

# Graph Kernel-based Learning for Gene Function Prediction from Gene Interaction Network

Xin Li, Zhu Zhang, Hsinchun Chen

Department of Management Information Systems  
University of Arizona,  
Tucson, AZ 85721, USA  
(xinli, zhuzhang, hchen)@eller.arizona.edu

Jiexun Li

College of Information Science & Technology  
Drexel University  
Philadelphia, PA 19104, USA  
jiexun.li@ischool.drexel.edu

## Abstract

*Prediction of gene functions is a major challenge to biologists in the post-genomic era. Interactions between genes and their products compose networks and can be used to infer gene functions. Most previous studies used heuristic approaches based on either local or global information of gene interaction networks to assign unknown gene functions. In this study, we propose a graph kernel-based method that can capture the structure of gene interaction networks to predict gene functions. We conducted an experimental study on a test-bed of P53-related genes. The experimental results demonstrated better performance for our proposed method as compared with baseline methods.*

## 1. Introduction

In recent years, the rapid development in genome sequencing research has led to the identification of a large number of genes. Understanding the functions of these genes becomes a major challenge to biologists in the post-genomic era. Even though several efforts have been made to study these genes, many gene functions remain unknown. On average, as many as 70% of the genes in a genome have poorly known or unknown functions [5].

Due to the large quantity of unknown genes, it is infeasible to test every function for each gene by experiments. It is necessary to apply computational methods on existing experimental results and knowledge to provide suggestions for further experimental research. In previous studies, high-throughput experimental data, including gene sequences [9] and gene microarray co-expression [17], have all been used as evidence for gene function prediction.

Genes and their products interact and regulate with one another in biological pathways. The different types of gene/protein interactions can be detected by various tech-

niques including synthetic lethals, co-expression, phylogenetic profile, and so on [12]. The revealed and confirmed interactions, which are often documented in literature or knowledge-bases [3], may provide important clues for uncovered gene functions. Recent studies on gene function prediction utilized gene interactions and demonstrated promising performance [7].

In this study, we propose a novel approach to predict gene functions based upon the structure of gene interaction networks. Specifically, we design a graph kernel to capture the network structural information and learn patterns in gene interaction networks that are related to individual gene's functions. Our proposed computational method is evaluated using a test-bed of p53-related genes.

The remainder of this paper is organized as follow. In Section 2 we review related studies on gene function prediction based on gene interaction networks. In Section 3 we introduce the proposed graph kernel approach. In Section 4 we describe our experiments on performance evaluation and discuss the results. Section 5 summarizes our conclusions and future directions.

## 2. Literature Review

In this section, we survey previous studies on gene function prediction using gene interaction information. We also briefly review kernel-based learning methods.

### 2.1. Gene function prediction with gene interaction information

Gene functions can be inferred not only from genes' own features (e.g., gene sequence structures) but also from their interactions with other genes or proteins. After mapping proteins to genes, such interactions can be represented as a gene interaction network, in which some gene functions have been revealed. In this section we review gene function

prediction studies based on such gene interaction networks from the following two perspectives: levels of interactions and analytical techniques.

**2.1.1. Levels of interactions.** For the purpose of gene function prediction, we can utilize different levels of interactions in the network.

To predict a target gene’s function, the local view focuses on the directly connected genes. Based on an assumption that direct interactions indicate a strong correlation in the functions of connected genes, previous research proposed the “guilt by association” rule for gene function prediction [19]. Thus, the functions that most frequently appear in neighboring genes will be propagated to the one of interest. Other research constructed features on direct neighbors to describe genes and infer their functions. For example, both statistical approaches [18] and support vector machine (SVM) [11] have been applied to predict gene functions based on the similarities of neighbor genes.

Considering that indirectly connected genes may also be relevant to target genes, a global view of the network expands the scope of analysis to multiple levels of interactions. By extending the rule of “guilt by association,” Hisigaki et al. selected the functions with the highest probability to appear in multiple levels of neighborhood as target gene’s functions [6]. The label propagation approach adds a damping factor while transferring gene functions along interactions. Thus, genes which are farther from the target gene may have smaller probabilities to affect the target genes’ function prediction [16]. Based a similar assumption, a diffusion kernel approach has been proposed to predict gene functions according to their relative positions, where the genes with more and shorter paths between them are considered to have higher similarity in functions [11, 21]. In these studies, the indirect connections between genes are also considered as clues of related gene functions.

**2.1.2. Analytical techniques.** Gene function prediction can be performed at either gene cluster level or an individual gene level. If a group of genes have strong interactions with each other and can be considered as a cluster, they may share similar functions [8]. In this work, we are more interested in the task of predicting functions of individual genes, of which the analytical techniques can be roughly categorized into heuristic approaches and learning approaches.

Heuristic approaches often require predefining rules for prediction. The “guilt by association” rule [19, 6] and the label propagation method [16] are examples of heuristic approaches. Based on the “guilt by association” rule, several studies have formulated the gene function prediction as an optimization problem to minimize the inconsistency of function assignments on neighbor genes. Vazquez et al proposed to use simulated annealing [22] and Karaoz et al

proposed to use an iterative local search method [10] to find satisfactory solutions to the problem.

Compared with heuristic approaches, learning approaches extract patterns from training instances and infer the categories of other instances. As effective machine learning methods, kernel methods have been used in several studies for gene function prediction based on gene interaction networks. For example, linear kernel and diffusion kernel are used to capture the local and global information of gene interactions to predict gene functions [11, 21]. Learning approaches are also widely used in the gene function prediction from other types of features [9].

## 2.2. A brief review of kernel methods

Kernel methods are an effective approach to building classifiers, which has been widely used in previous gene function prediction studies. A kernel-based method contains a kernel function (kernel) and a kernel machine. A kernel function implicitly defines a similarity measure between data instances  $k : \chi \times \chi \rightarrow \mathbb{R} \quad (x, x') \rightarrow k(x, x')$ , which maps the data instances in the input space  $\chi$  to a feature space  $H$  (named reproducing kernel Hilbert space, RKHS)  $\Phi(x) : \chi \rightarrow H$ , where for every pair of data instances  $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$ . A kernel machine is an algorithm which performs learning tasks in this feature space  $H$ . SVM is a well-known kernel machine, which seeks optimal separating hyperplanes with the largest margin to separate data instances in the feature space. During the learning process, a kernel machine only requires kernel matrices, which contain kernel values between data instances, to build classifiers. Although there are limited types of kernel machines, the performance of a kernel method usually highly depends on the selection and design of kernel functions [20].

## 2.3. Summary

In previous research of gene function prediction using gene interaction network information, both directly connected genes and the entire network have been utilized. However, most existing techniques that capture global information of gene network belong to heuristic approaches. There have been limited studies that take a learning approach to capture global information and predict the functions of individual genes. The utilization of gene interaction network global information in learning approaches for gene function prediction need to be further addressed.

## 3. Research Design

In order to address the limitations in previous research, this study designs new kernels to capture global information of gene interaction networks for gene function prediction.

### 3.1. Methodology

Our gene function prediction method includes four major steps: (1) Network Construction: we extract gene interactions from public knowledge databases such as the BioGRID [3] to construct gene networks. (2) Network Annotation: genes in the network are annotated with their functions if known. Gene functions can be extracted from Gene Ontology [1] and MIPS [14]. The genes whose functions are currently unknown are annotated as “unknown.” (3) Classifier Construction: we take a kernel method and construct kernels to capture gene similarities that are related to their functions. Since a gene may have multiple functions, we build a binary classifier for each gene function. (4) Evaluation: the trained classifiers are evaluated by predicting functions of genes in the testing set. The predicted functions of a gene are compared with its real functions for evaluation.

### 3.2. Labeled graph kernel

Based on the assumption that a gene’s function is correlated to the genes’ functions it directly or indirectly interact with, we propose a labeled graph kernel (LGK) predict functions of individual genes based on the function distributions on their associated gene interaction networks.

In previous research, kernel methods have been used to capture network structural similarities. Graph kernels can be designed based on pairwise comparison of (matching) random walk paths of graph pairs. Such a method has been used in molecule structure based protein function prediction [2]. In these studies, random walk paths can start from any nodes in the network. However, we need to compare networks associated with genes. In addition, we take into account the distance between target genes and their associated genes in the network. Thus, our proposed LGK only compares the random walk paths starting from target genes.

The random walks are conducted following gene interactions from one gene to another in the network  $G$  (Figure 1). In each step, a random walk either jumps to one of the

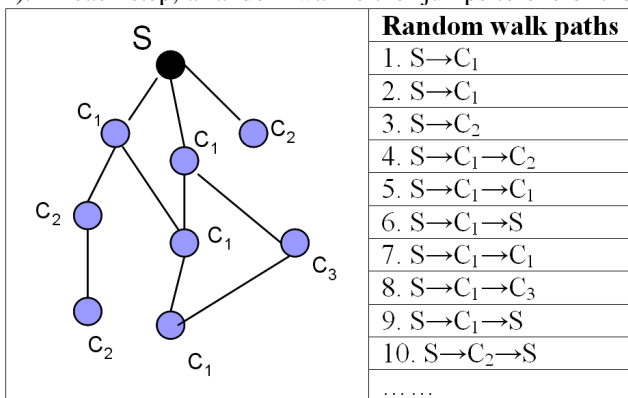


Figure 1. Random walk paths on a network.

neighbors or stops by following a probability distribution. Thus, any random walk path  $h$  have a probability  $P(h|G)$  to exist, which equals the multiplications all transition probabilities along the path. The similarity between two networks associated with two genes is calculated based on the pairwise comparisons of these random walk paths:

$$K'(G, G') = \sum_h \sum_{h'} k(h, h') P(h|G) P(h'|G')$$

where  $K(G, G')$  is the similarity between two genes’ associated networks.  $h$  ( $h'$ ) is a random walk path in network  $G$  ( $G'$ ).  $k(h, h')$  is the similarity of two random walk paths.

If the lengths of two paths are different, their similarities are considered as 0. Otherwise, each pair of random walk paths is compared according to the nodes (genes) on the paths. Two random walk paths’ similarity is calculated as the multiplication of all nodes’ similarities on the path.

$$k(h, h') = \prod_{i=1}^n \hat{k}(x_i, x'_i)$$

where  $x_i$  ( $x'_i$ ) represents the series of nodes on  $h$  ( $h'$ ).  $\hat{k}(x_i, x'_i)$  is the similarity of the two nodes.

The similarity of two nodes is calculated based on comparing their functions. As each node may have multiple functions, the comparison is conducted in a pairwise manner. Many gene function annotation systems, such as Gene Ontology, have a hierarchical structure. Thus, the similarity of these functions can be represented by the number of ancestors they share in the hierarchy. The more levels of ancestors two functions share, the higher similarity they have. An exponential damping effect  $\lambda$  is used to penalize the higher level of two functions’ shared ancestor:

$$\hat{k}(x_i, x'_i) = \sum_l \sum_{l'} 1 - \lambda^{com(l, l')}$$

where  $l$  ( $l'$ ) represent the multiple functions of  $x_i$  ( $x'_i$ ).  $com(l, l')$  is the level of ancestors two functions share.

At last, the kernel function needs to be normalized:

$$K(G_1, G_2) = K'(G_1, G_2) / \sqrt{K'(G_1, G_1) K'(G_2, G_2)}$$

## 4. Experimental Study

### 4.1. Test-bed

In our experiments, we used the collection of gene interactions from BioGRID database to build a gene network [3]. In BioGRID, there are 38,225 relations which are only related to *homo sapiens* genes from previous literature. By consolidating the duplicate relations, the final dataset contains 19,623 non-directional relations related to 7,167

genes. These interactions construct a graph with 150 individual components. The largest component contains 6,809 genes and 19,409 relations. On average, each gene interacted with 5.48 other genes. The graph has a clustering coefficient of 0.14, which is much higher than a random network of the same size (0.00076), indicating a high probability that genes form clusters.

Following a domain expert’s suggestion, we used the terms in the “biological process” hierarchy of Gene Ontology [1] as the functions and classification labels of individual genes. The “biological process” hierarchy is a 10-level structure with 7,172 GO terms. The first level (root level) is “biological process.” The second level has 7 nodes, including “unknown.” The third level has 264 nodes. Considering both the granularity of the function that we want to classify and the sample size needed to build a classifier, we chose the third level annotation of the GO “biological process” hierarchy as classification labels.

Among *homo sapiens* genes, the tumor suppressor gene, p53, has been well studied because of its central role in the regulation of apoptosis and cell cycle arrest in cancer development. Due to its significance, previous research provided a relatively reliable function annotation for p53 and its related genes. Therefore, in this study we focus on the p53-related genes as our test-bed. Based on our previous study on p53 pathway [13], we identified 2,045 p53-related genes. The 1,436 genes among them that have clearly defined biological process labels are selected for the experimental study. In total, these genes have 38 third level GO biological process annotations. To have enough data instances to learn the classifiers, we considered 9 of them which have more than 40 instances in our dataset for evaluation.

## 4.2. Evaluation metrics

We used precision, recall, and F-measure, which have been widely used in information retrieval and machine learning studies, to evaluate the classifiers’ performance. Since one gene can be assigned multiple functions and one function may have multiple genes, we can evaluate the performance of classifiers both at the instance (i.e., individual gene) level and at the class level. Instance-level precision ( $P_g$ ), recall ( $R_g$ ), and F-measure ( $F_g$ ) are defined as:

$$P_g = \frac{\text{correctly predicted functions of a gene}}{\text{all predicted functions of a gene}}$$

$$R_g = \frac{\text{correctly predicted functions of a gene}}{\text{all (known) functions of a gene}}$$

$$F_g = (2 \times P_g \times R_g) / (P_g + R_g)$$

Class-level precision ( $P_f$ ), recall ( $R_f$ ), and F-measure ( $F_f$ ) are defined as:

$$P_f = \frac{\text{correctly predicted genes of a class}}{\text{all predicted genes of a class}}$$

$$R_f = \frac{\text{correctly predicted genes of a class}}{\text{all (known) genes of a class}}$$

$$F_f = (2 \times P_f \times R_f) / (P_f + R_f)$$

## 4.3. Baseline methods

In the experimental study, we compared our proposed LGK method against two baseline methods: a linear kernel method(LK) [11] and a graph minimum cut algorithm called Gene Annotation using Interaction Networks (GAIN) [10]. In previous gene function prediction research using gene interaction networks, it is reported that the linear kernel has similar performance as the diffusion kernel[11] and GAIN has a similar performance as the “guilt by association” rule[15]. In addition, LK is a learning method using local information and GAIN is a heuristic method using global information.

**4.3.1. Linear kernel (LK).** Linear kernel has been used in previous studies to take advantage of gene interactions for function prediction [11]. In the LK, each gene  $g$  is represented by a vector of genes  $V_g$  which are directed connected with the target gene. The linear kernel  $K(g_1, g_2)$  is defined as the normalized inner product of the two vectors:

$$K'(g_1, g_2) = \langle V_{g_1}, V_{g_2} \rangle$$

$$K(g_1, g_2) = K'(g_1, g_2) / \sqrt{K'(g_1, g_1)K'(g_2, g_2)}$$

**4.3.2. Gene Annotation using Interaction Network (GAIN).** GAIN is a graph minimum cut algorithm developed by Karaoz et al. for gene function assignment [10]. This algorithm annotates gene functions in a network to minimize the number of linked gene pairs that have inconsistent assignment of functions. It represents the gene interaction network as a Hopfield network and conducts iterative local search on the network to find a satisfactory solution to the problem.

## 4.4. Experimental procedure

After we constructed the gene interaction network from interactions in BioGRID and annotated genes with their GO functions, we conducted 10-fold cross validation for each of the three gene function prediction methods. For each fold, functions of genes in the testing set are considered “unknown” as other genes whose functions have not been discovered. The genes in other nine folds are used as training set to build prediction models. For the two kernels(LK and LGK), we computed kernel matrices for each fold and used a well-known high-performance SVM package, “lib-SVM” [4], to train the classifiers. For the GAIN algorithm, we used the GAIN software package to make predictions (<https://bioinformatics.cs.vt.edu/~murali/software/gain/>).

The functions of genes in the testing set predicted by the three classifiers are compared with their true functions. We conducted pairwise *t*-test on precision, recall, and F-measure for each instance and for each class to compare the performance of different algorithms.

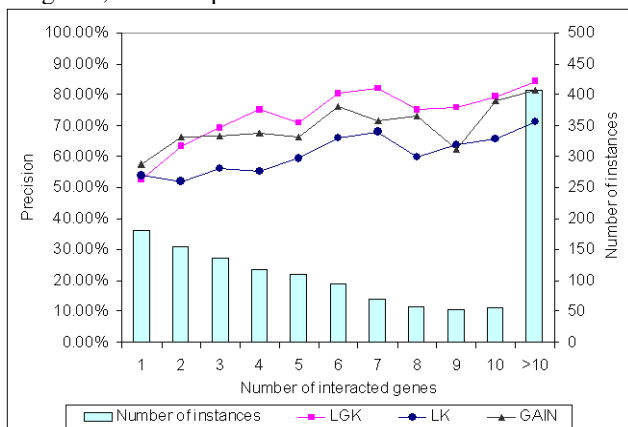
## 4.5. Experimental results and discussions

**4.4.1. Instance-level performance.** We first evaluated the prediction performances on individual genes. Table 1 reports the average precision, recall, and F-measure of predicting each gene’s functions achieved by the three classifiers. For all three measures, LGK consistently outperformed GAIN and LK. Pairwise *t*-tests showed that LGK outperformed GAIN on recall at the confidence level of 90%. LGK outperformed the two baseline methods on other measures at the confidence level of 99%.

**Table 1. Instance-level performance**

Classifiers	Average $P_g$	Average $R_g$	Average $F_g$
LGK	<b>73.57%</b>	<b>59.21%</b>	<b>61.45%</b>
LK	61.94%	56.55%	55.00%
GAIN	71.08%	57.91%	59.18%

It has been reported in previous research that the performances of classifiers are positively correlated to the number of genes that a gene interacts with. Our experimental results showed a similar pattern, especially for the precision measure (Figure 2). In addition to the positive correlation pattern, we also observe that when a gene interacting with one or two genes, the performance of LGK is slightly worse than GAIN. However, LGK’s performance increases significantly for genes interacting with more genes. For most of the genes, LGK outperformed GAIN and LK.



**Figure 2. Classification precision on different numbers of interacted genes.**

**4.4.2. Class-level performance.** Table 2 shows the performances achieved by the three classifiers on individual

classes. Although the three classifiers had similar performance on recall, their precisions have significant differences. In general, LGK had the highest precision for most classes. LGK and LK achieved the highest recall values in a similar number of classes. Class level pairwise *t*-test on precision, recall, and F-measure showed that LGK achieved higher precision than the other two classifiers at the confidence level of 95%. LGK also achieved higher recall and F-measure than GAIN at the confidence level of 95%. LGK achieved higher F-measure than LK at the confidence level of 90%, but there was no statistically significant difference in recall value between LGK and LK.

From Table 2, we also found that the performance for a class are positively correlated to the number of instances in the class. Although the three classifiers showed similar positive correlations between the performance and the number of instances in each class on recall and F-measure, they had different patterns for the precision measure. For classes with a large number of instances, all three classifiers achieved high precisions (from 70% to 80%). When a class had fewer instances, LK’s precision dropped faster than the other two methods. The performance of LGK on precision was the most stable of the three. Even when a class had a small number of instances, it still achieved a precision around 60%-70%.

## 5. Conclusions and Future Directions

In this research, we proposed a labeled graph kernel that can capture the structural information of gene interaction networks and provide effective predictions on gene functions. The experimental results showed that the proposed method significantly outperformed two baseline methods, especially for genes with a larger number of interactions or classes with a smaller number of instances.

In the future, we will extend the current research by incorporating other data sources. Furthermore, we will also extend the proposed approach to a multi-genome context to predict gene functions across different genomes.

## Acknowledgment

This work was supported by: NIH/NLM, 1 R33 LM07299-01. We thank the experts in Arizona Cancer Center for their comments on the research. We also thank Dr. Murali for providing the GAIN software that has been used as a baseline method in our study.

## References

- [1] 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.

**Table 2. Class level performance**

Class	Instances	$P_f$			$R_f$			$F_f$		
		LGK	LK	GAIN	LGK	LK	GAIN	LGK	LK	GAIN
GO:0006928	66	12.50%	<b>20.00%</b>	12.50%	1.52%	<b>4.55%</b>	3.03%	2.70%	<b>7.41%</b>	4.88%
GO:0030154	71	<b>66.67%</b>	29.03%	38.46%	5.63%	<b>12.68%</b>	7.04%	10.39%	<b>17.65%</b>	11.90%
GO:0016265	181	<b>76.54%</b>	48.28%	70.89%	<b>34.25%</b>	30.94%	30.94%	<b>47.33%</b>	37.71%	43.08%
GO:0009653	200	<b>48.72%</b>	26.72%	46.94%	<b>19.00%</b>	17.50%	11.50%	<b>27.34%</b>	21.15%	18.47%
GO:0009605	214	60.00%	40.00%	<b>68.75%</b>	<b>29.44%</b>	28.97%	25.70%	<b>39.50%</b>	33.60%	37.41%
GO:0006950	220	<b>67.47%</b>	40.74%	48.42%	<b>25.45%</b>	25.00%	20.91%	<b>36.96%</b>	30.99%	29.21%
GO:0008151	499	<b>60.83%</b>	48.64%	57.53%	38.28%	<b>39.48%</b>	38.28%	<b>46.99%</b>	43.58%	45.97%
GO:0007154	663	<b>78.35%</b>	73.36%	75.24%	68.78%	65.61%	<b>72.40%</b>	73.25%	69.27%	<b>73.79%</b>
GO:0008152	854	<b>82.25%</b>	73.43%	74.83%	78.69%	<b>79.63%</b>	76.23%	<b>80.43%</b>	76.40%	75.52%
Average		<b>61.48%</b>	44.47%	54.84%	33.45%	<b>33.82%</b>	31.78%	<b>40.54%</b>	37.53%	37.80%

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, G. Sherlock, and G. O. Consortium. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

- [2] K. M. Borgwardt, C. S. Ong, S. Schonauer, S. V. N. Vishwanathan, A. J. Smola, and H. P. Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21:147–156, 2005.
- [3] B. J. Breitkreutz, C. Stark, and M. Tyers. The grid: The general repository for interaction datasets. *Genome Biology*, 4(3):R23, 2003.
- [4] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>*, 2001.
- [5] A. J. Enright, V. Kunin, and C. A. Ouzounis. Protein families and tribes in genome sequence space. *Nucleic Acids Research*, 31(15):4632–4638, 2003.
- [6] H. Hishigaki, K. Nakai, T. Ono, A. Tanigami, and T. Takagi. Assessment of prediction accuracy of protein function from protein-protein interaction data. *Yeast*, 18(6):523–531, 2001.
- [7] P. Z. Hu, G. Bader, D. A. Wigle, and A. Emili. Computational prediction of cancer-gene function. *Nature Reviews Cancer*, 7(1):23–34, 2007.
- [8] M. A. Huynen, B. Snel, C. von Mering, and P. Bork. Function prediction and protein networks. *Current Opinion in Cell Biology*, 15(2):191–198, 2003.
- [9] L. J. Jensen, R. Gupta, H. H. Staerfeldt, and S. Brunak. Prediction of human protein function according to gene ontology categories. *Bioinformatics*, 19(5):635–642, 2003.
- [10] U. Karaoz, T. M. Murali, S. Letovsky, Y. Zheng, C. M. Ding, C. R. Cantor, and S. Kasif. Whole-genome annotation by using evidence integration in functional-linkage networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2888–2893, 2004.
- [11] G. R. G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004.
- [12] J. X. Li, X. Li, H. Su, H. C. Chen, and D. W. Galbraith. A framework of integrating gene relations from heterogeneous

data sources: an experiment on arabidopsis thaliana. *Bioinformatics*, 22(16):2037–2043, 2006.

- [13] X. Li, H. Chen, Z. Huang, H. Su, and J. Martinez. Global mapping of gene/protein interactions in pubmed abstracts: A framework and an experiment with p53 interactions. *Journal of Biomedical Informatics*, 2007.
- [14] 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 Research*, 30(1):31–34, 2002.
- [15] T. M. Murali, C. J. Wu, and S. Kasif. The art of gene function prediction. *Nature Biotechnology*, 24(12):1474–1475, 2006.
- [16] 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.
- [17] P. Pavlidis, J. Weston, J. S. Cai, and W. S. Noble. Learning gene functional classifications from multiple data types. *Journal of Computational Biology*, 9(2):401–411, 2002.
- [18] T. Schlitt, K. Palin, J. Rung, S. Dietmann, M. Lappe, E. Ukkonen, and A. Brama. From gene networks to gene function. *Genome Research*, 13(12):2568–2576, 2003.
- [19] B. Schwikowski, P. Uetz, and S. Fields. A network of protein-protein interactions in yeast. *Nature Biotechnology*, 18(12):1257–1261, 2000.
- [20] Y. Tan and J. Wang. A support vector machine with a hybrid kernel and minimal vapnik-chervonenkis dimension. *IEEE Transactions on Knowledge and Data Engineering*, 16(4):385–395, 2004.
- [21] K. Tsuda and W. Noble. Learning kernels from biological networks by maximizing entropy. *Bioinformatics*, 20(Suppl. 1):i326–i333, 2004.
- [22] A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani. Global protein function prediction from protein-protein interaction networks. *Nature Biotechnology*, 21(6):697–700, 2003.