Protein Protein Interaction using Fast Random Walk

Jing Xia
Kansas State Univeristy
234 Nichols Hall
Manhattan, KS USA
xiajing@ksu.edu

William Hsu
Kansas State Univeristy
234 Nichols Hall
Manhattan, KS USA
bhsu@ksu.edu

Abstract

Protein-protein interactions (PPI) refer to the associations between proteins and the study of these associations. Recent studies show that a network representation of proteins provides a more accurate model of biological systems and process compared to conventional pair-wise analysis. Many graph analysis has been proposed to the network for interactions prediction, pathway discovery and complex membership prediction. Random walk with restart (RWR) has been demonstrated to be a competitive approach on PPI network in terms of accuracy and effiency. RWR provides a good relevance score between two nodes in a weighted graph. However, with the high-thoroughput of the detection of PPI interactions, the straightforward application of RWR into the problem does not scale up well.

We propose fast solutions to this problem. The heart of the approach is to exploit the block-wise structural property of PPI and furthermore, an iterative aggregation and desegregation method is adapted into this problem. The results shows a speedup factor of 10 in terms of time and convergence. We proposed a new approach to integrate protein-protein pair information into network. Finally, we evaluate the proposed technique on prediction of interaction, pathway and complex membership using three different benchmark data sets. Our methods shows a similar and better results compared with previous work.

1 Introduction

In the past few years, much effort has gone into developing and applying methods for discovering the complete set of protein-protein interactions (PPI) in an organism. Identification of these functional modules in PPI network is the first step in understanding the organization and dynamics of cell functions. Experimentally, both large-scale and small-scale studies have collected a large number of results and store them in the database [8]. However, the comparison between results of large-scale methods and those of small-scale methods shows that high-throughout approaches such as yeast two-hybrid (Y2H) screens [26, 9] and mass spectrometric identification of protein complexes [7] are also prone to higher error rates that those conventional small-scale studies. It makes traditional presentation of interactions (binary interaction) and analysis methods [14, 20] not appropriate to use for analyzing the PPI network (without accounting for the quality or quantity of evidence supporting each interaction). Asthana *et al* in [1] proposed a probabilistic presentation of the PPI network, in which interactions are assigned confidence values to experimentally derived interactions using the manually curated catalogs of known complexes in MIPS (Munich Information Center for Protein Sequences) [15] as a trusted reference set. Other information integration techniques that utilize indirect genomic evidence [2, 13] and experimental interaction data sets [10] have provided methods that infer protein function, linkages and more accurate associations [28] with multiple supporting evidence.

Various graph analysis techniques have been proposed to mine these networks for pathway discovery [**?**, **?**] and prediction of complex membership [1]. The intrinsic cluster structure of a protein network provides more accurate biological insights compared to local pairwise comparisons. Bader and Hogue [**?**] propose a clustering algorithm to detect densely connected regions in a protein interaction network for discovering new molecular complexes. Other approaches post-process the cluster results through refining the cluster members based on functional homogeneity, cluster size and interaction density [**?**, **?**]. However, these clique-based, density-based approaches may miss the detection of some nodes may not be the same subgraphs. Incorporating additional biological information or graph characteristic into the network for preprocessing the structure of the network can be found in [4]. The authors add level-2 interactions into the PPI network and showed the increase of precision by this approach. These level-2 interactions represent indirect interactions between proteins which do not

directly interact and are calculated using only topological weights of the PPI network.

The characteristics of the PPI network, (1) error-prone identification of interactions and (2) incompleteness of the interactions, make the Random walk with restart (RWR) techniques be appropriate in this case. RWR represent the whole network as a weighted graph, in which each node represents a protein and each edge represents a probability between proteins. It may overcome the incompleteness property of the network because the random walk technique exploits the global structure of the network and provide a good relevance score between two nodes in a weighted graph. (i.e., RWR may assign a high probability of interaction between the nodes that do not directly interact).

In addition, random walk with restart (RWR) [18] has been successfully used in some applications, such as Automatic captioning of images [18, 25], outlier detection [16] and protein-protein interaction [3], personalized PageRank[6], and many more. The interpretation of RWR is that given a weighted graph, a particle starts from the initial state and move into another state randomly on every run, except that before the move, the particle returns back the initial state with a certain probability. After the number of runs goes to limit, there is a stationary distribution that represents the probability of every state which the particle will reach. This distribution will provide a good relevance score between two nodes.

The straightforward implementation of RWR needs a iteration of a large matrix or even worse an inverse of a matrix. It therefore does not scale well for large large graphs in many applications. In this paper, we propose a fast and sound solution to RWR and we apply it to the problem brought by protein-protein interaction (PPI) network. The basic idea underlying the algorithm is using the property that the network of PPI forms block-wise and community-like clusters (i.e. some proteins are more closely interactive to each other than to others). Based on this property, we propose a Iteratively Aggregation and Decomposition (IAD) method for RWR. The rest of the paper is organized as follows. We introduce random walk with restart models and present a motivation for our approach with its formal derivation in Section 3. Section 4 introduces the algorithm proposed in our work. Section 5 describes the data set and experiments in our experiments. We present experimental results in Section 6 and conclude with a summary and ideas for future work in Section 7.

2 Random Walk with Restart

The notation in this paper is shown in Table 1. RWR defines a transition $n \times n$ matrix P (n is the number of proteins) which model the probability of transition among proteins. Suppose P is row normalized (the sum of all elements in a row is 1), therefore P is irreducible and aperiodic and Frobenius Theorem [5] guarantee there is a unique stationary distribution of the matrix P (i.e. P has a left eigenvector corresponding eigenvalue 1). Based on the transition matrix P, RWR can be considered as a non-homogeneous case of Markov Chain. The formula of RWR is

Table 1. Symbols and Definition

Symbol	Definition
$\pi_{(k)}$	$1 \times n$ stationary distribution vector by running RWR from starting node k
$\pi_{(k)}^{(t)}$	distribution vector after t iterations runs
c	the restart probability, $0 < c < 1$
e_k	$1 \times n$ starting vector, the k^{th} element 1 and 0 for others
c_k	$1 \times n$ starting vector $c_k = ce_k$
n	the number of proteins in the graph
N	the number of partitions
P	the transition matrix
M	transition matrix, $M = (1-c)P$
A	$N \times N$ coupling matrix of M
m_i	the size of each sub matrix M_{ii}
c'_k	$1 \times m_k$ sub starting vector of the sub matrix M_{kk} where one element corresponding to the starting node k is 1 and 0 for others
π_i	sub eigenvector, $\pi = (\pi_1, \pi_2, ..., \pi_N)$

$$\pi_{(k)}^{(t+1)} = (1-c)\pi_{(k)}^{(t)} P + ce_k \tag{1}$$

where $\pi_{(k)}$ is the probability distribution of a particle starting at node k. $c \in (0,1)$ is a restarting probability, and e_k is an initial vector in which the k^{th} element is 1 and 0 for others. Each run, it plus a constant ce_k which has a meaning of "restart". Eq. (1), will converge after the number of iterations goes to a large value and the proof is shown in [**?**]. Thus we can have the equation

$$\pi_{(k)} = (1-c)\pi_{(k)} P + ce_k = \pi_{(k)} M + c_k \tag{2}$$

We observe a property of $\pi_{(k)}$, if a matrix M has the structure such as

$$M = \begin{pmatrix} M_{11} & 0 & ... & 0 & 0 \\ 0 & M_{22} & ... & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & ... & M_{N-1,N-1} & 0 \\ 0 & 0 & ... & 0 & M_{N,N} \end{pmatrix} \tag{3}$$

where each block sub matrix M_{ii} is size of m_i, $i = 1, 2, ..., N$ and M_{kk} has the starting protein k.

Replacing (2) with (3), we get

$$(\pi_1, \pi_2, ..., \pi_k, ..., \pi_N) = (\pi_1, \pi_2, ..., \pi_k, ..., \pi_N)$$
$$\times \begin{pmatrix} M_{11} & 0 & ... & 0 & ... & 0 \\ 0 & M_{22} & ... & 0 & ... & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & ... & M_{k,k} & ... & 0 \\ \vdots & \vdots & .... & \vdots & \ddots & \vdots \\ 0 & 0 & ... & 0 & ... & M_{N,N} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ c'_k \\ \vdots \\ 0 \end{pmatrix}^T$$

where π_k is the sub eigenvector for block sub matrix M_{kk} and c'_k is a $(1 \times m_k)$ vector of the sub matrix M_{kk} where one element corresponding to the starting node k is 1 and 0 for others.

Thus, each π_i can be found from

$$\pi_i M_{ii} = \pi_i, i \neq k \quad \text{and} \quad \pi_k M_{kk} + c'_k = \pi_k$$

Note that $\rho(M_{ii}) < 1$ [1], therefore $\pi_i = 0$, for $i = 1, 2, ..., N$, $i \neq k$. We only need to solve the equation $\pi_k M_{kk} + c'_k = \pi_k$. This property explains the nice observation of [23], in which the authors noticed that most of elements in the distribution are close to zero and therefore proposed an idea of performing RWR on the partitioned local block based on this property.

In the PPI network applications, we will exploit this type of community-like property, and find a way to construct a partition of M and the partition will make M which has a structure such as

$$M = \begin{pmatrix} M_{11} & M_{12} & ... & M_{1N} \\ M_{21} & M_{22} & ... & M_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ M_{N1} & M_{N2} & ... & M_{NN} \end{pmatrix}$$

where M_{ii} represents the links in a protein cluster and $M_{ij}, i \neq j$ represents the links among the clusters. The constraint of the partition is that we will maximize the weight (sum of all weight of links) in the cluster, whereas the weight between cluster will be minimized. With this property, we compute the left eigenvector for each diagonal square sub matrix $M_{ii}, i \neq k$

$$u_i M_{ii} = \lambda_i u_i \quad \text{and} \quad u_k M_{kk} + c'_k = u_k \tag{4}$$

where $\lambda_i \leqslant (1 - c)$. We want to use the eigenvectors u_i of M_{ii} as an approximation to π_i and further combine u_i into one eigenvector for the whole matrix. Following this idea, we adapt IAD [21, 22, 17] method to find an solution of the steady distribution of RWR.

[1] ρ is the spectral radius of a matrix, i.e. the maximal eigenvalue of a matrix

3 Iterative Aggregation and decomposition algorithm to RWR

We know that we can derive each eigenvector for each sub block. However, note that each sub eigenvector is a local eigenvector, thus we need a further step to combine these local eigenvectors into the whole one for the matrix M. The combination of each sub eigenvectors needs to take the weight of each sub block into account. The first part of IAD algorithm is for calculating this weight vector. The algorithm firstly constructs a coupling matrix from M by two steps. Suppose π_i is known for $i = 1, 2, ..., N$

1. replacing each row of each block M_{ij} with the sum of its elements in each row,
2. then multiple the column of each block from step one by a weighting factor ϕ_i, where $\phi_i = \pi_i / ||\pi_i||_1, i \neq k$, for $i = 1, 2, ..., N$

The coupled $(N \times N)$ matrix, A, can be, mathematically, represented as

$$a_{ij} = \phi_i M_{ij} e_{n \times N} \tag{5}$$

We want to solve the equation $\xi = \xi A + c_{1 \times N}$ and one observation is that A has such a stationary distribution

$$(||\pi_1||_1, ||\pi_2||_1, ..., ||\pi_N||_1) A = (||\pi_1||_1, ||\pi_2||_1, ..., ||\pi_N||_1)$$
$$\times \begin{pmatrix} \frac{\pi_1}{||\pi_1||_1} & 0 & ... & 0 \\ 0 & \frac{\pi_2}{||\pi_2||_2} & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & ... & 0 & \frac{\pi_N}{||\pi_N||_1} \end{pmatrix} M e_{n \times N} + c_{1 \times N}$$
$$= \pi_{(k)} M e_{n \times N} = (\pi_{(k)} - c_k) \begin{pmatrix} e & 0 & ... & 0 \\ 0 & e & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & ... & 0 & e \end{pmatrix} + c_{1 \times N}$$
$$= (||\pi_1||_1, ||\pi_2||_1, ..., ||\pi_k - c'_k||_1, ..., ||\pi_N||_1) + c_{1 \times N}$$
$$= (||\pi_1||_1, ||\pi_2||_1, ..., ||\pi_k||_1, ..., ||\pi_N||_1)$$

Note that $||\pi_k - c'_k||_1 + c = ||\pi_k||_1$, we can derive the last step. We let $\xi = (||\pi_1||_1, ||\pi_2||_1, ..., ||\pi_k||_1, ..., ||\pi_N||_1)$ be the this stationary distribution and consider it as the weight vector for each sub block matrix

ϕ_i is not known in advance, for $i = 1, 2, ..., N$; therefore, we hope to use the u_i as an approximation to π_i and expect this approximation not to cause a big error because the whole structure of M is close to the structure in Eq. (3) and $||M_{ii}||_1$ will be maximized if we can make a appropriate partition of M.

Therefore, we make an approximation

$$\phi_i^* = u_i / ||u_i||_1 \approx \phi_i = \pi_i / ||\pi_i||_1 \tag{6}$$

We use Eq. (6) to compute an approximation A^* to the coupling matrix A as

$$(A^*) = \phi_i^* M_{ij} e \tag{7}$$

We then determine an approximation eigenvector ξ^* from $\xi^* A^* + c_{1\times N} = \xi^*$ and use it to form the stationary distribution of M.

$$\pi_{(k)}^* = \left(\xi_1^*\phi_1^*, \quad \xi_2^*\phi_2^*, \quad ..., \quad \xi_N^*\phi_N^*\right) \tag{8}$$

The second part of the IAD is used for improving the approximation of Eq. (8). The simple way is to incorporate Eq. (8) back into Eq. (6) in hope of getting a better solution. However, direct using Eq. (8) will have no effect on the approximation [21].

Therefore, we adapt Takahashi [24] approach to improve the approximation before incorporating Eq. (8) back into Eq. (6). We will attempt to construct a matrix W_i, $i = 1, 2, ..., N$ as

$$W_i = \begin{pmatrix} M_{ii} & s_i \\ r_i^T & q_i \end{pmatrix} \tag{9}$$

and to complement the normalization of each row, the s_i will have to be a $m_i \times 1$ vector:

$$r_i^T = \begin{cases} \frac{1}{1-\xi_i} \sum_{j\neq i} \xi_j \phi_j M_{ji} & \text{if i} \neq \text{k} \\ \frac{1}{1-\xi_k}(c_k + \sum_{j\neq k} \xi_j \phi_j M_{jk}) & \text{if i = k} \end{cases} \tag{10}$$

one observation shows that

$$s_i = e - M_{ii} e \quad \text{i = 1, 2, ..., N}$$

and

$$q_i = 1 - r_i^T e \quad \text{i = 1, 2, ..., N}$$

And one observation shows that

$$(\pi_i, 1-\xi_i) \begin{pmatrix} M_{ii} & s_i \\ r_i^T & q_i \end{pmatrix} = (\pi_i, 1-\xi_i) \tag{11}$$

The proof can be seen in [**?**]. With the constructed block W_i, we can obtain a new π_i and ξ_i through solving the Eq. (11). Finally we update $\phi_i = \pi_i/\xi_i$ with the new π_i and ξ_i obtained from W_i,

Therefore, the algorithm is

1. Let $\pi_{(k)}^{(0)} = (\pi_1^{(0)}, \pi_2^{(0)}, ..., \pi_N^{(0)})$ be a given initial approximation to the solution $\pi_{(k)}^{(0)}$ and set $m = 1$.

2. Compute ϕ^{m-1}: For $i = 1, 2, ..., N$ compute

$$\phi_i^{(m-1)} = \pi_i^{(m-1)} / ||\pi_i^{(m-1)}||_1$$

3. Construct the aggregation matrix $A^{(m-1)}$ whose elements are given by

$$(A^{(m-1)})_{ij} = \pi_i^{(m-1)} M_{ij} e$$

4. Solve the eigenvector problem

$$\xi^{(m-1)} A^{(m-1)} + c_k = \xi^{(m-1)}$$

5. For $i = 1, 2, ..., N$ form W_i for $\pi_i^{(m)}$ and $\xi_i^{(m)}$ from Eq. (11) and update $\phi_i^{(m)} = \pi_i^{(m)}/\xi_i^{(m)}$

6. Construct a test for convergence. If the estimated accuracy is sufficient (less than ϵ), then stop and take

$$\pi_{(k)}^{(m)} = (\xi_1^{(m-1)}\phi_1^{(m)}, \xi_2^{(m-1)}\phi_2^{(m)}, ..., \xi_N^{(m-1)}\phi_N^{(m)})$$

Otherwise set $m = m + 1$ and go step (3).

4 Experiment and Results

4.1 Data Set

Qi *et al.* [19] divide the protein interaction prediction task into three sub-tasks: (1) prediction of physical (or actual) interaction among proteins, (2) prediction of proteins belonging to the same complex and (3) prediction of proteins belonging to the same pathway. We used three different protein-protein interaction data set in Qi's work as a benchmark from [19]. It includes data from the MIPS (Munich Information Center for Protein Sequences) [15], data from DIP (Database of Interacting Proteins) [29] and data from KEGG [11]. Each interaction in the three data sets has a feature vector of 162 dimension. For an example, if there is an interaction between protein A and protein B, then there is a feature vector to represent this interaction. The description how these features are extracted is orthogonal to this paper.

4.2 Network Construction and Partition

To construct a PPI network in which each interaction will be assigned a quantifier indicating the strength of its interaction, we use machine learning approach, specifically, support vector machine algorithm [27] to learn such a probability for each protein-protein pair example. A probability will be assigned to each pair in terms of its distance to the decision hyper-plane. In our work, CLUTO [12] is used to partition the graph with the constraint of maximizing the inner-weight of a cluster, and minimizing the crossing-weight among clusters. We tune the parameters such as the number of the clusters N and different types of similarity functions, and make choose the best parameter ($N = 180$)

Table 2. Characteristics of data sets

	DIP	MIPS	KEGG
number of proteins	1451	870	1129
number of interactions	5232	16472	77922

Table 3. Accuracy after First Iteration ($||\pi^{(1)} - \pi||_2$)

	Power Method	IAD
DIP	$5.93e^{-2}$	$1.47e^{-2}$
MIPS	$2.12e^{-2}$	$1.74e^{-3}$
KEGG	$1.02e^{-2}$	$2.95e^{-4}$

that returns best criteria (the size of each clusters and the inner-weights and crossing-weights of the partitions).

Figure (1) shows a preview and post-clustering view of PPI network.

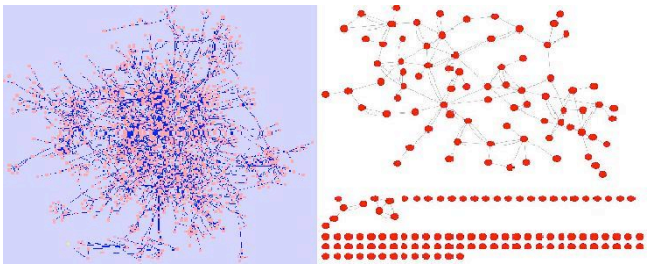

Figure 1. Using CLUTO on DIPS data set. The number of clusters is chosen to maximize a criterion in CLUTO which indicates maximal inner weights of a cluster . There are 180 clusters in the right figure.

4.3 Evaluation of Random Walk

We measure the efficiency of random walk in two perspectives of view, the converge rate and the overall converge steps. Table 3 and Table 4 compares the traditional power-method (i.e. multiply the transition matrix with π until the L_2 norm of successive estimates of π below the threshold ϵ) and the IAD-based RWR in terms of first step's accuracy and number of converge steps. Figure 2 compares the convergence rate of Power method and that of IAD-based algorithm on three sets of graph.

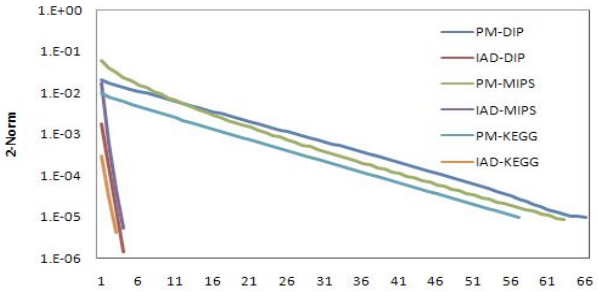

Figure 2. Converge rate

Table 4. Number of iterations for convergence ($\epsilon = 10^{-5}$)

	Power Method	IAD
DIP	62	4
MIPS	66	4
KEGG	57	3

4.4 Accuracy of RWR

In order to evaluate the performance of the random walk technique for the prediction protein-protein relationship, we used 10 randomly selected directed interaction pathways from DIPS the 27 MIPS complexes examined by Asthana et al. [1] and 10 selected pathways from the KEGG pathway database which is also used in [3]. We used the leave-one-out benchmark to assess the accuracy of the analysis techniques. In this benchmark, for each of the directed interaction pathways, the complexes and pathways examined, one member protein and its connected interactions in the path are left out in turn. We used this protein as query to the the remaining network (i.e. run random walk with restart starting from the left-out protein) and see if RWR can find the remaining set of proteins of the core complex or in the partially known pathway. The ratio of how many proteins given by the query to the size of the complex or pathway provides a measure of accuracy. A successful analysis method should report the all the remaining proteins in top ranks. Therefore, the accuracy in the Figure (3) shows, above a threshold rank k the average ratio of the number of complex proteins found by each leave-one-out query to the size of the complex or the pathway plus k.

5 Conclusion

In this paper, we propose a fast solution to random walk and we apply it into protein-protein network. We evaluated its efficiency and ability of predicting for the complex mem-

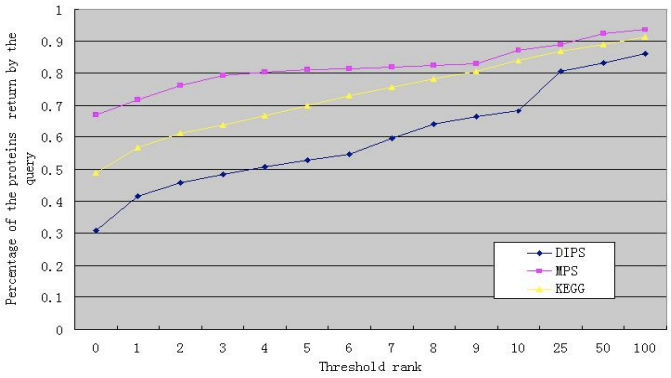

Figure 3. the accuracy of the number of complex proteins found by each leave-one-out query

bership problem in the protein-protein network. We also proposed a new approach to integrate protein-protein pair information into network (i.e. how to construct the similarity protein-protein network from the profile of protein protein interaction). The relevance score defined by RWR has many good properties: compared with those pair-wise metrics, it can capture the global structure of the graph; compared with those traditional graph distances (such as shortest path, maximum flow etc), it can capture the multi-facet relationship between two nodes. The experimental results shows that RWR it is a promising method that can scale well for large, genome-scale protein networks.

References

[1] S. Asthana, O. D. King, F. D. Gibbons, and F. P. Roth. Predicting protein complex membership using probabilistic network reliability. *Genome Research*, 14:11701175, May 2004.

[2] P. M. Bowers, M. Pellegrini, M. J. Thompson, J. Fierro, T. O. Yeates, and D. Eisenberg. Prolinks: a databaseof protein-functional linkagesderivedfrom coevolution. *Genome Biology*, 5(5):R35, 2004.

[3] T. Can, O. Çamoğlu, and A. K. Singh. Analysis of protein-protein interaction networks using random walks. *BIOKDD '05: Proceedings of the 5th international workshop on Bioinformatics*, 418:613–622, July 2005.

[4] H. N. Chua, K. Ning, W. K. Sung, H. W. Leong, and L. Wong. Using indirect protein-protein interactions for protein complex prediction. *Journal of bioinformatics and computational biology*, 6(3):435–466, June 2008.

[5] G. H. Golub and C. V. Loan. *Matrix Computation*. 1996.

[6] T. H. Haveliwala. Topic-sensitive pagerank. *WWW*, (517-527), 2002.

[7] Y. Ho, A. Gruhler, A. Heilbut, G. Bader, L. Moore, S. Adams, P. Millar, A. adn Taylor, K. Bennett, and K. Boutilier. Systematic identification of protein complexes in saccharomycescerevisiae by mass spectrometry. *Nature*, 415:180–183, 2002.

[8] P. Hodges, A. McKee, B. Davis, W. Payne, and J. Garrels. The yeast proteome database (ypd): a model for the organization and presentation of genome-wide functional data. *Nucleic Acids Research*, 27(1):6973, May 1999.

[9] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci.*, 98:45694574, 2001.

[10] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein. A bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302:449–453, October 2003.

[11] M. Kanehisa and S. Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, January 2000.

[12] G. Karypis. Cluto - a clustering toolkit. *University of Minnesota, Technical Report*, 2002.

[13] I. Lee, S. V. Date, A. T. Adai, and E. M. Marcotte. A probabilistic functional network of yeast genes. *Science*, 306:1555–1558, November 2004.

[14] E. Marcotte, M. Pellegrini, M. Thompson, T. Yeates, and D. Eisenberg. A combined algorithm for genome-wide prediction of protein function. *Nature*, 402:8386, 1999.

[15] H. W. Mewes, D. Frishman, K. F. X. Mayer, M. Mnsterktter, O. Noubibou, T. Rattei, M. Oesterheld, and V. Stmpflen. Mips: Analysis and annotation of proteins from whole genomes. *Nucleic Acids Res*, 32:41–44, 2004.

[16] H. D. K. Moonesignhe and P.-N. Tan. Outlier detection using random walks. *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, 418:613–622, July 2006.

[17] D. P. O'Leary. Iterative methods for finding the stationary vector for markov chains. *Linear Algebra, Markov Chains, and Queuing Models*, 418:613–622, July 1992.

[18] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. *Proceedings of the 10th ACM SIGKDD Conference*, 418:236–243, July 2004.

[19] Y. Qi. Learning of protein interaction networks. dissertation. *Carnegie Mellon University*, 2008.

[20] B. Schwikowski, P. Uetz, and S. Fields. A network of proteinprotein interactions in yeast. *Nat. Biotechnol.*, 18:1257–1261, 2000.

[21] W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. 1996.

[22] W. J. Stewart. Numerical methods for computing stationary distribution of finite irreducible markov chains. *of Advances in Computational Probability*, 418:613–622, July 1998.

[23] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. Neighborhood formation and anomaly detection in bipartite graph. *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, pages 418–425, July 2005.

[24] Y. Takahashi. A lumping method for numerical calculations of stationary distributions of markov chains. *Technical Report B-18, Dept. of Information Sciences, Tokyo Institute of Technology,*, 418:613–622, July 1975.

[25] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, 418:613–622, July 2006.

[26] P. Uetz, L. Giot, G. Cagney, T. Mansfield, R. Judson, J. Knight, D. Lockshon, V. Narayan, M. Srinivasan, and P. Pochart. A comprehensive analysis of proteinprotein interactions in saccharomyces cerevisiae. *Nature*, 403:623627, 2000.

[27] V. N. Vapnik. *The Nature of Statistical Learning Theory (Statistics for Engineering and Information Science).* Springer Verlag, December 1999.

[28] C. von Mering, L. J. Jensen, B. Snel, S. D. Hooper, M. Krupp, M. Foglierini, N. Jouffre, M. A. Huynen, and P. Bork. String: known and predicted protein-protein associations, integrated and transferred across organisms. *Nucleic Acids Research,*, 33:D433–D437, 2005.

[29] I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S.-M. Kim, and D. Eisenberg. Dip, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res*, 30(1):303–305, 2002.