

Function Prediction Using Neighborhood Patterns*

Petko Bogdanov[†]

Department of Computer Science, University of
California, Santa Barbara, CA 93106
petko@cs.ucsb.edu

Ambuj Singh

Department of Computer Science, University of
California, Santa Barbara, CA 93106
ambuj@cs.ucsb.edu

ABSTRACT

The recent advent of high throughput methods has generated large amounts of protein interaction data. This has allowed the construction of genome-wide networks. A significant number of proteins in such networks remain uncharacterized and predicting the function of these proteins remains a major challenge. A number of existing techniques assume that proteins with similar functions are topologically close in the network. Our hypothesis is that proteins with similar functions observe similar annotation patterns in their neighborhood, regardless of the distance between them in the interaction network. We thus predict functions of uncharacterized proteins by comparing their functional neighborhoods to proteins of known function. We propose a two-phase approach. First we extract functional neighborhood features of a protein using *Random Walks with Restarts*. We then employ a kNN classifier to predict the function of uncharacterized proteins based on the computed neighborhood features. We perform leave-one-out validation experiments on two *S. cerevisiae* interaction networks revealing significant improvements over previous techniques. Our technique also provides a natural control of the trade-off between accuracy and coverage of prediction.

Categories and Subject Descriptors

I.5 [Pattern Recognition]: Applications

General Terms

Methodology

*Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

BIOKDD '08, August 24, 2008, Las Vegas, Nevada, USA.
Copyright 2007 ACM 978-1-60558-302-0 ...\$5.00.

[†]Corresponding author.

Keywords

Protein Function Prediction, Feature Extraction, Classification, Protein Interaction Network

1. INTRODUCTION

The rapid development of genomics and proteomics has generated an unprecedented amount of data for multiple model organisms. As has been commonly realized, the acquisition of data is but a preliminary step, and a true challenge lies in developing effective means to analyze such data and endow them with physical or functional meaning [24]. The problem of function prediction of newly discovered genes has traditionally been approached using sequence/structure homology coupled with manual verification in the wet lab. The first step, referred to as computational function prediction, facilitates the functional annotation by directing the experimental design to a narrow set of possible annotations for unstudied proteins.

Significant amount of data used for computational function prediction is produced by high-throughput techniques. Methods like Microarray co-expression analysis and Yeast2Hybrid experiments have allowed the construction of large interaction networks. A protein interaction network (PIN) consists of nodes representing proteins, and edges representing interactions between proteins. Such networks are stochastic as edges are weighted with the probability of interaction. There is more information in a PIN compared to sequence or structure alone. A network provides a global view of the context of each gene/protein. Hence, the next stage of computational function prediction is characterized by the use of a protein's interaction context within the network to predict its functions.

A node in a PIN is annotated with one or more functional terms. Multiple and sometimes unrelated annotations can occur due to multiple active binding sites or possibly multiple stable tertiary conformations of a protein. The annotation terms are commonly based on an ontology. A major effort in this direction is the Gene Ontology (GO) project [11]. GO characterizes proteins in three major aspects: *molecular function*, *biological process* and *cellular localization*. Molecular functions describe activities performed by individual gene products and sometimes by a group of gene products. Biological processes organize groups of interactions into "ordered assemblies." They are easier to predict since they localize in the network. In this paper, we seek to predict the GO molecular functions for uncharacterized (target) proteins.

The main idea behind our function prediction technique is that function inference using only local network analysis but without the examination of global patterns is not general enough to cover all

possible annotation trends that emerge in a PIN. Accordingly, we divide the task of prediction into the following sequence of steps: extraction of neighborhood features, accumulation and categorization of the neighborhood features from the entire network, and prediction of the function of a target protein based on a classifier. We summarize the neighborhood of a protein using *Random Walks with Restarts*. Coupled with annotations on proteins, this allows the extraction of histograms (on annotations) that serve as our features. We perform a comprehensive set of experiments that reveal a significant improvement of prediction accuracy compared to existing techniques.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 presents our methods. In Section 4, we present experimental results on two *S. cerevisiae* interaction networks, and conclude in Section 5.

2. RELATED WORK

According to a recent survey [22], most existing network-based function prediction methods can be classified in two groups: *module assisted* and *direct methods*. Module assisted methods detect network modules and then perform a module-wide annotation enrichment [16]. The methods in this group differ in the manner they identify modules. Some use graph clustering [23, 10] while others use hierarchical clustering based on network distance [16, 2, 4], common interactors [20] and Markov random fields [15].

Direct methods assume that neighboring proteins in the network have similar functional annotations. The *Majority* method [21] predicts the three prevailing annotations among the direct interactors of a target protein. This idea has later been generalized to higher levels in the network [13]. Another approach, *Indirect Neighbor* [7], distinguishes between direct and indirect functional associations, considering level 1 and level 2 associations. The *Functional Flow* method [19] simulates a network flow of annotations from annotated proteins to target ones. Karaoz et al. [14] propose an annotation technique that maximizes edges between proteins with the same function.

A common drawback of both the direct and module-assisted methods is their hypothesis that proteins with similar functions are always topologically close in the network. As we show, not all proteins in actual protein networks corroborate this hypothesis. The direct methods are further limited to utilize information about neighbors up to a certain level. Thus, they are unable to predict the functions of proteins surrounded by unannotated interaction partners.

A recent approach by Barutcuoglu et al. [3] formulates the function prediction as a classification problem with classes from the GO biological process hierarchy. The authors build a Bayesian framework to combine the scores from multiple Support Vector Machine (SVM) classifiers.

A technique called *LaMoFinder* [6] predicts annotations based on network motifs. An unannotated network is first mined for conserved and unique structural patterns called motifs. The motifs are next labeled with functions. Pairs of corresponding proteins in different motif occurrences are expected to have similar annotations. The method is restricted to target proteins that are part of unique and frequent structural motifs. A less conservative approach for pattern extraction (that is robust to noise in network topology) is needed for the task of whole genome annotation.

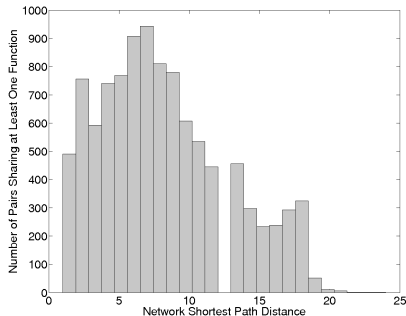


Figure 1: Proteins sharing annotations do not always interact in the Filtered Yeast Interactome (FYI) [12]. Similar functions are sometimes at large network distances.

We hypothesize that the simultaneous activity of sometimes functionally diverse functional agents comprise higher level processes in different regions of the PIN. We refer to this hypothesis as *Similar Neighborhood*, and to the central idea in all direct methods as *Function Clustering*. Our hypothesis is more general, since a clique of similar function proteins can be equivalently treated as a set of nodes that observe the same functional neighborhood. Hence *Similar Neighborhood* is a natural generalization of *Function Clustering*. A justification for our approach is provided by Figure 1 which shows that proteins of similar function may occur at large network distances.

3. METHOD

Our approach divides function prediction into two steps: extraction of neighborhood features, and prediction based on the features. According to our *Similar Neighborhood* hypothesis, we summarize the functional network context of a target protein in the neighborhood features extraction step. We compute the steady state distribution of a *Random Walk with Restarts (RWR)* from the protein. The steady state is then transformed into a functional profile. In the second step, we employ a *k-Nearest-Neighbors (kNN)* classifier to predict the function of a target protein based on its functional profile. As confirmed by the experimental results, the desired trade-off between accuracy of prediction and coverage of our algorithm can be controlled by k , the only parameter of the kNN classification scheme. Such a decoupled approach allows for the possibility that other kinds of neighborhood features can be extracted, and that other kinds of classifiers can be used.

3.1 Extraction of functional profiles

The extraction of features is performed in two steps. First, we characterize the neighborhood of a target node with respect to all other nodes in the network. Second, we transform this node-based characterization to a function-based one.

We summarize a protein’s neighborhood by computing the steady state distribution of a *Random Walk with Restarts (RWR)*. We simulate the trajectory of a random walker that starts from the target protein and moves to its neighbors with a probability proportional to the weight of each connecting edge. We keep the random walker close to the original node in order to explore its local neighborhood, by allowing transitions to the original node with a probability of r , the restart probability [5].

The PIN graph is represented by its adjacency matrix $M_{n,n}$. Each element $m_{i,j}$ of M encodes the probability of interaction between proteins i and j . The outgoing edge probabilities of a each protein are normalized, i.e. M is row-normalized. We use the power method to compute the steady state vector with respect to each node. We term the steady state distribution of node j as the *neighborhood profile* of protein j , and denote it as $S^j, j \in [1, n]$. The neighborhood profile is a vector of probabilities $S_i^j, i \neq j, i, j \in [1, n]$. Component S_i^j is proportional to the frequency of visits to node i in the RWR from j . More formally, the power method is defined as follows:

$$S^j(t+1) = (1-r)M^T S^j(t) + rX. \quad (1)$$

In the above equation, X is a size- n vector defining the initial state of the random walk. In the above scenario, X has only one non-zero element corresponding to the target node. $S^j(t)$ is the neighborhood profile after t time steps. The final neighborhood profile is the vector S^j when the process converges. A possible interpretation of the neighborhood profile is an affinity vector of the target node to all other nodes based solely on the network structure.

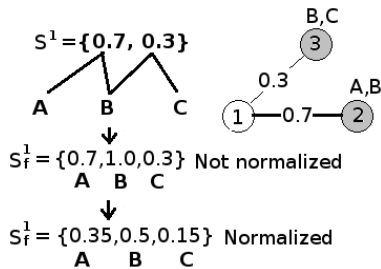


Figure 2: Transformation of the neighborhood profile of node 1 into a functional profile. Node 2 is annotated with functions A and B and node 3 is annotated with functions B and C. The neighborhood profile of node 1 is computed and transformed using the annotations on the nodes into a functional profile.

As our goal is to capture the functional context of a protein, the next step in our feature extraction is the transformation of a neighborhood profile into a functional profile. The value S_i^j of node j to node i can be treated as affinity to the annotations of i . Figure 2 illustrates the transformation of a neighborhood profile to a functional profile. Assume that RWR performed from node 1 results in the neighborhood profile (0.7, 0.3), where 0.7 corresponds to node 2, and 0.3 to node 3. Annotations on these two nodes are weighted by the corresponding values, resulting in the vector (0.7, 1.0, 0.3) over functions A, B, and C, respectively. This vector is then normalized, resulting into the functional profile (0.35, 0.5, 0.15).

More formally, based on the annotations of a protein, we define an annotation flag e_{ia} that equals 1 if protein i is annotated with function a and 0 otherwise. The affinity to each function a in the neighborhood profile is computed as:

$$S_f^j(a) = \sum_{i=1, i \neq j}^n S_i^j e_{ia}. \quad (2)$$

Vector S_f^j is normalized to yield the functional profile for node j .

3.2 Function prediction by nearest neighbor classification

The second step in our approach is predicting the annotations of a given protein based on its *functional profile*. According to our *Similar Neighborhood* hypothesis, proteins with similar functional profiles are expected to have similar annotations. An intuitive approach in this setting is to annotate a target protein with the annotations of the protein with most similar neighborhood. Alternatively, we can explore the top k similar proteins to a target protein and compute a consensus set of candidate functions.

We formulate function prediction as a multi-class classification problem. Each protein’s profile is an instance (feature vector). Each instance can belong to one or more classes as some proteins have multiple functions. We choose a distance based classification approach to the problem, namely the k-Nearest-Neighbor (kNN) classifier. The classifier uses the L1 distance between the instances and classifies an instance based on the distributions of classes in its k nearest L1 neighbors.

The consensus set of predicted labels is computed using weighted voting. Annotations of a more similar neighborhood are weighted higher. The result is a set of scores for each function where a function’s score is computed as follows:

$$F_a^j = \sum_{i=1}^k f(d(i, j)) e_{ia}, \quad (3)$$

where e_{ia} is an indicator value set to 1 if protein i is annotated with a , $d(j, i)$ is the distance between functional profiles of proteins i and j and $f(d(i, j))$ is a function that transforms the distance to score. We use a distance-decreasing function of the form $f(d) = \frac{1}{1+\alpha d}, \alpha = 1$. It has the desirable property of a finite maximum at 1 for $d = 0$, and anti-monotonicity with respect to d . As our experiments show, the accuracy did not change significantly when alternative distance transform functions are used.

It is worth mentioning that since the two steps of our approach are completely independent, different approaches can be adopted for feature extraction and classification. Additionally, it is possible to exploit possible dependencies between the dimensions of the functional profile for the purposes of dimensionality reduction.

4. EXPERIMENTAL RESULTS

4.1 Interaction and annotation data

We measure the performance of our method on two yeast protein interaction networks. As a high confidence interaction network, we use the *Filtered Yeast Interactome (FYI)* from [12]. This network is created by using a collection of interaction data sources, including high throughput yeast two-hybrid, affinity purification and mass spectrometry, *in silico* computational predictions of interactions, and interaction complexes from MIPS [18]. The network contains 1379 proteins and 1450 interactions. *FYI* is an unweighted network, since every edge is added if it exists in more than two sources [12]. When performing the random walk on this network, the walker follows a uniformly chosen edge among the outgoing edges.

The second yeast interaction network is constructed by combining 9 interaction data sources from the *BioGRID* [1] repository. The method of construction is similar to the ones used in [7, 19, 17]. The network consists of 4914 proteins and 17815 interactions among them. The *BioGRID* network contains weighted edges based on scoring that takes into account the confidence in each data source and the magnitude of the interaction.

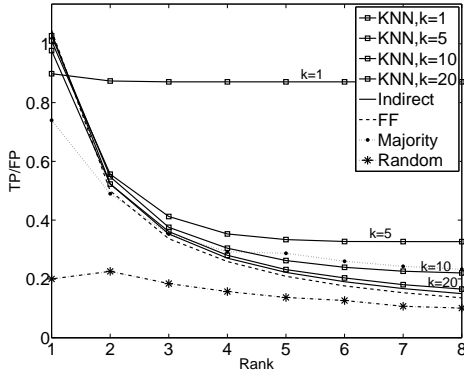


Figure 3: TP/FP ratio for the *BioGRID* network. All genes are labeled with exactly one annotation and the value of the frequency threshold is $T = 30$.

The protein GO annotations for *S. cerevisiae* gene products were obtained from the Yeast Genome Repository [9].

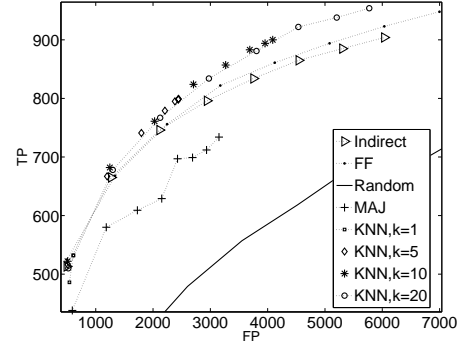
4.2 Existing techniques

We compare our *KNN* technique to *Majority (MAJ)* [21], *Functional Flow (FF)* [19] and *Indirect Neighbors (Indirect)* [7]. *Majority* scores each candidate function based on the number of its occurrences in the direct interactors. The scores of candidate functions in edge-weighted networks can be weighted by the probabilities of the connecting edges. *Functional Flow* [19] simulates a discrete-time flow of annotations from all nodes. At every time step, the annotation weight transferred along an edge is proportional to the edge’s weight and the direction of transfer is determined by the difference of the annotation’s weight in the adjacent nodes. The *Indirect* [7] method exploits both indirect and direct function associations. It computes *Functional Similarity* score based on *level 1* and *level 2* interaction partners of a protein. We used the implementation of the method as supplied by the authors, with weight function: *FSWEIGHT* and with minor changes related to the selection of informative functional annotations.

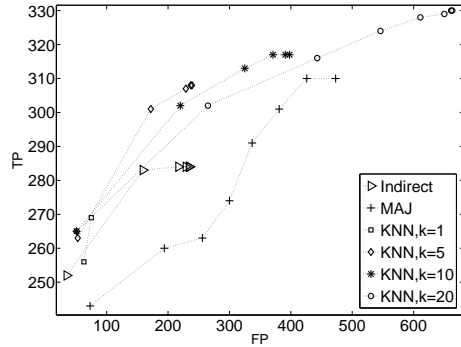
4.3 Experimental setup

The frequency of a functional annotation (class) is the number of proteins that are annotated with it. We call functions whose frequency exceeds a given threshold T as *informative*. An informative instance is a protein (represented by its functional profile) annotated with at least one informative class. For a given T , our training instance set contains all informative instances in the network. We exploit all available annotation information and predict functions at different levels of specificity. Unlike the approach in [8], we predict informative functions, even if their descendants are also informative.

We compare the accuracy of the techniques by performing leave-one-out validation experiments. We use leave-one-out validation because many annotations in the actual network are of relatively low frequency, and thus limiting the training set. Our classifier is working with actual networks, containing significant number of uncharacterized proteins and hence this is a realistic measure of the accuracy. Moreover, since the competing techniques implicitly use all available annotations, leave-one-out provides a fair comparison to our method. In this setup, a target protein is held out (i.e. its



(a) *BioGRID*, $T = 30$



(b) *FYI*, $T = 20$

Figure 4: TP versus FP for the (a) *BioGRID* and (b) *FYI* networks. All genes are labeled with exactly one annotation and the frequency thresholds are set respectively to 30 and 20.

annotations are considered unknown) and a prediction is computed using the rest of the annotation information in the network. All competing methods compute a score distribution for every class. We use the scores to rank the candidate functions and then analyze the accuracy for different ranks. An ideal technique would rank the true (held-out) annotation(s) as the top-most one. We penalize a technique for ranking false annotations above the actual ones. Additionally, we do not consider functions of zero score as actual predictions of the techniques.

A true positive (TP) prediction is a protein predicted as its actual label or any of the label’s ontological descendants. This is also known as the *true path* prediction criterion and has been used in previous ontology-aware prediction studies [8]. The motivation for the true path criterion is the gradual characterization of a given protein with more specific terms as more wet-lab experiments are performed. We analogously define a false positive (FP) prediction as a prediction of a function that is not part of the annotation of a target protein.

Though we pose the annotation prediction as a classification problem, it is not a general classification task. A domain scientist would be more interested in the TP and FP predictions, than in the number of True Negatives (TN) and False Negatives (FN). TNs in the prediction setting cannot facilitate the wet-lab experiments since the space of all possible functions is large, hence characterizing a protein using positive predictions is more tractable compared to using

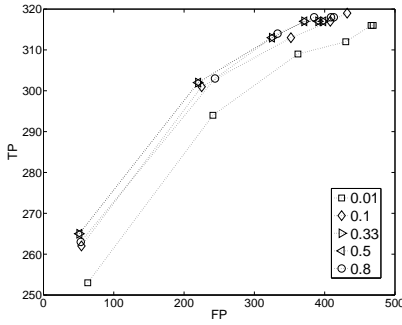


Figure 5: The effect of different restart probabilities r on the accuracy in the *FYI* network ($T = 20, k = 10$).

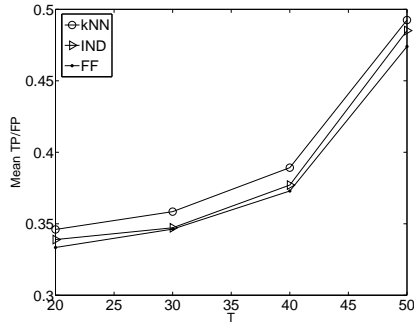


Figure 6: Effect of the informative functions threshold T . The average TP/FP ratio for each of the first 4 ranks is plotted for the *BioGRID* network. All genes are labeled with exactly one annotation and the value of k is set to 10.

negative ones.

The Receiver Operating Characteristic (ROC) is a commonly used metric for comparison of competing classification techniques. In an ROC setting, the True Positive Rate ($TPR = TP/P$) is plotted as a function of the False Positive Rate ($FPR = FP/N$). We show a variation of the ROC that skips the normalization terms, so that the actual number of false predictions is explicit in the plots.

4.4 Effect of parameters k, r, T and the distance conversion function $f(d)$

We first analyze the effect of the number of neighbors k in our kNN technique on the accuracy of the method. Figure 3 presents the ratio of TP/FP (accuracy ratio) as a function of the rank up to which labels are considered as predictions. We analyze this statistic for four different values of k and all competing techniques. We also examine the performance of a random predictor (*Random*) that uses solely the prior probabilities of the annotations.

The highest rank prediction for most of the methods produces roughly equal number of TP and FP. Compared to a random model, this accuracy is significantly higher as there are 18 candidate labels in this specific experiment ($T = 30$). The average number of classes for which 1NN gives predictions, i.e. classes that score greater than 0, is 1.2, hence the accuracy ratio of 1NN remains fairly stable for increasing ranks. The number of predictions increases with k and

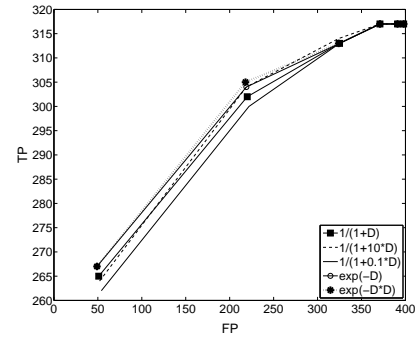


Figure 7: Effect of the distance to score conversion function on the accuracy of kNN. Our method is not sensitive to the exact form of the function (*FYI*, $T = 20, k = 10$).

more TP are discovered at the price of more FP.

A good predictor needs to be balanced with respect to its accuracy ratio and coverage, where the coverage is defined as the number of TP for increasing ranks, regardless of the FP introduced. According to this definition, kNN for small values of k can be regarded as high accuracy and low coverage method, and with increasing k , the coverage is increased at the price of lower accuracy. This effect can be observed in Figure 4(a). As k increases, the curves become less steep, however coverage improves. This effect of k is even more evident for the high confidence *FYI* network (Figure 4(b)). Traces for the FF and Random predictor are omitted for the *FYI* network for clarity since their performance is significantly dominated by the rest of the techniques.

We next study the effect of the restart probability of the Random Walks on the quality of the functional neighborhoods. As evident from Figure 5, the classification accuracy is not sensitive to the value of restart, as long as it is not chosen extremely low or extremely high. Values of 0.5 and 0.33 result in identical performance. Hence for all experiments we use a restart value of 0.33.

The overall relative performance of the techniques for varying informative threshold T is presented in Figure 6. We vary T from 20 to 50 for the *BioGRID* network and compare the average accuracy ratio of the first four ranks. Our technique dominates for all values of T . Note that when predicting low frequency classes, a lower value for k would result in a better prediction accuracy. However, for this specific experiment, we use a uniform value of $k = 10$ for all T .

We experiment with different distance conversion functions $f(d)$ in order to assess the sensitivity of our method to this parameter. The accuracy for three versions of our fractional function $f(d) = \frac{1}{1+\alpha d}$, $\alpha = 0.1, 1, 10$ as well as two exponential functions e^{-d} and e^{-d^2} are presented in Figure 7. Our method is not sensitive to the exact form of function, however we do not exclude the possibility of learning the optimal function for a given dataset.

4.5 Prediction accuracy

The prediction accuracy for single-labeled proteins is presented in Figures 4(a) and 4(b). As we already discussed, kNN outperforms the competing techniques when predicting single classes.

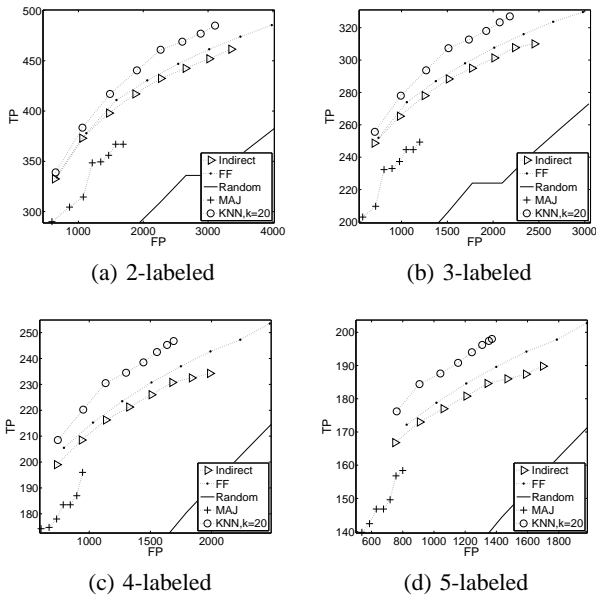


Figure 8: Performance comparison on the *BioGRID* network for (a) 2-, (b) 3-, (c) 4- and (d) 5-labeled proteins, $T = 30$.

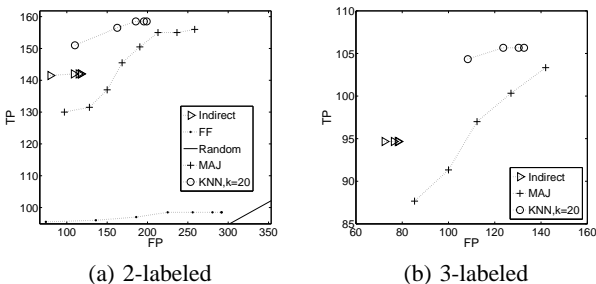


Figure 9: Performance comparison on the *FYI* network for (a) 2- and (b) 3-labeled proteins, $T = 20$.

A significant number of proteins in most genomes perform several functions, and hence have multiple annotations. We thus would like to analyze the performance of our technique on multi-labeled proteins. In this experiment, we group the proteins by the cardinality of their label set and perform leave-one-out validation. In this case every TP label is counted as a TP/C fraction of a true positive, where C is the cardinality of the label set. We take a similar approach when counting the false positives. In this set of experiments, we vary the rank up to which a label is considered predicted starting from C. Figures 8(a)-8(d) present the accuracy curves for proteins labeled with two and more annotations in the *BioGRID* network. The difference in performance between our method and competing methods is preserved when predicting more than one label. Similar plots are shown for the small *FYI* network in Figures 9(a), 9(b).

4.6 Discussion

The semantics of both GO *process* and *localization* imply that same terms would interact and hence cluster in a PIN. According to its definition, a GO *process* is a conglomerate of GO *functions* performed in a sequence. Genes localized in the same compartment of the cell, i.e. share GO *localization* terms, are also expected to

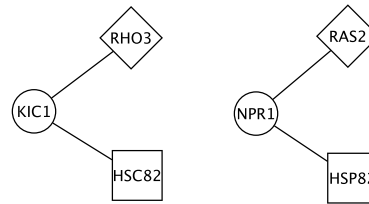


Figure 10: An example of two *Kinases* KIC1 and NPR1 that both interact with *GTPases* (RHO3 and RAS2) and *Unfolded Protein Binding* genes (HSC82 and HSP82)

interact more than ones of different localization. On the contrary, a GO *function* describes a molecular activity without specifying where, when or in what context this activity takes place. An example of two *Kinases* interacting with *GTPases* and *Unfolded Protein Binding* genes is presented in Figure 10. They share a functionally diverse pattern in their neighborhood, which could be captured by our feature extraction step.

We further analyzed the annotations of the three GO components in the high confidence *FYI* network. We call a label *related* to a target label if it is the same or any of the target’s ontological ancestors. More than 73% of the *localization* and 64.2% of the *process* annotations interact with more related annotations than unrelated ones. This percentage for the *function* hierarchy is only 58%. The semantic uniqueness of the GO function hierarchy makes it harder for *Direct methods* to infer uncharacterized proteins and this is why we concentrate on this specific part of GO. Our experiments on *process* and *localization* did not reveal a significant advantage of our method over existing ones.

Our method is robust to the density of the interaction network. This is demonstrated by the consistent accuracy dominance of our technique over the competing ones on two yeast interaction networks of different size, density and origin. A possible explanation for the robustness is the preservation of the neighborhood patterns in networks of diverse size and origin, which we think is a promising further direction for exploration.

5. CONCLUSION

We proposed a novel framework for predicting functional annotation in the context of protein interaction network. It is comprised of two independent components: extraction of neighborhood features and prediction (formulated as classification) based on these features. The only parameter k to which our approach is sensitive provides an intuitive interface for control of the trade-off between accuracy and coverage of our method. Our method is robust to the density and size of a PIN and its prediction accuracy is higher than that of previous methods.

The predictive power of our method gives further insight about the topological structure of functional annotations in a genome-wide network. The commonly adopted idea that similar functions are network neighbors does not hold for all annotations. A different structural annotation trend emerges, namely functions that observe similar (but sometimes heterogeneous) functional neighborhoods. Our approach incorporates this idea and has a better predictive power.

6. ACKNOWLEDGMENTS

This work is supported in part by National Science Foundation IIS-0612327. We thank Prof. Limsoon Wong for providing the implementation of the *Indirect Neighbor* [7] technique.

7. REFERENCES

- [1] Biogrid: General repository for interaction datasets. <http://www.thebiogrid.org/>, 2006.
- [2] V. Arnau, S. Mars, and I. Marin. Iterative clustering analysis of protein interaction data. *Bioinformatics*, 2005.
- [3] Z. Barutcuoglu, R. Schapire, and O. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 2006.
- [4] C. Brun, F. Chevenet, D. Martin, J. Wojcik, A. Guenoche, and B. Jacq. Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. *Genome Biology*, 5:R6, 2003.
- [5] T. Can, O. Camoglu, and A. K. Singh. Analysis of protein interaction networks using random walks. *Proceedings of the 5th ACM SIGKDD Workshop on Data Mining in Bioinformatics*, 2005.
- [6] J. Chen, W. Hsu, M. L. Lee, and S.-K. Ng. Labeling network motifs in protein interactomes for protein function prediction. *ICDE*, 2007.
- [7] H. Chua, W. Sung, and L. Wong. Exploiting indirect neighbors and topological weight to predict protein function from protein-protein interactions. *Bioinformatics*, 2006.
- [8] H. Chua, W. Sung, and L. Wong. Using indirect protein interactions for the prediction of gene ontology functions. *BMC Bioinformatics*, 2007.
- [9] Y. G. Database. <http://www.yeastgenome.org/>.
- [10] R. Dunn, F. Dudbridge, and C. Sanderson. The use of edge-betweenness clustering to investigate the biological function in protein interaction networks. *BMC Bioinformatics*, 2005.
- [11] T. gene ontology consortium. Gene ontology: Tool for the unification of biology. *Nature*, 2000.
- [12] J. Han, N. Bertin, and T. H. et Al. Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*, 2004.
- [13] H. Hishigaki, K. Nakai, T. Ono, A. Tanigami, and T. Takagi. Assessment of prediction accuracy of protein function from protein-protein interaction data. *Yeast*, 2001.
- [14] U. Karaoz, T. M. Murali, S. Letovsky, Y. Zheng, C. Ding, C. R. Cantor, and S. Kasif. Whole-genome annotation by using evidence integration in functional-linkage networks. *PNAS*, 101:2888–2893, 2004.
- [15] S. Letovsky and S. Kasif. Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics*, 19:i197–i204, 2003.
- [16] K. Maciag, S. Altschuler, M. Slack, N. Krogan, A. Emili, J. Greenblatt, T. Maniatis, and L. Wu. Systems-level analyses identify extensive coupling among gene expression machines. *Molecular Systems Biology*, 2006.
- [17] C. V. Mering, M. Huynen, D. Jaeggi, S. Schmidt, P. Bork, and B. Snel. String: a database of predicted functional associations between proteins. *Nucleic Acids Res.*, 2003.
- [18] H. W. Mewes, D. Frishman, U. Guldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Munsterkotter, S. Rudd, and B. Weil. Mips: a database for genomes and protein sequences. *Nucleic Acids Res.*, 30:31–34, 2002.
- [19] E. Nabieva, K. Jim, A. Agarwal, B. Chazelle, and M. Singh. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, 21:i302–i310, 2005.
- [20] M. P. Samanta and S. Liang. Predicting protein functions from redundancies in large-scale protein interaction networks. *PNAS*, 100:12579–12583, 2003.
- [21] B. Schwikowski, P. Uetz, and S. Fields. A network of protein-protein interactions in yeast. *Nature*, 2000.
- [22] R. Sharan, I. Ulitsky, and R. Shamir. Network-based prediction of protein function. *Molecular Systems Biology*, 2007.
- [23] V. Spirin and L. Mirny. Protein complexes and functional modules in molecular networks. *PNAS*, 2003.
- [24] G. X. Yu, E. M. Glass, N. T. Karonis, and N. Maltsev. Knowledge-based voting algorithm for automated protein functional annotation. *PROTEINS: Structure, Function, and Bioinformatics*, 61:907–917, 2005.