *neighbour counting* method [9]. In this test, function prediction performance from the weighted interaction network was assessed comparing to an unweighted interaction network. The result shows how significantly the weighted graph can improve the prediction performance.

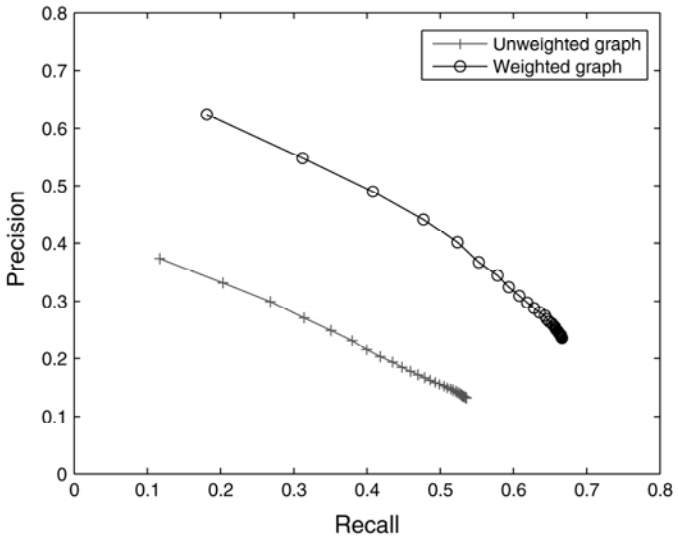


Figure 2. Comparison of neighbour counting method with unweighed and weighted protein interaction graphs.

This new interaction network is different from the former one in the following ways:

- The neighbours of one annotated protein include both direct neighbours and indirect neighbours up to level-3.
- These functional similarity based weighted networks have the weight for each pair of neighbours, which indicates the chance that they may have similar functions.

3. Function prediction algorithm

Typically one protein can have multiple functions, so we transfer function prediction problem into a typical multi-label problem with functions as labels and proteins as instances or items. Recently, the issue of learning from multi-label data has attracted significant attention from a lot of researchers in the area of machine learning and pattern recognition. Many existing problems can be classified into this category, such as semantic annotation of images [27] and video [28], music categorization into emotions [29] and directed marketing [30]. Traditionally, instances or items are independent and don't connect with each other, so many probabilistic methods can be used in this scenario. But in our case, as proteins are connected and the protein interaction network has been proved to have small-world properties [26], it automatically leads us to use ANN to solve this problem and use the PathRatio as the weight W and the neighbours as the nodes. ANN is a computational model based on biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases, an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Here we escape the adaptive part and only use a simple model called *perceptron*. In the network we built from the last section, every annotated protein should have a list of variables indicating if this protein has any particular functions. For example, if there are 4 functions $\{f1, f2, f3, f4\}$, and protein p has $\{f1, f2, f4\}$, the function vector v of protein p will be $(1, 1, 0, 1)$. Then for an unannotated protein u, the set of possible functions f it may have can be predicted using the following formula:

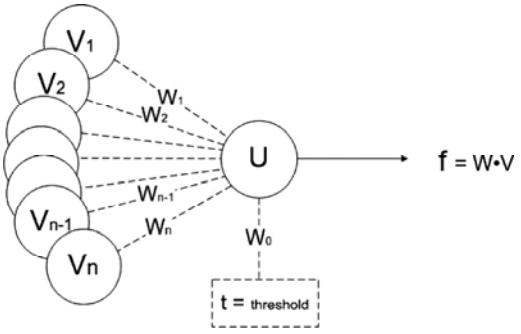


Figure 3. ANN-based function prediction model. U is the unannotated protein, $V_{1\sim n}$ are U's neighbours, $W_{1\sim n}$ are the PathRatio value of the edge, t is the user defined threshold, f is the binary vector indicating the functions predicted for U.

$$\begin{aligned}\hat{f} &= sign(\mathbf{w} \times \mathbf{v}) \\ &= sign[w_d v_d + w_{d-1} v_{d-1} + \ldots + w_1 v_1 + w_0 v_0]\end{aligned} \tag{4}$$

where $w_0 = -t$, $v_0 = 1$, $\mathbf{w} \times \mathbf{v}$ is a dot product between the weight vector $\mathbf{w}$ and the input attribute matrix $\mathbf{v}$, t is the threshold value to be set, v_i is the function vector of the neighbour i and w_i is the *functional similarity* weight between neighbour i and protein u, where i is the index of the neighbours of protein u, and d is the number of neighbours that the unannotated proteins has. Function *sign* outputs 1 if $\mathbf{w} \times \mathbf{v}$ is bigger than 0, otherwise, it outputs -1. The result $\mathbf{f}$ is a vector indicating if protein u has the corresponding function. This ANN based model is shown in Fig. 3. The proposed ANN model unites the functional information from the neighbours of the unannotated protein and assigns those information with different weights which represent the functional similarities between the neighbours and the unannotated protein. As this model fully uses the neighbours up to level-3 and treat different neighbours separately, it overcomes weak points of previous methods, such as narrow neighbourhood and equal effection from neighbours. The flowchart of the proposed algorithm is shown in Fig. 4 and it is divided into four major steps: input and initialization, building weighted network, function prediction model, result and evaluation. The detail of each step is explained as follows:

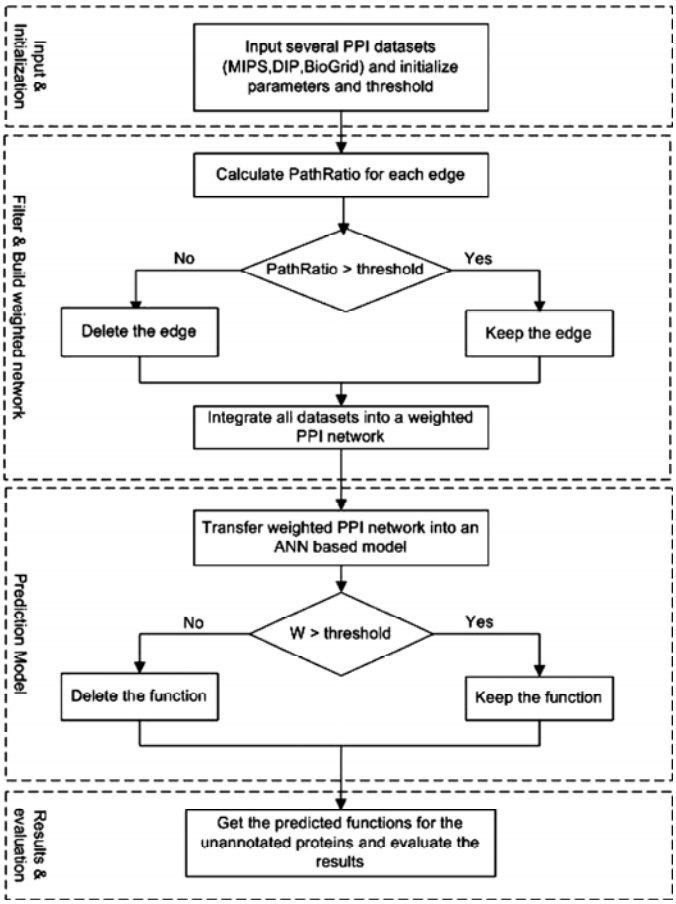


Figure 4. A flowchart of the proposed function prediction algorithm.

1. *Input and initialization*: Input DIP, MIPS and BioGrid PPI data sets and initialize the threshold of the weight for the edge.
2. *Filtering and building weighted network*: Use the formula of PathRatio to calculate the weight of each edge and keep the edges which have bigger weight than the threshold. Then integrate all interactions into one weighted network.
3. *Prediction model*: Transfer the weighted network into an ANN-based model. For each unannotated protein, keep the functions which have bigger value than the threshold.
4. *Result and evaluation*: Get the predicted functions for each unannotated protein and evaluate the result with leave-one-out cross-validation.

4. Experimental Results

4.1 Cross-Validation of Function Prediction

For our experiments, we built protein interaction networks from three different yeast interaction networks. The first one is MIPS data set [19], which contains 3,882 proteins and 13,877 interactions. The second one is BioGrid data set [31], which contains 4,265 proteins and 117,675 interactions. The third one is DIP data set [32], which contains 4,935 proteins and 14,162 interactions.

To evaluate the effectiveness of our method, we used FunCat as the functional annotations from MIPS database [19]. The scheme of FunCat is a tree-shaped hierarchical structure. To avoid overly specific annotations, we cut the scheme at the third level and obtained 259 functional categories.

We assessed the performance of our function prediction approach by the leave-one-out cross-validation method [33]. For each protein in annotations, we assumed it is unannotated and predicted its function using its interaction information and the annotations of the other proteins. Then we compared the predicted functions with the true annotations. Let n_i be the number of annotated functions for protein P_i, m_i be the number of predicted functions for P_i, k_i be the size of common functions of m_i and n_i and n is the total number of distinct proteins with annotations. Precision and recall are then calculated as:

$$Recall = \frac{\sum_1^n k_i}{\sum_1^n n_i} \tag{5}$$

and

$$Precision = \frac{\sum_1^n k_i}{\sum_1^n m_i} \tag{6}$$

When we implemented our proposed method, first we integrated MIPS, DIP and BioGrid protein interaction data sets using (3), and the reliability of each data set was estimated by EPR (expression profile reliability) index [21]: 0.85 for DIP, 0.73 for MIPS, 0.81 for BIOGRID. Then we rebuilt the weighted graph with weight threshold 0.2, which means we only keep interactions whose weight is above 0.2. At this point, we achieved an interaction graph. Then we used the prediction algorithm we proposed above to predict the functions of each protein in the data set.

Figure 5 shows the precision and recall plots with respect to the threshold of prediction confidence, which is a user-dependent parameter in our algorithm. When we use 3.8 as the threshold of prediction confidence, our algorithm predicts fewer functions for each protein, but most of the functions are correctly predicted comparing to the actual annotations, and the precision for this threshold is close to 0.9. As a lower threshold is used, recall increases while precision decreases monotonically. Approximately, when the recall is 0.2 and 0.4, we had the precision of 0.8 and 0.6, respectively.

4.2 Comparison with Other Approaches

We evaluated the performance of our ANN method with two previous approaches: the neighbour-counting method [9] and the indirect-neighbour method [11].

Indirect-neighbour method [11] computes the likelihood that an unknown protein p has a function using the functional similarity weights between p and direct and indirect neighbours. The functional similarity weight of two proteins is calculated by the commonality of their neighbours in the protein interaction network. We used a threshold of the likelihood to generate the output set of

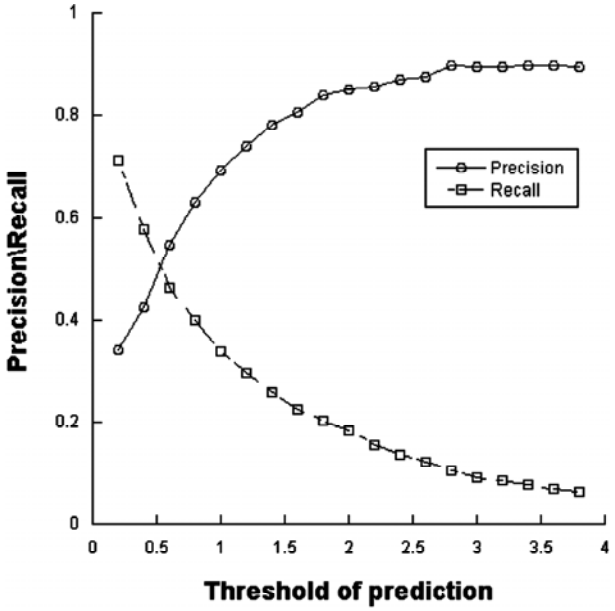


Figure 5. Precision and recall plots by cross-validation for protein function prediction. The performance of our function prediction algorithm was assessed by the leave-one-out cross-validation using the proteins that appear in the interaction data from DIP and are annotated on the functional categories in MIPS. As a higher threshold of prediction confidence is used, precison increases whereas recall decreases.

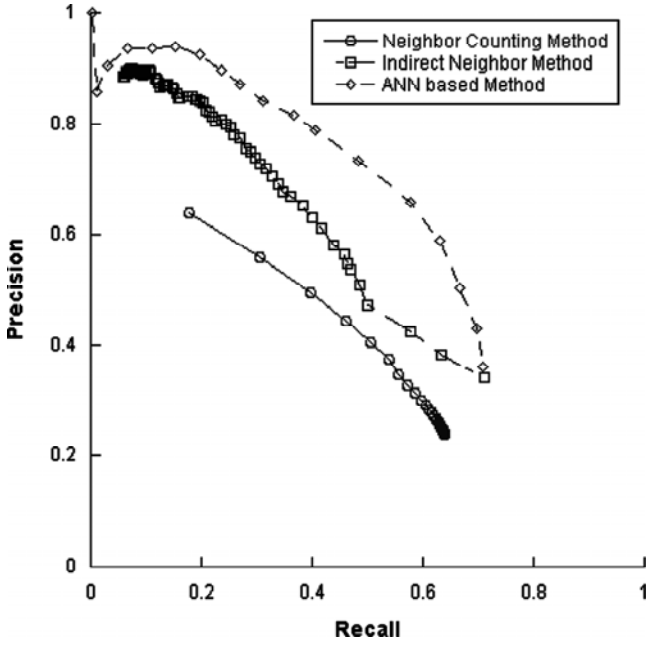


Figure 6. The precision–recall relationships of our ANN-based method are compared with two competing methods: indirect neighbours methods and neighbour counting methods. For any recall value, our approach substantially outperformed the other two methods.

predicted functions for each protein. We then obtained different output sets by various thresholds. Neighbour-counting method computes the frequency of each function among the direct neighbours of protein p and then sorts it to get the top k functions.

Figure 6 shows the precision and recall of the three approaches on the filtered data set. Our ANN-based method remarkably outperforms the neighbour-counting method, as neighbour-counting method only considered the direct neighbours and missed lots of functional information from other proteins in the protein interaction network. Our approach is slightly better than indirect-neighbour method when recall is between 0 and 0.2, but when recall is bigger than 0.2, our method has the precison of more than 0.1 higher than the indirect-neighbour method. This result indicates three things: (1) fully understanding the small-world property of the protein interaction network is very important to predict the functions of proteins. (2) the more functional information you use to predict unknown proteins, the better result you may get, and (3) a weighted graph is more suitable to represent protein interaction networks than unweighted graph to predict the functions of proteins, and the more reliable the graph is, the more accurate the result will be.

It is worth mentioning that since building a weighted graph and function prediction are completely independent, different approaches can be adopted for these two steps, such as using IRAP [34] or IG2 [35] to build the weighted graph and using KNN or other machine learning methods to predict the functions of proteins.

4.3 Discussion

Through recent advances of high-throughput techniques, a significant amount of protein interaction data has been accumulated. Protein functions have been predicted from the interaction data because the evidence of interactions can be interpreted as functional links. However, we observe that only a small fraction of current interaction data from major interaction databases are related to functional linkage. The results indicate that more than 60% of interacting protein pairs are not linked by similar functions. In other words, at most 40% of protein pairs have been motivated by similar functions. This observation has been also demonstrated by the limited accuracy of previous function prediction methods.

Our method uses the small-world property of protein interaction networks and derives functional information from both direct neighbours and up to level-3 neighbours, which is more comprehensive than just using direct neighbours and neighbourhood information. Also using a weighted interaction network is more suitable than using a unweighted network as different neighbours have different contributions to the functions of unknown proteins.

In our experiments, function prediction has been conducted with yeast PPI data. However, our ANN-based framework can be well-applicable to higher-level organisims because of its efficiency.

5. Conclusion

Functional characterization of proteome is a central goal in the field of bioinformatics. The experimentally determined

protein interactions are crucial data sources to uncover the functional knowledge of uncharacterized proteins. However, a pre-process to access the functional linkage of interacting proteins from current interaction data is required for predicting protein function successfully.

In this paper, we presented an ANN-based method to integrate direct neighbours, level-2 neighbours and level-3 neighbours based on a weighted protein interaction network to predict the functions of proteins. Our results imply that function prediction from protein interaction networks using a weighted network is a promising way, and integrating more data sets and more protein function related information may achieve better results. This is also our future research for functional knowledge discovery.

Acknowledgement

This work was partly supported by NSF grant DBI-0234895.

References

[1] W.R. Pearson & D.J. Lipman, Improved tools for biological sequence comparison, *Proceedings of the National Academy of Sciences of the United States of America*, *85*(8), 1988, 2444–2448.

[2] D.J. Lockhart, H. Dong, M.C. Byrne, M.T. Follettie, M.V. Gallo, M.S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, & E.L. Brown, Expression monitoring by hybridization to high-density oligonucleotide arrays, *Nature Biotechnology*, *14*(13), 1996, 1675–1680.

[3] E.M. Marcotte, M. Pellegrini, H.L. Ng, D.W. Rice, T.O. Yeates, & D. Eisenberg, Detecting protein function and protein–protein interactions from genome sequences, *Science*, *285*(5428), 1999, 751–753.

[4] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, & Y. Sakaki, A comprehensive two-hybrid analysis to explore the yeast protein interactome, *Proceedings of the National Academy of Sciences of the United States of America*, *98*(8), 2001, 4569–4574.

[5] P. Uetz, L. Giot, G. Cagney, T.A. Mansfield, R.S. Judson, J.R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields, & J.M. Rothberg, A comprehensive analysis of protein–protein interactions in *Saccharomyces cerevisiae*, *Nature*, *403*(6770), 2000, 623–627.

[6] A.C. Gavin, M. Bosche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J.M. Rick, A.M. Michon, C.M. Cruciat, M. Remor, C. Hofert, M. Schelder, M. Brajenovic, H. Ruffner, A. Merino, K. Klein, M. Hudak, D. Dickson, T. Rudi, V. Gnau, A. Bauch, S. Bastuck, B. Huhse, C. Leutwein, M.A. Heurtier, R.R. Copley, A. Edelmann, E. Querfurth, V. Rybin, G. Drewes, M. Raida, T. Bouwmeester, P. Bork, B. Seraphin, B. Kuster, G. Neubauer, & G. Superti-Furga, Functional organization of the yeast proteome by systematic analysis of protein complexes, *Nature*, *415*(6868), 2002, 141–147.

[7] Y. Ho, A. Gruhler, A. Heilbut, G.D. Bader, L. Moore, S.L. Adams, A. Millar, P. Taylor, K. Bennett, K. Boutilier, L. Yang, C. Wolting, I. Donaldson, S. Schandorff, J. Shewnarane, M. Vo, J. Taggart, M. Goudreault, B. Muskat, C. Alfarano, D. Dewar, Z. Lin, K. Michalickova, A.R. Willems, H. Sassi, P.A. Nielsen, K.J. Rasmussen, J.R. Andersen, L.E. Johansen, L.H. Hansen, H. Jespersen, A. Podtelejnikov, E. Nielsen, J. Crawford, V. Poulsen, B.D. Sorensen, J. Matthiesen, R.C. Hendrickson, F. Gleeson, T. Pawson, M.F. Moran, D. Durocher, M. Mann, C.W. Hogue, D. Figeys, & M. Tyers, Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry, *Nature*, *415*(6868), 2002, 180–183.

[8] A.H. Tong, G. Lesage, G.D. Bader, H. Ding, H. Xu, X. Xin, J. Young, G.F. Berriz, R.L. Brost, M. Chang, Y. Chen, X. Cheng, G. Chua, H. Friesen, D.S. Goldberg, J. Haynes, C. Humphries, G. He, S. Hussein, L. Ke, N. Krogan, Z. Li, J.N. Levinson, H. Lu, P. Menard, C. Munyana, A.B. Parsons, O. Ryan, R. Tonikian, T. Roberts, A.M. Sdicu, J. Shapiro, B. Sheikh, B. Suter, S.L. Wong, L.V. Zhang, H. Zhu, C.G. Burd, S. Munro, C. Sander, J. Rine, J. Greenblatt, M. Peter, A. Bretscher, G. Bell, F.P. Roth, G.W. Brown, B. Andrews, H. Bussey, & C. Boone, Global mapping of the yeast genetic interaction network, *Science*, *303*(5659), 2004, 808–813.

[9] B. Schwikowski, P. Uetz, & S. Fields, A network of protein–protein interactions in yeast, *Nature Biotechnology*, *18*(12), 2000, 1257–1261.

[10] H. Hishigaki, K. Nakai, T. Ono, A. Tanigami, & T. Takagi, Assessment of prediction accuracy of protein function from protein–protein interaction data, *Yeast*, *18*(6), 2001, 523–531.

[11] H.N. Chua, W.K. Sung, & L. Wong, Exploiting indirect neighbours and topological weight to predict protein function from protein–protein interactions, *Bioinformatics*, *22*(13), 2006, 1623–1630.

[12] A. Vazquez, A. Flammini, A. Maritan, & A. Vespignani, Global protein function prediction from protein–protein interaction networks, *Nature Biotechnology*, *21*(6), 2003, 697–700.

[13] U. Karaoz, T.M. Murali, S. Letovsky, Y. Zheng, C. Ding, C.R. Cantor, & S. Kasif, Whole-genome annotation by using evidence integration in functional-linkage networks, *Proceedings of the National Academy Sciences of the United States of America*, *101*(9), 2004, 2888–2893.

[14] E. Nabieva, K. Jim, A. Agarwal, B. Chazelle, & M. Singh, Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps, *Bioinformatics*, *21*(Suppl 1) 2005, i302–i310.

[15] M. Deng, K. Zhang, S. Mehta, T. Chen, & F. Sun, Prediction of protein function using protein–protein interaction data, *Journal of Computational Biology*, *10*(6), 2003, 947–960.

[16] S. Letovsky & S. Kasif, Predicting protein function from protein/protein interaction data: A probabilistic approach, *Bioinformatics*, *19*(Suppl 1) 2003, i197–i204.

[17] H. Lee, Z. Tu, M. Deng, F. Sun, & T. Chen, Diffusion kernel-based logistic regression models for protein function prediction, *OMICS*, *10*(1), 2006, 40–55.

[18] E. Sprinzak, S. Sattath, & H. Margalit, How reliable are experimental protein–protein interaction data? *Journal of Molecular Biology*, *327*(5), 2003, 919–923.

[19] H.W. Mewes, D. Frishman, K.F. Mayer, M. Munsterkotter, O. Noubibou, P. Pagel, T. Rattei, M. Oesterheld, A. Ruepp, & V. Stumpflen, MIPS: Analysis and annotation of proteins from whole genomes in 2005, *Nucleic Acids Research*, *34*(Database issue), 2006, D169–D172.

[20] P. Pei & A. Zhang, A topological measurement for weighted protein interaction network, *Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference*, 2005, 268–278.

[21] C.M. Deane, L. Salwinski, I. Xenarios, & D. Eisenberg, Protein interactions: Two methods for assessment of the reliability of high throughput observations, *Molecular and Cellular Proteomics*, *1*(5), 2002, 349–356.

[22] I. Xenarios, L. Salwinski, X.J. Duan, P. Higney, S.M. Kim, & D. Eisenberg, DIP, the database of interacting proteins: A research tool for studying cellular networks of protein interactions, *Nucleic Acids Research*, *30*(1), 2002, 303–305.

[23] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N.J. Krogan, S. Chung, A. Emili, M. Snyder, J.F. Greenblatt, & M. Gerstein, A Bayesian networks approach for predicting protein–protein interactions from genomic data, *Science*, *302*(5644), 2003, 449–453.

[24] C. von Mering, R. Krause, B. Snel, M. Cornell, S.G. Oliver, S. Fields, & P. Bork, Comparative assessment of large-scale data sets of protein–protein interactions, *Nature*, *417*(6887), 2002, 399–403.

[25] G.D. Bader & C.W. Hogue, Analyzing yeast protein–protein interaction data obtained from different sources, *Nature Biotechnology*, *20*(10), 2002, 991–997.

[26] D.J. Watts & S.H. Strogatz, Collective dynamics of "small-world" networks, *Nature*, *393*(6684), 1998, 440–442.

[27] M.R. Boutell, J.B. Luo, X.P. Shen, & C.M. Brown, Learning multi-label scene classification, *Pattern Recognition*, *37*(9), 2004, 1757–1771.

[28] G.J. Qi, X.S. Hua, Y. Rui, J. Tang, T. Mei, & H.-J. Zhang, Correlative multi-label video annotation, *Proceedings of the 15th International Conference on Multimedia*, Augsburg, Germany, 2007, 17–26.

[29] T. Li & M. Ogihara, Detecting emotion in music, *Proceedings of the International Symposium on Music Information Retrieval*, Washington DC, USA, 2003, 239–240.

[30] Y. Zhang, S. Burer, & W.N. Street, Ensemble pruning via semi-definite Programming, *Journal of Machine Learning Research*, *7*, 2006, 1315–1338.

[31] C. Stark, B.J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, & M. Tyers, BioGRID: A general repository for interaction datasets, *Nucleic Acids Research*, *34*(Database issue), 2006, D535–D539.

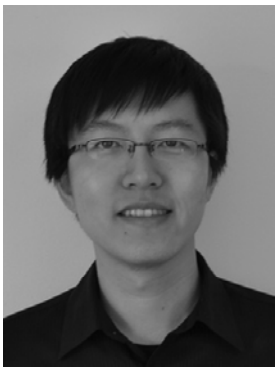
[32] L. Salwinski, C.S. Miller, A.J. Smith, F.K. Pettit, J.U. Bowie, & D. Eisenberg, The database of interacting proteins: 2004 update, *Nucleic Acids Research*, *32*(Database issue), 2004, D449–D451.

[33] A. Zhang, *Protein interaction networks: Computional analysis* (Cambridge university, 2009).

[34] J. Chen, W. Hsu, M.L. Lee, & S.-K. Ng, Systematic assessment of high-throughput experimental data for reliable protein interactions using network topology, *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, 2004, 368–372.

[35] R. Saito, H. Suzuki, & Y. Hayashizaki, Construction of reliable protein–protein interaction networks with a new interaction generality measure, *Bioinformatics*, *19*(6), 2003, 756–763.

Biographies



Lei Shi received his B.S. and M.S. degrees in Electronic Information from Sichuan University in 2003 and 2006. Currently, he is a Ph.D. candidate in the Department of Computer Science and Engineering, State University of New York at Buffalo. His research interests include bioinformatics, machine learning, data-base system and data mining.

Young-Rae Cho is currently an Assistant Professor in the Department of Computer Science at Baylor University. He received his Ph.D. degree in Computer Science and Engineering at State University of New York at Buffalo in 2009, and his Master's degree in Computer Science at University of Illinois at Urbana-Champaign in 2003. His current research interests include bioinformatics, data mining and computational systems biology. He is an author of over 15 publications in journals, conferences and book chapters.

Dr. Aidong Zhang is Professor and Chair in the Department of Computer Science and Engineering at State University of New York at Buffalo. Her research interests include bioinformatics, data mining, multimedia and database systems and content-based image retrieval. She is an author of over 200 research publications in these areas. She has chaired or served on over 100 program committees of international conferences and workshops, and currently serves several journal editorial boards. She has published two books "*Protein Interaction Networks: Computational Analysis*" (Cambridge University Press, 2009) and "*Advanced Analysis of Gene Expression Microarray Data*" (World Scientific Publishing Co., Inc. 2006). Dr. Zhang is a recipient of the National Science Foundation CAREER award and State University of New York (SUNY) Chancellor's Research Recognition award.

Dr. Zhang is an IEEE Fellow.