BIOINFORMATICS ORIGINAL PAPER Vol. 22 no. 18 2006, pages 2283–2290
doi:10.1093/bioinformatics/btl370

Systems biology

Detecting functional modules in the yeast protein–protein interaction network

Jingchun Chen and Bo Yuan*

Integrated Biomedical Science Graduate Program, Department of Biomedical Informatics and Department of Pharmacology, The Ohio State University, 333 W. 10th Avenue, Columbus, OH 43210, USA

Received on April 20, 2006; revised on July 3, 2006; accepted on July 4, 2006
Advance Access publication July 12, 2006
Associate Editor: Golan Yona

ABSTRACT

Motivation: Identification of functional modules in protein interaction networks is a first step in understanding the organization and dynamics of cell functions. To ensure that the identified modules are biologically meaningful, network-partitioning algorithms should take into account not only topological features but also functional relationships, and identified modules should be rigorously validated.

Results: In this study we first integrate proteomics and microarray datasets and represent the yeast protein–protein interaction network as a weighted graph. We then extend a betweenness-based partition algorithm, and use it to identify 266 functional modules in the yeast proteome network. For validation we show that the functional modules are indeed densely connected subgraphs. In addition, genes in the same functional module confer a similar phenotype. Furthermore, known protein complexes are largely contained in the functional modules in their entirety. We also analyze an example of a functional module and show that functional modules can be useful for gene annotation.

Contact: yuan.33@osu.edu

Supplementary Information: Supplementary data are available at *Bioinformatics* online

1 INTRODUCTION

As a critical level of biology hierarchy, functional modules are cellular entities that perform certain biological functions, which are relatively independent from each other (Barabasi and Oltvai, 2004; Hartwell *et al*., 1999). Revealing modular structures in biological networks will help us in understanding how cells function (Hartwell *et al*., 1999; Bork *et al*., 2004). Many questions remain to be answered, but the detection of the functional modules is a preliminary step.

Recently a number of network partition algorithms have been designed to find community and modular structures in complex networks. On the basis of shortest-path algorithm in graph theory, Girvan and Newman generalized the concept of vertex betweenness to edges to distinguish between inter-community edges and intra-community edges. They designed an algorithm that iteratively removes the edges of the highest betweenness until a given network breaks into desired number of clusters (Girvan and Newman, 2002). Building on this work, Parisi and colleagues strengthened the definition of community and proposed a local topology-based concept of 'edge clustering coefficient' to replace the global edge betweenness measurement (Radicchi *et al*., 2004). In another study, using shortest-distance as a metric, Rives and Galitski applied a hierarchical clustering algorithm to reveal the modular organization of yeast signaling networks (Rives and Galitski, 2003). Spirin and Mirny combined clique detection, superparamagnetic clustering (SPC) and Monte Carlo optimization (MC) to search for functional modules in the yeast protein network (Spirin and Mirny, 2003). Berg and Lassig used a probabilistic model to expand the motif concept and proposed a local graph alignment algorithm to detect such probabilistic motifs in the transcription network of *Escherichia coli* (Berg and Lassig, 2004). More recently, Xiong and colleagues applied an association pattern discovery method to find the 'hyper-cliques' (functional modules) in the yeast proteome network (Xiong *et al*., 2005). One common theme shared by these work is that networks were represented as unweighted graphs. Even though they do capture essential features of many complex networks, unweighted graph representations will impose a big limitation on the study of biological networks. Protein–protein interaction networks, in particular, have a very high degree of inter-module cross-talk (Rives and Galitski, 2003), which makes it very difficult to partition them using algorithms based solely on topology. Some recent works do take this into consideration and use weighted graph representations. Shamir and his colleagues applied a biclustering algorithm to the integrated genomic data to partition the molecular network of yeast (Tanay *et al*., 2004). However, their weighting scheme is applied on the bipartite graph to represent the level of association between genes and properties, not between pairs of interacting genes. Another interesting work is from Ouzounis's group (Pereira-Leal *et al*., 2004). They first transformed the yeast protein interaction network into a line graph, and then applied a graph flow-based clustering algorithm to find functional modules. In their work, the weight of an edge represents the level of confidence attributed to that interaction, which may not indicate the functional correlation between the two proteins. In recent years high-throughput studies have generated a huge amount of functional genomic data. In particular, microarray technology has been applied to study yeast gene expressions under all kinds of conditions, and the results of these studies are centralized for public access (Ball *et al*., 2005). It is therefore highly desirable to develop new methods that would take advantages of functional genomics information and partition protein–protein interaction networks in a biologically more meaningful way.

Here we report our study on detecting the functional modules in the protein–protein interaction network of *Saccharomyces*

*To whom correspondence should be addressed

cerevisiae. Our first goal was to develop an algorithm that partitions weighted graph into communities. Our next goal was to apply this new algorithm to find functional modules in the yeast protein–protein interaction network and to rigorously validate these modules at both topological and functional level. We also wanted to assess the functional modules in the context of protein complexes and gene annotation. Our results indicate that (1) our algorithm is a useful tool in studying the modularity and organization of biological networks; (2) genes in the same functional module confer similar deletion phenotype; (3) known protein complexes are largely contained in the functional modules in their entirety and (4) module identification could be very useful for gene annotation.

2 METHODS

2.1 The protein–protein interaction network of yeast

Recently, several studies addressed the issue of confidence in the protein–protein interaction dataset of *Saccharomyces cerevisiae* that were obtained by high-throughput techniques (Uetz *et al*., 2000; Ito *et al*., 2001; Ho *et al*., 2002), assigning each interaction a confidence score (von Mering *et al*., 2002; Bader *et al*., 2004; Patil and Nakamura, 2005). We downloaded these datasets from the publishers' websites. We then selected from each of the datasets only high confidence interactions, which were then unioned together. After removing redundancy, the final dataset contains 10 899 interactions between 3409 proteins.

2.2 Weighted graph representation of the protein-protein interaction network

The protein–protein interaction network of yeast is represented as a weighted graph $G = (V, E)$. The vertices of the graph are the set of unique proteins, and therefore $|V| = 3409$. The edges of the graph are the interactions, and therefore $|E| = 10\ 899$.

To add weights to the edges, we exploited the abundant information of microarray expression profiles. A total of 265 microarray datasets were downloaded from *Saccharomyces* Genome Database (SGD). The raw data are expression change ratios. We transformed the raw score into a *Z*-score so that data from different experiments were comparable. If the expression of a given gene g in a microarray experiment m is changed by the ratio r, the normalized *Z*-score is

$$Z_g^m = \frac{(r - \mu)}{\sigma}, \tag{1}$$

where μ is the mean in that experiment and σ is the standard deviation. The edge weight is defined as the average of the *Z*-score differences over all the experiments. For a given interaction between protein i and protein j, the weight is

$$W_{i,j} = \left| \frac{1}{n} \sum_{m=1}^{n} (Z_i^m - Z_j^m) \right|, \tag{2}$$

where n is the total number of microarray experiments in the dataset. This way the weight represents the 'dissimilarity' between the expression profiles of two genes, which is the equivalent of 'distance' in graph theory.

2.3 Betweenness-based partitioning algorithm for weighted graph

Girvan and Newman first proposed the concept of edge betweenness in the context of network communities (Girvan and Newman, 2002). The idea is that inter-community edges are more likely to be on some shortest paths than intra-community edges. By computing the all-against-all shortest paths of a graph and calculating the number of times each edge is traveled, one could identify the linkers between communities. By removing these linkers step-by-step, one would eventually obtain the community structure of a graph as a hierarchical tree (Girvan and Newman, 2002). This algorithm (GN for short) is intuitively very appealing. However, not all interactions are equally important within a network. Some interactions may be used more frequently than others. With the yeast protein–protein interaction network being represented as a weighted graph, we extended the GN algorithm so that the shortest path was based on edge weights.

Besides this extension, we also modified the measurement of edge betweenness. In the GN algorithm, the betweenness of an edge is essentially the number of all-against-all shortest paths that run through it. In the example graph shown in Figure 1A, there are two subgraphs. In the left subgraph the edge CD has a betweenness of 24. This is because it is the only bridge that connects vertices A, B, C and vertices D, E, F, G, H, I, J, K, and therefore there are total $3 \times 8 = 24$ distinct all-against-all shortest paths. Similarly, in the right subgraph, the edge ST has a betweenness of 20. It can be shown that, in the whole graph, edge CD has the highest betweenness. Therefore edge CD is removed at this step. However, by simple visual inspection we tend to say that edge ST is a better candidate that connects two communities {P, Q, R, S} and {T, U, V, W, X}, and that the left subgraph is a separate community. From the topological point of view, the original definition of betweenness may lead to unbalanced partitioning under certain circumstances.

To resolve this issue we introduced the idea of 'non-redundancy' into the computation of edge betweenness. When counting the number of shortest paths for an edge, the end points must be distinct. For example, when counting the shortest paths that go through edge ST, if path P–T is counted, no other path that starts or ends with P (P–U, P–V, P–W, P–X) or T (T–Q, T–R, T–S) should be counted (Fig. 1B). Based on this idea, the betweenness of an edge is the maximum number of non-redundant all-against-all shortest paths passing through it. We expected this change to keep the intuitiveness of the original algorithm, while making it more robust against unbalanced partition.

For implementation of this modification we used the Maximum Bipartite Matching (MBM) algorithm. Following the Floyd–Warshall algorithm, all the shortest paths passing through the given edge are identified. Then the end vertices of all the paths are divided into two groups, depending on which side each vertex sits with respect to the given edge. A bipartite graph is constructed on the two vertex groups. Each shortest path is converted to an edge in the bipartite graph. Finally, the MBM algorithm is applied to find the maximum matching number, which is the betweenness of the given edge. In the example shown in Figure 1, edge CD has betweenness of 3, and edge ST has betweenness of 4 (Fig. 1B). Therefore edge ST is removed to give a more meaningful result.

2.4 Quantitative definition of community

Communities, or modules, have been loosely referred to as 'densely connected subgraphs'. However, many quantitative definitions for this concept exist in the literature (Radicchi *et al*., 2004). For simplicity we use the term 'in-degree' (k_{in}) of a vertex to represent the number of its within-subgraph connections, and we use the term 'out-degree' (k_{out}) to represent the number of its outside-subgraph connections. Please note that these are not the notations used in a directed graph to denote incoming and outgoing edges. A necessary condition for a subgraph to be called a module is that the sum of in-degrees of all the vertices in the subgraph is greater than the sum of out-degrees. This is a weak definition. A much stronger definition requires that for every vertex in the subgraph the in-degree is larger than the out-degree. We think that this later definition is too stringent for a real-life network, which may have many complicated crosstalks between modules. Furthermore, even if the modularity of a network is so clear-cut that every module satisfies such a strong definition, the algorithm has to be perfect to actually find the modules.

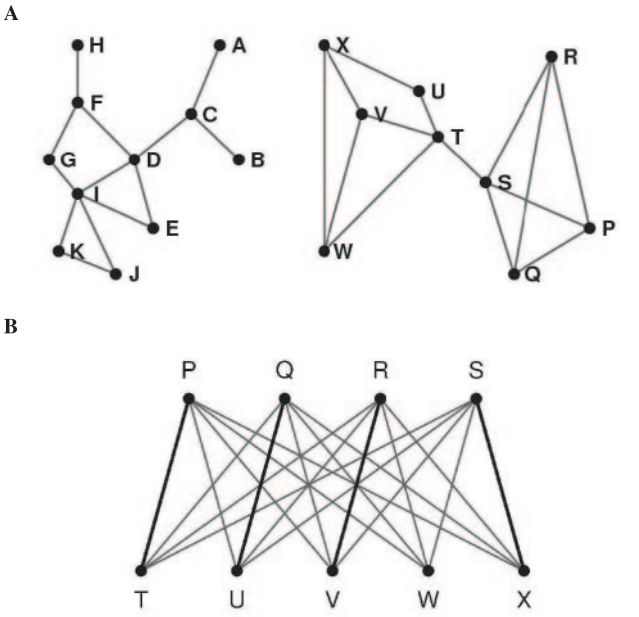

Fig. 1. Edge betweenness based on non-redundant shortest path. (**A**) An example graph containing two subgraphs. (**B**) A bipartite graph representing all the shortest paths passing through edge ST. Vertices P, Q, R, S are the end vertices on one side of the edge and vertices T, U, V, W, X are on the other side. An edge is drawn between two vertices if there is a shortest path between them that passes through ST. One set of non-redundant paths is shown by the four dark edges P–T, Q–U, R–V and S–X.

Here we propose a quantitative definition of community that we believe is both strong and practical. Let k_{in} and k_{out} be the in-degree and out-degree of a vertex, respectively. A subgraph of n vertices is a module if

$$\sum_{i=1}^{n} k_{in}^{i} > \sum_{i=1}^{n} k_{out}^{i} \tag{3}$$

$$\{k_{in}^{1}, k_{in}^{2}, \ldots, \pm k_{in}^{n}\} \gg \{k_{out}^{1}, k_{out}^{2}, \ldots, k_{out}^{n}\}. \tag{4}$$

The first criterion is the weak definition as stated above. The second criterion states that, collectively, the in-degrees of the vertices in the subgraph are significantly greater than the out-degrees. This is less stringent than the strong definition, but the definition still captures the essence of the concept 'densely connected subgraph'. In implementing this second criterion, we used the Wilcox two-sample test to compare the in-degrees and out-degrees, and we used p-value of 0.01 as the cutoff value for significance.

2.5 Computer-generated graphs

The artificial graphs with known community structures were generated exactly as the examples in the original GN-algorithm paper. Briefly, each graph contains 128 vertices that are divided into 4 communities, each of which contains 32 vertices. Between each pair of vertices an edge is added with certain probability. The probability is p_{in} if the two vertices are within the same community, and p_{out} if the two vertices belong to different communities. The p_{out} is varied to produce graphs with different levels of crosstalks. The higher the p_{out} is, the more crosstalks exist between communities. The probability p_{in} is chosen accordingly so that the average number of connections per vertex is 16.

2.6 Functional module datasets from previous studies

For the purpose of comparison, we obtained two sets of functional modules that were identified in two previous studies. One dataset was kindly provided by Victor Spirin and was retrieved from http://insilico.mit.edu/modules/allOurClusters.html (Spirin and Mirny, 2003). The other dataset was retrieved from http://www.cs.tau.ac.il/%7Ershamir/samba/ (Tanay *et al*., 2004).

3 RESULTS

3.1 Algorithm test on computer-generated graphs

We first tested our algorithm on artificial graphs produced exactly as the examples in the original GN-algorithm paper. For each graph, we applied our algorithm until four communities were obtained. We then compared the obtained communities with the actual structure and calculated the fraction of vertices classified correctly. As shown in Figure 2, our algorithm could correctly find the community structures in simple networks, but it started to make more mistakes when the community structures became more complicated. Interestingly, we found that for simple networks, our algorithm tended to make slightly more mistakes than the GN algorithm. But for networks with more complicated structures, our algorithm outperformed the GN algorithm. These results suggest that the extension and modifications we made on the GN algorithm make it more robust against noise and the blurring of community boundaries. Since community structures in real-life networks are usually very complex, our extended algorithm may produce more meaningful results in real applications.

3.2 The partition of the protein–protein interaction network of yeast

Next we applied the algorithm to the protein–protein interaction network of yeast. We took note of the perspectives of Hartwell *et al*. (1999) and Spirin and Mirny (2003), and let the algorithm terminate when no subgraph had more than five vertices. We then applied the definition of module to these candidate subgraphs and obtained 266 functional modules. Out of the 3409 proteins in the network, 3150 (92.4%) are included in these modules. This indicates a good coverage among the functionalities of the yeast, and a good sensitivity that is probably the result of the combination of the modified algorithm and the proposed filtering criteria. The module sizes range from 5 to 98, 56.2% of which fall within 5–25, a size range proposed by Spirin and Mirny (2003). A list of these modules is available as Supplementary data (Table S1), along with a preliminary annotation based on Gene Ontology (GO) and some results relevant to validations (see below).

3.3 Validation through connectivity density

We first assessed the validity of the obtained functional modules from the topology perspective. For this purpose, we propose a simple measurement, the connectivity density. The connectivity density of a subgraph is the ratio of total in-degrees to the total number of connections. Obviously the density of a subgraph is always between 0 and 1 and, according to the weak definition, the density of a module should be between 0.5 and 1. The lower the density, the less likely a module. Next, we asked which control to use for comparison. For a given module, a simple control would be a set of the same number of proteins randomly picked up in the network. However, a random set of proteins are unlikely to be connected to each other, and therefore such a comparison is not very convincing. Instead, we used a more rigorous control. For each functional module, we randomly replaced a small portion of the

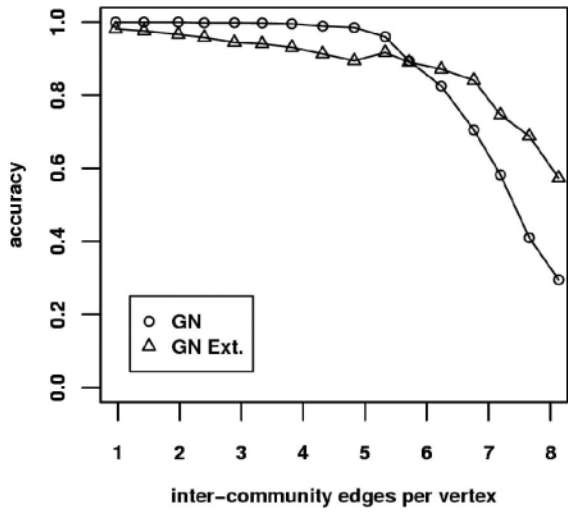

Fig. 2. Algorithm performance on artificial networks. The *x*-axis represents the complexity of crosstalks between the communities in a network. The *y*-axis is the percentage of vertices that are correctly classified. GN is the abbreviation of the original algorithm and GN Ext. is the algorithm used in this study.

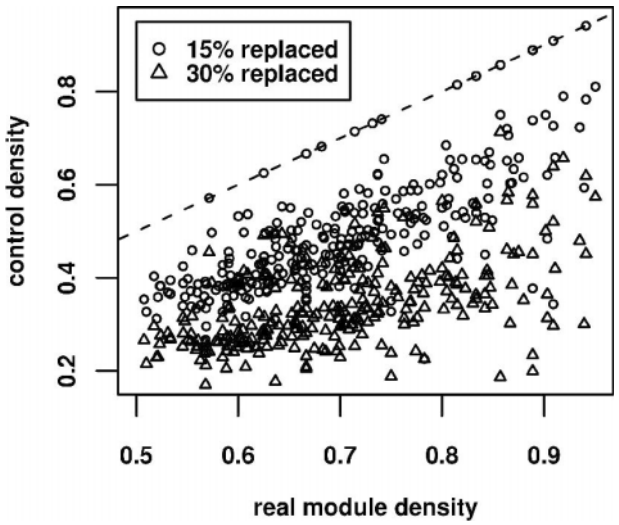

Fig. 3. Identified functional modules are densely connected subgraphs. In this scatter plot, each data point represents the connectivity density of a functional module (*x*-axis) and its replacement control (*y*-axis). The dashed line is $y = x$, which means that the connectivity density is the same for the module and its control. Any data point above the line corresponds to the case where controls have higher connectivity density, while data points below the line represent the case where control has lower connectivity density than the actual functional module. The further below the line, the less dense the control is compared with the original module. Each datapoint is the average of 20 randomization experiments.

proteins in the module with the same number of proteins outside this module. The replacement proteins are connected with the proteins in the module but do not belong to it. In this way, the control is guaranteed to be connected. Comparison to such controls is equivalent to asking the question, if we shift the module a little, do we get a less connected or more connected subgraph?

Figure 3 is a scatter plot of the connectivity densities of the functional modules and their controls. For most of the modules, 15% component replacement causes the connectivity density to decrease significantly. For many of them, the density drops below 0.5, suggesting that they do not even qualify for functional modules anymore. If more proteins are replaced (30%), the connectivity densities decrease even more. The replacing experiment was repeated 20 times, each time a different set of the proteins was randomly replaced. These observations suggest that the identified modules are indeed densely connected local subgraphs, and thus are good candidates for functional modules in the yeast protein network.

3.4 Genes in the same functional module confer similar phenotype

Since a functional module performs a relatively independent cellular function, a similar phenotype is expected to appear if the genes in the same module are knocked out. To verify this, we represented each gene's phenotype as a vector of 31 dimensions, which correspond to the 31 experimental conditions (Giaever *et al*., 2002). We used the Euclidean distance of two vectors to represent the phenotype difference between two genes, and we used the average difference of all gene pairs to represent the phenotype divergence of a module. Figure 4A shows the distribution of the phenotype divergence of all the functional modules. For 185 out of 254 (72.8%) functional modules, the phenotype divergence is lower than the average phenotype difference over all the yeast open reading frames (ORFs). In other words, genes in the same functional modules display more similar phenotypes than those in different functional modules. To exclude the possibility of artifact owing to module size difference, we also checked the relationship between module size and phenotype divergence, and found no significant correlation (Fig. 4B). This is also confirmed by Pearson correlation analysis ($r = 0.01$).

To further confirm these observations, we did 20 randomization experiments where 30% of the proteins were replaced in each functional module to generate controls, and phenotype divergence was compared between each module and its control. We found that for 60.9% of the functional modules, randomization increases phenotype divergence. Overall, the controls have higher phenotype divergence than the original modules (p-value < 0.001). Altogether, these results suggest that most of the functional modules we found are not only topologically meaningful, but they are also biologically significant.

We noted that phenotype similarity is also shown to be correlated with functional similarities between genes in another study (Gunsalus *et al*., 2005), which also supports its use for validating biological significance. To further evaluate our method, we compared the phenotype divergence of our results with that of two previous studies, which used a biclustering method on integrated functional genomics data (Tanay *et al*., 2004) or a mixture of three different methods (Spirin and Mirny, 2003). We found that the phenotype divergences of the modules in this study are comparable with that of the mixed methods (p-value $= 0.46$), both of which are significantly lower than that of the biclustering method (p-values $<$ 0.002) (Supplementary data, Figure S1). It is worth noting that in

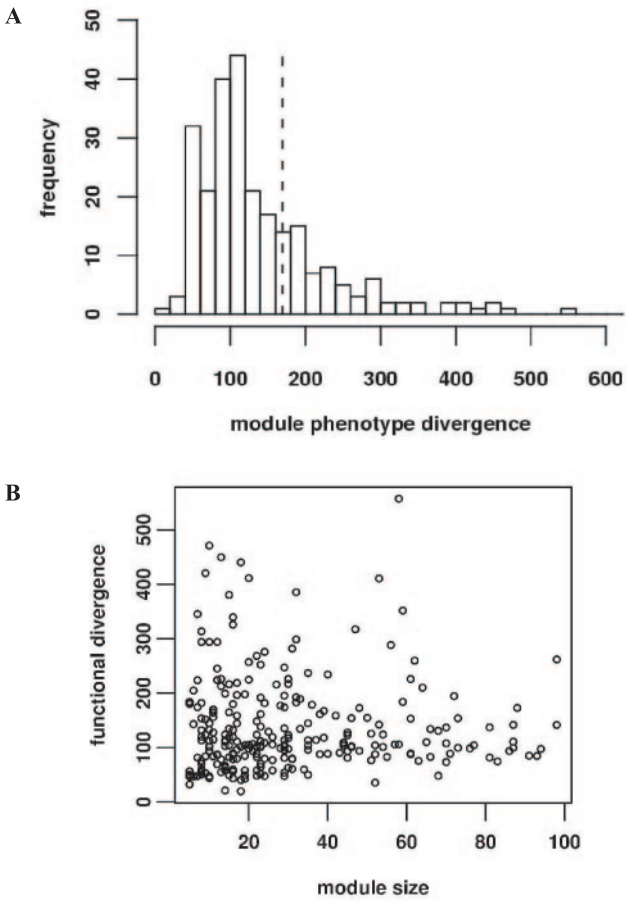

Fig. 4. Genes within a functional module confer similar deletion phenotype. (**A**) Histogram of the phenotype divergence of the functional modules. The dashed line indicates the phenotype difference averaged over all pairs of the yeast genes. (**B**) The scatter plot of phenotype divergence and module size. The Pearson correlation coefficient is 0.01.

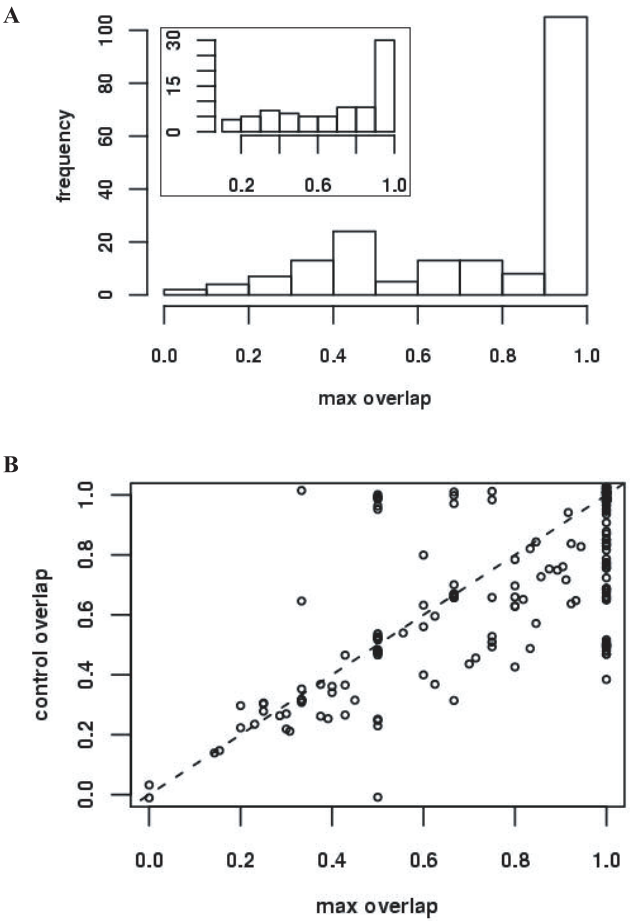

Fig. 5. Protein complexes are contained in functional modules. Protein complexes annotated by CYGD were matched against the identified functional modules. The overlap between each complex and a functional module is identified and the ratio of overlap to complex size was calculated. (**A**) Histogram of the maximum overlap ratios of all the complexes. The inset shows the histogram for the subset of the protein complexes of five or more components. (**B**) Scatter plot of the complex overlaps of the modules and their controls. Each data point represents the complex overlap ratio with the actual modules (x-axis) and with the control modules (y-axis). The dashed line is $y = x$, representing the cases where the overlap ratios remain unchanged. The data points above the line represent the cases where complexes match better with the controls; data points below the line represent the cases where complexes match better with the actual functional modules.

this study 266 functional modules were detected, which is more than that found by the biclustering method (205) and is three times as many as that found by the mixed techniques (90). This again suggests that our algorithm is capable of finding biologically relevant functional modules.

3.5 Known protein complexes are largely contained in the functional modules

A protein complex is an aggregate of multiple proteins that interact with each other and perform certain biological activities (Gingras *et al*., 2005). Since this is conceptually very similar to the definition of a functional module, we asked whether our algorithm could detect protein complexes in their entirety, or whether they would be randomly divided into fragments during partitioning. First, we matched each protein complex against the identified functional modules and calculated the maximum overlap between each complex and the functional modules. As shown in Figure 5A, majority of the 194 protein complexes annotated by the Comprehensive Yeast Genome Database (CYGD) at MIPS are largely contained in the functional modules we found (overlap > 0.75). A total of 98 protein complexes (51%) were identified in their entirety by our algorithm. Knowing that small protein complexes are likely to be contained in large functional modules by chance, we applied this analysis to large protein complexes. Of the 78 complexes that contain 5 or more proteins, 45 are largely contained in the functional modules, and 23 were identified completely. Similar results were obtained by analyzing the protein complex dataset annotated by the SGD (Supplementary data, Figure S2).

To further confirm these results, we applied the overlapping analysis against the control modules obtained by replacing 15% of the module components. Then, for each of the protein complexes in the CYGD database, we compared the overlap ratios before and after

the replacement. As shown in Figure 5B, for the majority of the protein complexes (107 out of 194), the overlap ratios with the control modules are lower than the overlap with the actual functional modules. The overlap ratios remain unchanged for about one-third of the complexes (67 out of 194). Only a very small portion of the complexes (20 out of 194) are better overlapped with the controls. In addition, the number of completely contained complexes decreases to 73. It should be noted that in the controls only a small portion (15%) of the module components are replaced, while the overlapping with the complexes changed significantly. These analyses show that known protein complexes are largely contained in the functional modules we found, and many of them are identified completely, without a single component missing. This further indicates that our algorithm is capable of detecting functional modules that are biologically meaningful.

3.6 The chromosome segregation functional module: an example

Figure 6 shows an example of one of the functional modules we identified in the protein–protein interaction network of yeast. This functional module has 18 proteins. In the CYGD annotation, the genes *Nuf*2, *Spc*25, *Dam*1, *Duo*1, *Tid*3, *Spc*34, *Dad*1, *Dad*2, *Ask*1, *Spc*24 and *Spc*19 are annotated as playing a role in chromosome segregation, or spindle pole body, or both. *Ulp*1 and *Smc*5 are annotated as playing a role in mitotic cell division. *Cnn*1 is annotated with meiosis. Similar annotations are given in the Gene Ontology (GO) database. This functional module is obviously the core machinery responsible for the separation of chromosomes. Out of the 18 genes, 14 have a lethal deletion phenotype. This is consistent with the fact that chromosome segregation is a housekeeping process for budding yeast, just like for any other organism.

As discussed above, this functional module contains two protein complexes. NUF2, TID3, SPC24 and SPC25 form the highly conserved Ndc80 protein complex. This complex is the core of kinetochore (Asakawa *et al*., 2005), and is responsible for proper alignment and attachment of chromosomes (Wei *et al*., 2005). DAM1, DAD1, DAD2, DUO1, ASK1, SPC19 and SPC34 form the DAM1–DUO1 protein complex. This complex is a ring-shaped interface between microtubule and kinetochore, and it is capable of translating the force generated by microtubule depolarization into movement to facilitate chromosome segregation (Westermann *et al*., 2006, 2005). It is interesting to note that each of the two protein complexes contains a complete subgraph (clique) of size 4. Since functional modules are densely connected subgraphs, cliques are indeed expected to appear more frequently.

*YLR*419*w* is the only component in this module that has no definite functional annotation in either CYGD or SGD. The most updated SGD and GO annotation regard it as a hypothetical protein with ATP-dependent helicase activity, based on the homology of a small portion of the amino acid sequence. Likewise, in CYGD it is called a putative helicase. Based on the fact that YLR419W is an integral component of this functional module, we predicted its biological function to be chromosome segregation. A number of lines of evidence are in line with this prediction. First of all, evolutionary studies showed that this gene belongs to a family of helicase with very diverse functions, many of which has multiple functions (Sanjuan and Marin, 2001). Second, overexpression of a dominant negative form of RHA, a RNA helicase, causes aberrant mitosis

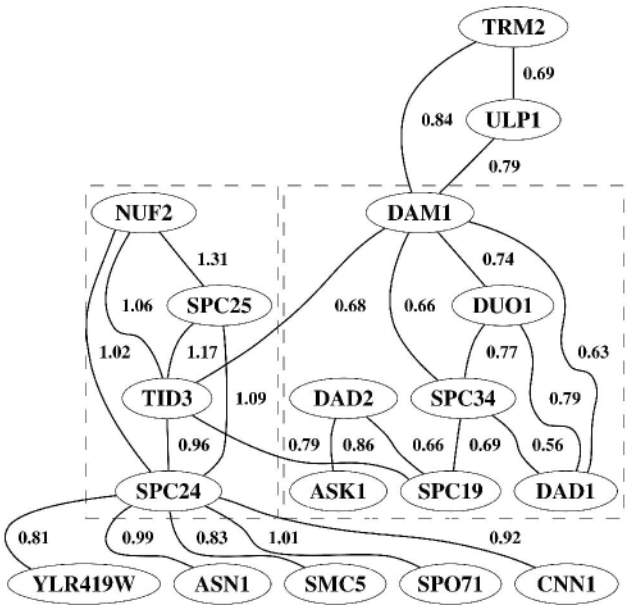

Fig. 6. The chromosome segregation module. This module contains 18 proteins, most of which are annotated with chromosome segregation or cell division functions. The two dashed boxes indicate the two protein complexes, each of which contains a complete subgraph of four vertices (C_4). The label on each edge represents the weight, i.e. the expression dissimilarity between the two genes. This diagram was generated using the DOT program of the graph visualization package Graphviz.

with extra centrosome and tetraploidy in human breast epithelial cells, suggesting its role in centrosome formation and chromosome segregation (Schlegel *et al*., 2003). In addition, RUVBL1/TIP49a, a human ATP-dependent helicase, was shown to associate with tubulins and colocalize with centrosome and mitotic spindle (Gartner *et al*., 2003).

4 DISCUSSION

In this study, we first integrated diverse datasets and represented the interaction network of *Saccharomyces cerevisiae* as an undirected weighted graph. Then, on the basis of a betweenness-based algorithm, we developed a partition algorithm for weighted graphs, and identified 266 functional modules in the yeast protein–protein interaction network. We validated these functional modules by exploring the relationship between module topology and gene phenotype and the relationship between protein complexes and functional modules.

The protein–protein interaction network of yeast used here was obtained through integrating the high confidence datasets from three rigorous studies. With 10 899 interactions between 3409 proteins, this network is very complex. Uncovering the modular structure of such a network is a challenging task. To make things worse, not all the interactions are stable, and not all the interactions occur at the same time. In other words, the network is not a real snapshot of the interactions in yeast, but an overlap of many snapshots. How much confidence do we have with the results obtained from such a network? Our study addressed both issues by using expression dissimilarity as the weights for the interactions. First of all, adding

weights based on domain knowledge to represent the strengths of connections can enhance the network analysis (Barrat *et al*., 2004; Newman, 2001). Second, the algorithm is shortest-path based, which makes the use of weight highly desirable. In the case of this study, in order to obtain functional modules with biological significance, it is highly desirable to incorporate functional genomics information into the partitioning process. The expression dissimilarity was computed by taking into consideration hundreds of microarray expression profiles. If the distance between two interacting proteins is very small, it means that the two corresponding genes' expression profiles are very similar. In other words, they are co-regulated. Those unstable, transient interactions will probably have a larger distance owing to less correlated expression profiles. Therefore by using expression dissimilarity as weight we emphasize the functional correlations between interacting partners. The community structures obtained in such a weighted graph very likely represent real functional modules.

The betweenness-based partitioning algorithm was proved to be intuitive and powerful in module detection in real world networks (Girvan and Newman, 2002). In this study we developed an extended algorithm to partition weighted graphs, which can be used on other types of networks. For example, expression profiles can also be used to add weights to graphs representing transcriptional networks (Ihmels *et al*., 2002), and our algorithm can be used on such datasets to identify regulatory modules in yeast or other organisms. We note that a number of algorithms have recently been developed to study the modularity in biological networks, which are summarized in the introduction. Owing to certain limitations, we were only able to do a limited comparison with two of those studies. Ideally a more comprehensive analysis of these methods, such as a competition style study, will greatly benefit the community by guiding future investigations in this field.

By comparing the pairwise phenotype difference we showed that, in general, genes confer a similar deletion phenotype if their protein products belong to the same functional module. This result has significant relevance for showing that biology is modular. Even though various methods have been developed to detect functional modules at the topological level, it is the modularity at the functional and the phenotypical level that interests us. Our results indicate that phenotype similarity could be used to evaluate the biological significance of functional modules detected using topological features. Furthermore, with more detailed analysis, phenotype data can be very valuable resources for understanding biological processes. For example, certain treatments may cause growth defects among the deletion mutants for a given module. This would be a strong evidence that this module is involved in the cellular response to these treatments.

In this study, we used the topological relationship between protein complexes and functional modules to validate the biological relevance of the modules we identified. An interesting question that remains to be answered is, what is the functional relationship between protein complexes and functional modules? Our preliminary analysis did not find significant correlation between a module's phenotype divergence and the percentage of its components involved in protein complex formation (data not shown). Of course, phenotype is just one of many ways to assess the functional significance of the modules. Further studies are needed to address this important issue to help us understand the internal organizations of functional modules.

In this study, we showed an example of predicting gene function based on the associated functional module. The yeast gene *YLR*419*w* is not annotated in any detail owing to lack of significant homology to any known gene. However, through functional module classification we are able to predict its biological function to be chromosome segregation. In recent years, whole genome sequencing projects have generated a huge amount of DNA sequence information, and various sophisticated gene finding algorithms have been applied to find large numbers of ORFs. Even though homology-based gene annotation provides the first clue to the biological functions of new ORFs, functional genomics-based annotation methods have become increasingly important (Troyanskaya, 2005; Bentley, 2000). This is particularly true for those ORFs with poor or no homology. Our study showed that functional module detection could be yet another complimentary method for gene annotation.

ACKNOWLEDGEMENTS

The authors wish to thank V. Spirin, L. A. Mirny, A. Tanay, R. Sharan, M. Kupiec and R. Shamir for sharing their data with them. The authors also wish to thank T. Dowling, N. Robertson, S. Gibbs, C. Q. Zhang and Y. Xu for stimulating discussions on graph theory. The authors are very grateful to two anonymous reviewers, both of whom helped improve the manuscript substantially.

Conflict of Interest: none declared.

REFERENCES

Asakawa,H. *et al*. (2005) Dissociation of the nuf2–ndc80 complex releases centromeres from the spindle-pole body during meiotic prophase in fission yeast. *Mol. Biol. Cell.*, **16**, 2325–2338.

Bader,J.S. *et al*. (2004) Gaining confidence in high-throughput protein interaction networks. *Nat. Biotechnol.*, **22**, 78–85.

Ball,C.A. *et al*. (2005) The stanford microarray database accommodates additional microarray platforms and data formats. *Nucleic Acids Res.*, **33**, D580–D582.

Barabasi,A.L. and Oltvai,Z.N. (2004) Network biology: Understanding the cell's functional organization. *Nature Rev. Genet.*, **5**, 101–113.

Barrat,A. *et al*. (2004) The architecture of complex weighted networks. *Proc. Natl. Acad. Sci. USA*, **101**, 3747–3752.

Bentley,D.R. (2000) The human genome project–an overview. *Med. Res. Rev.*, **20**, 189–196.

Berg,J. and Lassig,M. (2004) Local graph alignment and motif search in biological networks. *Proc. Natl. Acad. Sci. USA*, **101**, 14689–14694.

Bork,P. *et al*. (2004) Protein interaction networks from yeast to human. *Curr. Opin. Struct. Biol.*, **14**, 292–299.

Gartner,W. *et al*. (2003) The ATP-dependent helicase RUVBL1/TIP49a associates with tubulin during mitosis. *Cell. Motil. Cytoskeleton.*, **56**, 79–93.

Giaever,G. *et al*. (2002) Functional profiling of the Saccharomyces cerevisiae genome. *Nature*, **418**, 387–391.

Gingras,A.C. *et al*. (2005) Advances in protein complex analysis using mass spectrometry. *J. Physiol.*, **563**, 11–21.

Girvan,M. and Newman,M.E. (2002) Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA*, **99**, 7821–7826.

Gunsalus,K.C. *et al*. (2005) Predictive models of molecular machines involved in *Caenorhabditis elegans* early embryogenesis. *Nature*, **436**, 861–865.

Hartwell,L.H. *et al*. (1999) From molecular to modular cell biology. *Nature*, **402**, C47–C52.

Ho,Y. *et al*. (2002) Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry. *Nature*, **415**, 180–183.

Ihmels,J. *et al*. (2002) Revealing modular organization in the yeast transcriptional network. *Nature Genet.*, **31**, 370–377.

Ito,T. *et al*. (2001) A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci. USA*, **98**, 4569–4574.

Newman,M.E. (2001) Scientific collaboration networks II. Shortest paths, weighted networks, and centrality. *Phys. Rev. E Stat. Nonlin. Soft. Matter Phys.*, **64**, 016132.

Patil,A. and Nakamura,H. (2005) Filtering high-throughput protein–protein interaction data using a combination of genomic features. *BMC Bioinformatics*, **6**, 100.

Pereira-Leal,J.B. *et al.* (2004) Detection of functional modules from protein interaction networks. *Proteins*, **54**, 49–57.

Radicchi,F. *et al.* (2004) Defining and identifying communities in networks. *Proc. Natl. Acad. Sci. USA*, **101**, 2658–2663.

Rives,A.W. and Galitski,T. (2003) Modular organization of cellular networks. *Proc. Natl. Acad. Sci. USA*, **100**, 1128–1133.

Sanjuan,R. and Marin,I. (2001) Tracing the origin of the compensasome: Evolutionary history of DEAH helicase and MYST acetyltransferase gene families. *Mol. Biol. Evol.*, **18**, 330–343.

Schlegel,B.P. *et al.* (2003) Overexpression of a protein fragment of rna helicase a causes inhibition of endogenous brca1 function and defects in ploidy and cytokinesis in mammary epithelial cells. *Oncogene*, **22**, 983–991.

Spirin,V. and Mirny,L.A. (2003) Protein complexes and functional modules in molecular networks. *Proc. Natl. Acad. Sci. USA*, **100**, 12123–12128.

Tanay,A. *et al.* (2004) Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genome wide data. *Proc. Natl. Acad. Sci. USA*, **101**, 2981–2986.

Troyanskaya,O.G. (2005) Putting microarrays in a context: Integrated analysis of diverse biological data. *Brief. Bioinform.*, **6**, 34–43.

Uetz,P. *et al.* (2000) A comprehensive analysis of protein–protein interactions in *Saccharomyces cerevisiae*. *Nature*, **403**, 623–627.

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

Wei,R.R. *et al.* (2005) Molecular organization of the ndc80 complex, an essential kinetochore component. *Proc. Natl. Acad. Sci. USA*, **102**, 5363–5367.

Westermann,S. *et al.* (2005) Formation of a dynamic kinetochore- microtubule interface through assembly of the dam1 ring complex. *Mol. Cell.*, **17**, 277–290.

Westermann,S. *et al.* (2006) The dam1 kinetochore ring complex moves processively on depolymerizing microtubule ends. *Nature*, **440**, 565–569.

Xiong,H. *et al.* (2005) Identification of functional modules in protein complexes via hyperclique pattern discovery. *Pac. Symp. Biocomput.*, 221–232.