# Inferring Protein-Protein Interactions Using Interaction Network Topologies

Alberto Paccanaro[†*], Valery Trifonov[‡*], Haiyuan Yu[†], Mark Gerstein[†]
[†]Department of Molecular Biophysics and Biochemistry
[‡]Department of Computer Science
Yale University, New Haven, CT 06520, USA
E-mail: {alberto.paccanaro, valery.trifonov, haiyuan.yu, mark.gerstein}@yale.edu
[*these authors contributed equally to this work]

*Abstract*— **We describe two novel methods for predicting protein interactions, using only the topology of an observed protein interaction network. The first method searches the protein interaction network for defective cliques (*i.e.* nearly complete complexes of pairwise interacting proteins), and predicts the interactions that complete them. The second method computes the diffusion distance between each pair of proteins and then infers an interaction when such distance is below a given threshold. We show that both methods have a good predictive performance and compare their results.**

## I. Introduction

A fundamental problem in modern biology is the identification of the complete set of interactions among the proteins in a cell [12], [10], [7]. Different experimental methods are available to identify such interactions, and they can be roughly divided into two main categories: small-scale (low throughput) and large-scale (high throughput) techniques. Given a set of proteins, small-scale techniques such as co-IP determine the interaction between one pair of proteins at a time [18], [11], [17], [1]. On the other hand, large-scale techniques, *e.g.* yeast two-hybrid and TAP-tagging, allow identifying a large number of interacting pairs in a single experiment [3], [4], [5], [15].

With the advent of genome-wide analysis, we are interested in the identification of the interaction among a great number of proteins (even of all the proteins in a genome). When the number of proteins is in the thousands, the number of possible interacting pairs is in the millions [8]. To discover all these interactions using small-scale experiments becomes very labor-intensive and time-consuming, and in this situation large-scale experiments are preferred.

However, low throughput experiments allow much more precise identification of the interacting pairs than high throughput experiments — the latter are known to be more error-prone [6], [16].

Two types of errors are possible: the large-scale experiment can wrongly indicate that an interaction exists, *i.e.* yield a false positive (FP); or it can fail to detect an interaction that actually exists, thus producing a false negative (FN). However, experimentalists would agree that the these two types of errors occur with different frequency in large-scale experiments. While false positives have "higher visibility" due to the relatively small number of true interactions, it is generally observed that experiments allow a higher absolute degree of confidence when an interaction is observed, but a much lower degree when no interaction is detected. In other words, most of the errors (as an absolute count, not relative to the numbers of actual interacting or noninteracting protein pairs) are false negatives: it is believed that when no interaction is detected, it is not unlikely that the interaction actually exists, but the experiment has failed to detect it. In support of this observation, Figure 1 shows the differences between the low-throughput and high-throughput experimental data on protein-protein interactions in a subset of 56 proteins of *S. cerevisiae*, for which we were able to obtain complete matrices of experimental results. Of the 1596 pairs of proteins (including possible self-interactions), the results of the two types of experiments were the same for 1033; in the 563 cases when the results were different, 521 (92.5%) were false negatives and 42 (7.5%) were false positives.

Ideally, we would like to have a computational method which would be able to correct many of the errors made by large-scale interaction experiments.

In this paper we propose two new methods, based purely on topological properties of graphs representing protein interaction networks, that attempt to infer those interactions that have been missed by large-scale experiments.

## II. The Defective Clique Completion algorithm

The basic idea of the defective clique completion algorithm derives from the way in which pull down experiments, a particular type of large scale experiments, are carried out, and particularly from the matrix model interpretation of their results [3], [4], [13], [2]. In these experiments one protein—the bait—is used to pull out the set of proteins interacting with it, *i.e.* its protein complex, in the form of a list. When such lists differ only in a few elements, it is reasonable to assume that this is due to experimental errors, and such elements should therefore be added (thus making the lists equal). Each list can be represented as a fully connected graph in which proteins occupy the nodes. Then the problem of identifying lists that differ in only a few elements is equivalent to finding a clique

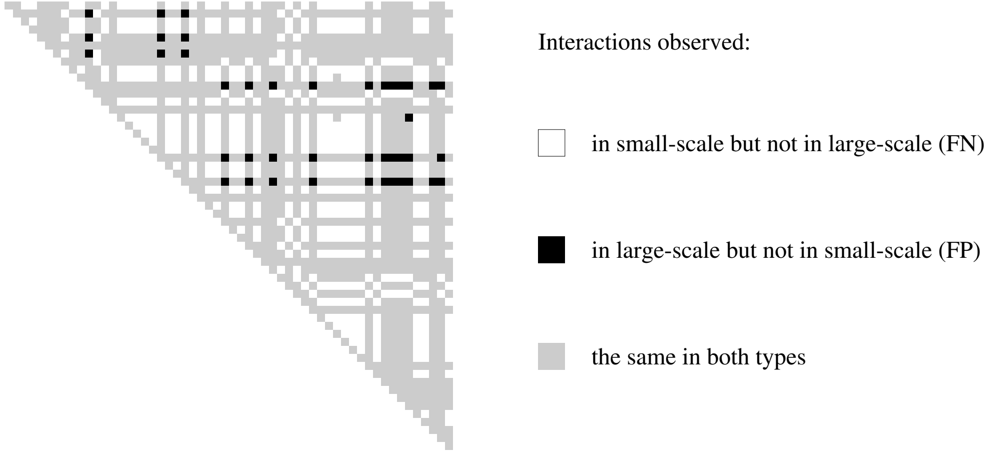

Fig. 1. A graphical representation of the symmetric matrix of the differences between complete protein-protein interaction data obtained in small-scale and large-scale experiments on 56 proteins of *S. cerevisiae*. Only the upper triangular part is shown, and element $(i, j)$ of the matrix represents the interaction between protein $i$ and protein $j$. White squares indicate interactions observed in small-scale but not in large-scale experiments (false negatives); black squares stand for interactions observed in large-scale but not in small-scale experiments (false positives); gray squares show protein pairs for which both the small- and the large-scale experiments produced the same result. The number of false negatives exceeds the number of false positives by an order of magnitude.

with a few missing edges, which we shall call a defective clique.

We shall represent a protein interaction network with a graph $G$, whose vertices $V$ are proteins, and whose edges $E$ are the pairs of interacting proteins. A *clique* in a graph is a set $K$ of vertices such that $K \times K \subseteq E$, *i.e.* each pair of vertices in $K$ is connected by an edge in $E$. The *size* of this clique is the number of vertices in it.

As we just discussed, under the matrix model interpretation of the results of large-scale experiments, two proteins interacting with the same protein clusters are likely to interact with each other. Thus in graph-theoretic terms our approach is based on the following observation about protein interaction networks:

($*$) If vertices $P$ and $Q$ are both adjacent to each vertex in a clique $K$, then it is likely that $P$ and $Q$ are adjacent to each other, if they are not adjacent already.

This observation can be depicted as shown in Figure 2; in this example the size of the clique $K$ is 5. The dashed edge between $P$ and $Q$ corresponds to an interaction which is missing from the experimental data, but which (according to observation ($*$)) is very likely to occur. We say that $P, Q$, and $K$ form a *defective clique* $KPQ$ with a missing edge $PQ$.

Clearly the size of $K$ plays an important role in determining how likely it is that $P$ and $Q$ interact. For example, if the size of $K$ is 1 (*i.e.* $P$ and $Q$ both interact with one or more proteins, but those proteins do not interact among themselves), the likelihood of an interaction between $P$ and $Q$ is much smaller than it is in the case when the size of $K$ is, say, 42. Thus a natural parameter of a prediction algorithm based on observation ($*$) is the minimal size $k$ of $K$ for which the interaction $PQ$ is predicted.

Another parameter with which we can extend observation ($*$) is the number of edges missing from the clique when its size is sufficiently large. We will discuss the effects of this parameter in subsection B, when we describe our algorithm in detail.

### A. *An algorithm for finding defective cliques*

Our definition of a defective clique does not suggest immediately a method for finding such patterns in a protein interaction network. For this purpose it is useful to find an alternative characterization of a defective clique in standard graph-theoretic terms, which will allow us to use some off-the-shelf algorithms.

The main idea of our algorithm is based on the realization that a defective clique $KPQ$ of size $n$ with one missing edge is the union of two (complete) cliques of size $n-1$, namely $K \cup \{P\}$ and $K \cup \{Q\}$, as shown in Figure 3. Thus we can reduce the algorithm for finding defective cliques to repeating the following steps until reaching a fixed point:

1) Step 1: Find all cliques in the network.
2) Step 2:
   - Find pairs of cliques overlapping on all but one node each.
   - In each of these pairs predict the edges between the non-overlapping nodes.
   - Add the new edges to the network.

The algorithm terminates when no new edges were added in Step 2.

However, directly applying this naïve recipe to typical protein interaction networks is unrealistic, for the following reason: Since every subset of nodes in a given clique is itself a clique, the number of all cliques in a graph is at least $2^m$, where $m$ is the size of the largest clique in the graph. For example, the experimental data for the protein interaction

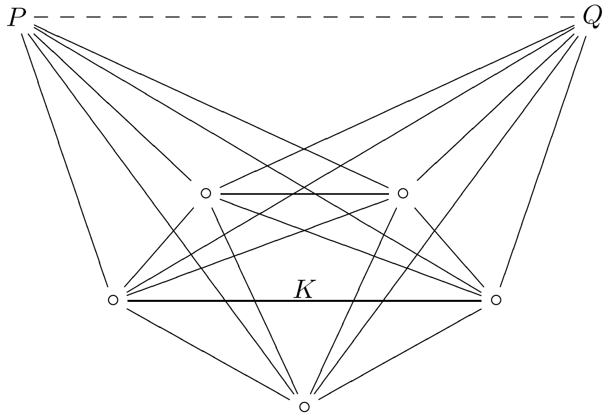

Fig. 2. A defective clique in a protein interaction network; the dashed edge between proteins $P$ and $Q$ corresponds to a predicted interaction.

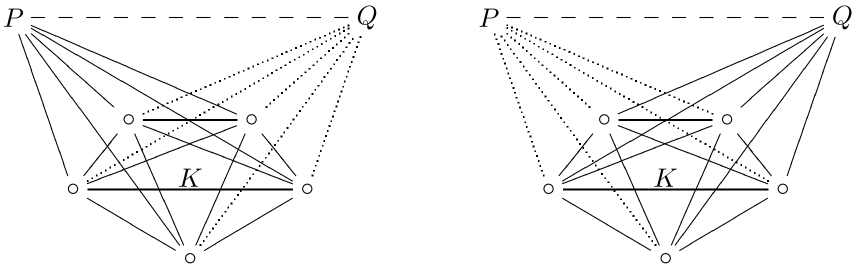

Fig. 3. The decomposition of a defective clique into the union of two overlapping cliques.

network of *S. cerevisiae* we used to test our algorithm (see Section IV) contains four cliques of size 38; this yields more than $10^{12}$ cliques (even if we do not consider cliques of size less than 5, whose number is negligible), hence more than $10^{23}$ pairs of cliques to check in Step 2 of the algorithm. Since this number is prohibitively large, we need a more effective formulation of the algorithm. For this purpose in the next section we design an equivalent algorithm which only considers the *maximal* cliques in the graph.

### *B. Improving Efficiency Using Maximal Cliques*

A *maximal* clique in a graph $G$ is one which is not contained in any other clique in $G$. In the worst case the problem of finding all maximal cliques still takes time exponential in the size of the graph[1]; however, if Step 1 is modified to only produce the maximal cliques in the graph, as discussed in the previous section its output would be exponentially smaller than with the naïve approach. This would reduce by an exponent the running time of Step 2 of the algorithm.

In practice, the protein interaction networks are rather sparse (*e.g.* less than 15,000 interactions are observed with high confidence in the network of *S. cerevisiae*, out of over 18 million possible pairs of about 6,000 proteins [16]). Our results show that existing algorithms for finding maximal cliques [14] are very efficient on graphs with this structure.

However, if we only compare maximal cliques for overlap on all but one node each, as we did with all cliques in the naïve version, the output of this algorithm will not be the same as that of the naïve version. The reason is that if a defective clique $KPQ$ consists of a core clique $K$ and two nodes $P$ and $Q$, we know that $K \cup \{P\}$ and $K \cup \{Q\}$ are cliques, but in general they are not maximal cliques. Suppose $K_P$ and $K_Q$ are cliques containing $P$ and $Q$ respectively, and such that $K \cup K_P$ and $K \cup K_Q$ are maximal cliques. (If $K \cup \{P\}$ and $K \cup \{Q\}$ are cliques, then $K_P$ and $K_Q$ always exist, but are not necessarily unique.) Then Step 2 of the algorithm will compare $K \cup K_P$ and $K \cup K_Q$; however, $K_P$ in general may contain other nodes in addition to $P$, and these nodes will not necessarily all be in $K_Q$. As a result, the nonoverlapping parts of the maximal cliques will consist of more than one node each, and the naïve algorithm will ignore the pair and thus fail to predict the edge $PQ$.

Hence, to obtain the same results as with our original algorithm, we have to modify Step 2 of the algorithm to look for partial overlaps of maximal cliques which differ in more than one node. This leads us to generalize the notion of a

[1] More precisely, the problem is NP-complete, *i.e.* only exponential-time algorithms are known for it.

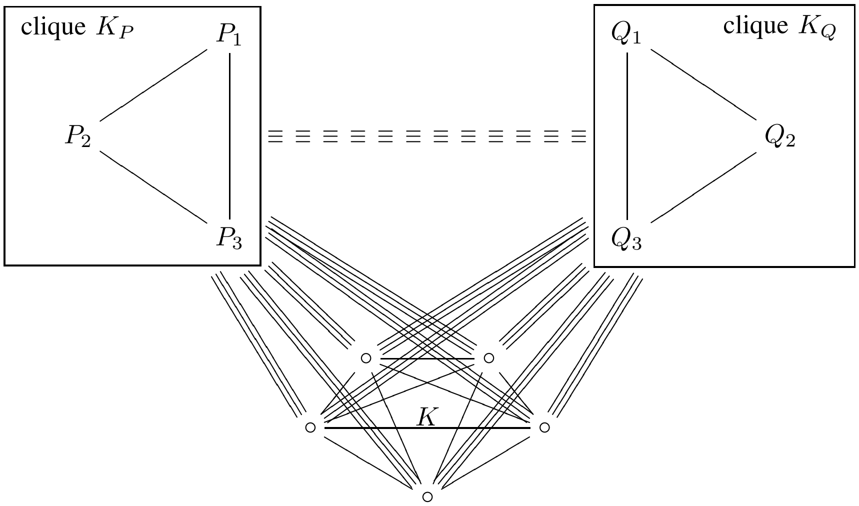

Fig. 4. Generalized defective cliques.

defective clique as shown in Figure 4. To obtain the same result as in the original approach, any pair of nodes $P_i$ and $Q_j$, belonging to the two non-overlapping components $K_P$ and $K_Q$ respectively, must be predicted as interacting, because the original algorithm would have predicted it (since it completes the defective non-maximal clique $KP_iQ_j$). The maximal size $n$ of non-overlapping sub-cliques $K_P$ and $K_Q$ is a parameter of the algorithm.

Since even the number of maximal cliques can be significant (in the hundreds of thousands for some of our experimental datasets), and their sizes can be in the hundreds of nodes, the number of comparisons between nodes in pairs of cliques in Step 2 can still be formidable in practice. We further reduce the time complexity of Step 2 by organizing the cliques (represented as strings sorted by node index) into a prefix tree. This structure allows us to reuse some comparison results among cliques sharing a common prefix of nodes.

## III. The Diffusion Distance Algorithm

As we stated earlier, two proteins interacting with the same protein cluster are likely to interact with each other. This principle is derived from the way in which pull down experiments are carried out, and the clique completion algorithm that we have just described is the literal implementation of this idea. One possible shortcoming of this method is that edges are inferred only when they are found within specific defective cliques which comply with a particular setting of the parameters of the algorithm. However, we may want to think about our principle in a softer way, and realize that ideally what we would really like is to infer an edge between two proteins whenever they are connected by *many short* paths in the graph. A possible way of implementing this idea is by considering diffusion distances.

Let us again think about our protein-protein interaction prediction problem in terms of graphs, but this time we think of a graph as a system with some dynamics. We can imagine that at any given time there are some particles on the vertices of the graph, and at each time step these particles jump from one vertex to another with certain probabilities. Since our links are binary (either an edge exists between two proteins or it is missing) we shall assign equal probability $p_{i,j}$ to each existing link going from node $i$ to node $j$: $p_{i,j} = 1/d_i$ where $d_i$ is the degree of node $i$. We can collect these probabilities into a matrix: $M = A \cdot D^{-1}$ where $A$ is the adjacency matrix of the graph and $D$ is the diagonal matrix of the degree of the nodes. This matrix is called Markov transition matrix and a path that a particle travels is called a random walk.

Given a graph with $n$ vertices, we can describe the initial position of a particle as a discrete probability distribution over the $n$ vertices, that can be written out as a vector $v_0 \in R^n$ whose components are all positive and sum to one. Then the probability distribution of the particle at the next time step is given by: $v_1 = M \cdot v_0$ and the probability after $k$ iterations is given by: $v_k = M^k \cdot v_0$. Therefore, for any initial configuration, we can compute the probability of ending up in any final configuration after a given number of steps.

We can now think of this probability as defining some kind of distance: starting from a given initial node, the higher the probability of ending up at a certain node, the smaller the distance between the two nodes. Clearly, such distance between two nodes will depend on two factors: how many different paths connect the two nodes, and how long these paths are.

This type of distance is called Diffusion Distance, and our idea is to use it in order to measure the connectivity between two nodes: when two proteins are connected by many short paths in the graph such distance would be small, and we could infer an edge between them.

It is possible to show [9] that such distance between nodes $x$ and $y$ has a simple form:

$$D_m(x, y) = \sum_{i=1}^{n} \lambda_i^m \cdot [u_i(x) - u_i(y)]^2$$

where $u_i$ and $\lambda_i$, $i = 1 \dots n$ are respectively the eigenvectors and eigenvalues of a symmetric matrix similar to $M$, and $m$ is a parameter denoting the maximum length of the Markov random walks between $x$ and $y$ which are taken into account by the measure. The algorithm for inferring protein-protein interaction then consists of two steps:

1) For each pair of proteins compute the diffusion distance between them
2) Infer that two proteins interact if their diffusion distance is lower than a certain threshold $\tau$.

Therefore, as for the clique completion algorithm, this method requires the setting of two parameters: $m$, the maximum length of the Markov random walks; and $\tau$, the threshold for the distance, below which we should infer the interaction.

## IV. Results

Here we shall present the results obtained using our two methods for inferring protein-protein interactions which had been missed by large scale experiments.

### A. *Performance of the Clique Completion Algorithm*

We applied the clique completion method to a large scale experimental dataset of the protein interaction network of *S. cerevisiae* obtained combining the results of different separate experiments by [3], [4], [5], [15]. For this organism we also had available a gold standard set of protein pairs known with high degree of confidence to be "positive" (interacting) or "negative" (noninteracting), published in [7]. The gold standard set for these tests contained 8250 positive and 2708622 negative pairs. Our idea was to use the gold standard set in order to check the performance of our algorithm at predicting protein-protein interactions from a large scale experiment.

Given their experimental and heterogeneous origin, these datasets present some complications. Firstly, the adjacency matrix for the gold standard is incomplete, in the sense that for many pairs of proteins no experiment was performed in order to verify their interaction. Secondly, the large scale dataset is also incomplete, in the sense that its adjacency matrix does not overlap perfectly with the gold standard dataset — some proteins present in the gold standard were not included in any of the large scale experiments. So we had to decide how to treat such missing datapoints and how to evaluate the performance of the algorithm.

As regards the missing values in the large scale experimental data, we assumed that in these cases the input data showed no interaction between the proteins (therefore notice that the performance of the algorithm should improve if all possible experiments were performed).

For evaluating the performance of the algorithm we used the likelihood ratio of the predicted interactions, defined in [7] as

$$L = \frac{\frac{P_+}{G_+}}{\frac{P_-}{G_-}}$$

where

$P_+$ is the number of true positives – predicted interactions which are positive in the gold standard;

$P_-$ is the number of false positives – predicted interactions which are negative in the gold standard;

$G_+$ is the total number of positive pairs in the gold standard; and

$G_-$ is the total number of negative pairs in the gold standard.

Higher values of $L$ correspond to sets of predictions having higher overlap with the positive and/or lower overlap with the negative gold standard, and generally indicate better predictors.

The initial large scale experiment graph contains 7047 edges between 2283 nodes. In this graph the Maximal Cliques algorithm found 543 maximal cliques of size at least 4. Step 2 of the algorithm, configured to search for partial overlaps of size at least $k = 5$ and non-overlapping parts of size at most $n = 3$, predicted 270 new interactions. Of these, 49 were in the gold standard set; of them 38 were positive and 11 negative, which yields a likelihood ratio of 1134.19, significantly higher than the likelihood ratios of other single features reported in [7] (essentiality, expression correlation, MIPS function, and GO biological process), which are below 400.

The chosen values of the parameters are in a "plateau" of relative stability of the results. The likelihood ratio of the predicted set was between 59.13 and 3720.94 when varying the parameters of the algorithm as follows: $k$ (the minimal overlap size) between 4 and 7, and $n$ (the maximal size of the non-overlapping parts) between 1 and 20; the number of predicted interactions was between 12 and 8993. The average running time was below 4 seconds on a desktop machine.

Taking into account the size of the predicted set, we believe its high likelihood ratio with respect to the gold standard is a strong argument for the usefulness of this method as a predictor of new interactions.

### B. *Performance of the Diffusion Distance Algorithm*

Here we show a preliminary result of the performance of the diffusion distance algorithm. We took a sub-graph of the protein interaction network of S. cerevisiae for a set of 43 proteins for which we had information about the interactions of each pair of proteins in the gold standard set. In other words, this is a subset of the gold standard set for which there were no missing values. Then we simulated the experimental noise present in the large scale experiments by randomly creating false negatives (FN) in this dataset; that is we randomly turned a certain percentage of the 1s into 0s. We then tried to see how many of these FN were fixed by running our diffusion distance algorithm. Figure 5 shows the results obtained for different level of noise, for $m = 3$, and $\tau = 0.1$.

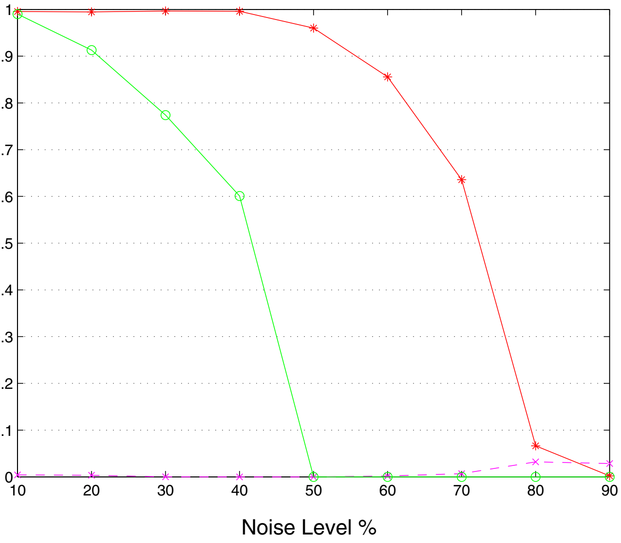

Fig. 5. Performance of the two algorithms on the 43x43 subset of the gold standard, for different values of the noise level. The continuous lines represent the ratio of the number of errors which were recovered over the total number of errors — the (red) '*' line represents the result for the diffusion distance algorithm, the (green) 'o' line represents the result for the clique completion algorithm. The (magenta) 'x' dashed line represents the ratio of the number of errors which were introduced over total size of the matrix for the diffusion distance algorithm. The clique completion algorithm never introduced any error.

We can see that the diffusion distance algorithm is able to recover a very high percentage of FN, even for very high level of noise: when 40% of all 1s are turned into 0s, the algorithm can correctly recover 99% of them; with 40% noise the algorithm still recovers about 96% of them. At the same time, almost no errors are introduced even for high level of noise.

These results are possible because in this dataset things are as anticipated by our biological hypothesis: proteins interact in complexes. Clearly when the noise level is too high, the complex structure is disrupted and therefore interactions cannot be recovered and many errors are introduced.

### *C. Comparison of the two methods*

We performed the same experiment using the clique completion algorithm, with parameters $k = 6$, $n = 17$, and results are also shown in fig.5. We can see that this algorithm also performs quite well, although it seems to be more sensitive to noise than the diffusion distance algorithm. For this experiment, the clique completion algorithm never introduced a false positive, *i.e.* when it inferred an interaction between two proteins it was always correct.

## V. Conclusion

We presented two methods for predicting new protein-protein interactions, based purely on topological properties of networks of observed interactions. Each of the two methods discussed in this paper has its advantages. The main advantage of the clique completion algorithm over the diffusion distance one, is that it can always provide an explanation of why a certain interaction has been inferred, in terms of the cliques that are completed. Also, it will never introduce a false negative. On the other hand, the diffusion distance algorithm seem to provide a better performance due to the extra flexibility afforded by such distance.

We believe that these methods, although computationally expensive, have the advantage of being more robust than other protein-protein interaction prediction methods by virtue of their independence of non-topological features such as functional classification.

## Acknowledgment

We are grateful to Ronald Jansen, David Lu, Jason Lu, and Yu Xia for many discussions and insightful suggestions.

## References

[1] G D Bader, D Betel, and C W Hogue. BIND: the Biomolecular Interaction Network Database. *Nucleic Acids Res*, 31(1):248–50, 2003.

[2] G D Bader and C W Hogue. Analyzing yeast protein-protein interaction data obtained from different sources. *Nat Biotechnol*, 20(10):991–7, 2002.

[3] A C Gavin et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868):141–7, 2002.

[4] Y Ho et al. Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. *Nature*, 415(6868):180–3, 2002.

[5] T Ito et al. Toward a protein-protein interaction map of the budding yeast: A comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. *Proc Natl Acad Sci USA*, 97(3):1143–7, 2000.

[6] R Jansen et al. Integration of genomic datasets to predict protein complexes in yeast. *J Struct Funct Genomics*, 2:71–81, 2002.

[7] R Jansen et al. A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302(5644):449–53, 2003.

[8] A Kumar and M Snyder. Protein complexes take the bait. *Nature*, 415(6868):123–4, 2002.

[9] S Lafon. *Diffusion maps and geometric harmonics*. PhD thesis, Yale University, 2004.

[10] E M Marcotte et al. Detecting protein function and protein-protein interactions from genome sequences. *Science*, 285(5428):751–3, 1999.

[11] H W Mewes et al. MIPS: a database for genomes and protein sequences. *Nucleic Acids Res*, 30(1):31–4, 2002.

[12] M Pellegrini et al. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc Natl Acad Sci USA*, 96(8):4285–8, 1999.

[13] G Rigaut et al. A generic protein purification method for protein complex characterization and proteome exploration. *Nat Biotechnol*, 17(10):1030–2, 1999.

[14] S Tsukiyama, M Ide, H Ariyoshi, and I Shirakawa. A new algorithm for generating all the maximal independent sets. *SIAM J. Comput.*, 6(3):505–17, September 1977.

[15] P Uetz et al. A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, 403(6770):623–7, 2000.

[16] C von Mering et al. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417(6887):399–403, 2002.

[17] I Xenarios et al. DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res*, 30(1):303–5, 2002.

[18] Y Xia et al. Analyzing cellular biochemistry in terms of molecular networks. *Annu Rev Biochem*, 73:1051–87, 2004.