BIOINFORMATICS ORIGINAL PAPER
Vol. 23 no. 2 2007, pages 207–214
doi:10.1093/bioinformatics/btl562

Systems biology

Modular organization of protein interaction networks

Feng Luo[1,3,*], Yunfeng Yang[2], Chin-Fu Chen[3], Roger Chang[5], Jizhong Zhou[4] and Richard H. Scheuermann[5]

[1]Department of Computer Science, 100 McAdams Hall, Clemson University, Clemson, SC 29634-0974, [2]Environmental Sciences Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, [3]Department of Genetics and Biochemistry, 100 Jordan Hall, Clemson, SC 29634, [4]Insitute for Environmental Genomics and Department of Botany and Microbiology, University of Oklahoma, Norman, OK 73019 and [5]Department of Pathology, U.T. Southwestern Medical Center, 5323 Harry Hines Boulevard Dallas, TX 75390-9072

Received on May 12, 2006; revised on October 30, 2006; accepted on November 2, 2006

Advance Access publication November 8, 2006
Associate Editor: Satoru Miyano

ABSTRACT

Motivation: Accumulating evidence suggests that biological systems are composed of interacting, separable, functional modules. Identifying these modules is essential to understand the organization of biological systems.

Result: In this paper, we present a framework to identify modules within biological networks. In this approach, the concept of degree is extended from the single vertex to the sub-graph, and a formal definition of module in a network is used. A new agglomerative algorithm was developed to identify modules from the network by combining the new module definition with the relative edge order generated by the Girvan-Newman (G-N) algorithm. A JAVA program, MoNet, was developed to implement the algorithm. Applying MoNet to the yeast core protein interaction network from the database of interacting proteins (DIP) identified 86 simple modules with sizes larger than three proteins. The modules obtained are significantly enriched in proteins with related biological process Gene Ontology terms. A comparison between the MoNet modules and modules defined by Radicchi *et al.* (2004) indicates that MoNet modules show stronger co-clustering of related genes and are more robust to ties in betweenness values. Further, the MoNet output retains the adjacent relationships between modules and allows the construction of an interaction web of modules providing insight regarding the relationships between different functional modules. Thus, MoNet provides an objective approach to understand the organization and interactions of biological processes in cellular systems.

Availability: MoNet is available upon request from the authors.

Contact: luofeng@cs.clemson.edu

Supplementary information: Supplementary Data are available at *Bioinformatics* online.

1 INTRODUCTION

System level understanding of biological organization is a key objective of the post-genomic era. Accumulating evidence suggests that biological systems are composed of interacting modules of individual components (Barabasi and Oltvai, 2004; Hartwell *et al*., 1999; Ravasz *et al*., 2002; River and Galitski, 2003). With the recent advance in high-throughput experimental technologies, more and more large-scale biological networks are being defined. Identifying the modular structure of these biological networks is important to understand the organization and interaction of the cellular processes they represent. Here, we present a new framework for exploration of the modular organization in protein interaction networks.

Previous studies of protein interaction networks have focused on detecting highly connected protein clusters [e.g. Fig. 1B (1)] (Snel *et al*., 2002; Spirin and Mirny, 2003; Wilhelm *et al*., 2003; Bader and Hogue, 2003; Bu *et al*., 2003; Xiong *et al*., 2005; Chen and Yuan, 2006). However, these approaches neglect many peripheral proteins that connect to the core protein clusters with few links, even though these peripheral proteins may represent true interactions that have been experimentally verified. In addition, biologically meaningful protein modules that do not have highly connected topologies are ignored by these approaches. Furthermore, protein clusters detected by these approaches are usually isolated from each other. Thus, it is not possible to obtain relationship among clusters.

Recently, clustering methods (Pereira-Leal *et al*., 2004; Arnau *et al*., 2005) have been applied to protein interaction networks to identify biological modules. Application of clustering analysis to protein interaction networks usually involves transforming them into weighted networks. Pereira-Leal *et al*. (2004) proposed an approximate solution to weight a protein interaction based on the number of experiments that support the interaction. River and Galitski (2003) and Arnau *et al*. (2005) weighted the distance between two proteins by the length of the shortest path between them. However, this approach usually generates many identical distances and leads to a 'tie in proximity' (MacCuish *et al*., 2001) problem during hierarchical clustering. Arnau *et al*. applied the hierarchical algorithm iteratively to eliminate the 'tie in proximity' problem. However, repetitive hierarchical clustering may not be computationally feasible for a large protein interaction networks at a whole-genome level.

Alternative approach for identifying modules in networks is to divide the network into sub-networks, and then to identify modules based on their topology. Girvan and Newman (2002) proposed the concept of edge betweenness, which is defined as the number of shortest paths between all pairs of vertices that run through the edge. Edges between modules tend to have more shortest paths running

[*]To whom correspondence should be addressed.

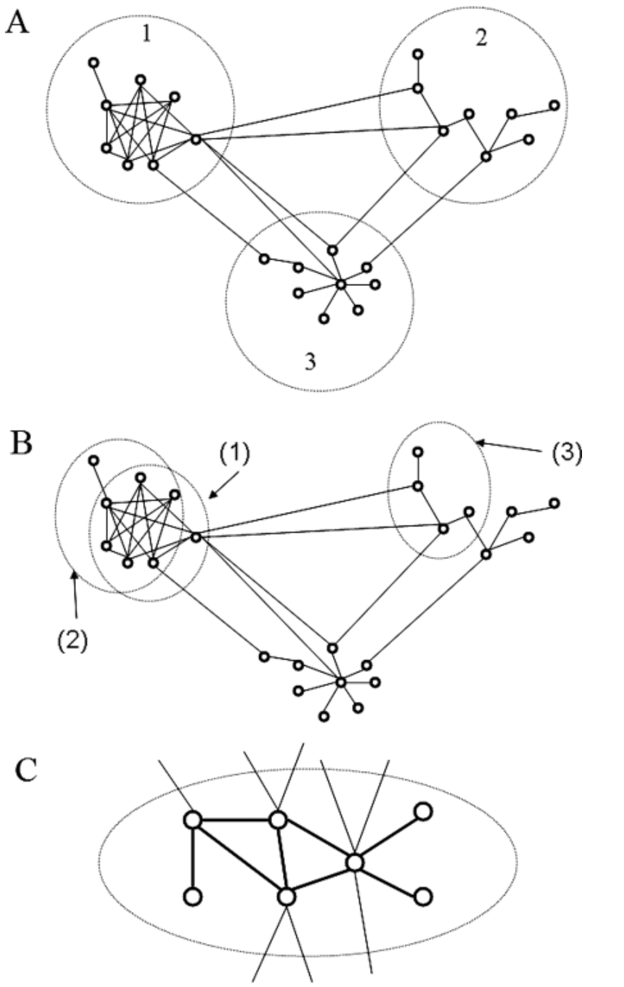

Fig. 1. Sample network topologies. (**A**) A sample network including three different topological modules (1–3), which are intuitively separated by the gray circles. (**B**) Modules identified from this sample network using previously described approaches: (1) highly connected networks; (2) strong modules (Radicchi *et al*., 2004); (3) weak modules (Radicchi *et al*., 2004). (**C**) An example of a sub-network that would be defined as a strong module, which is not a MoNet module.

through them than do edges inside modules, and thus have higher betweenness values. The deletion of edges with high betweenness can separate the network, while keeping the modular structure in the network intact. By gradually removing the edge with the highest betweenness value, Girvan-Newman proposed a divisive algorithm (G-N algorithm) to construct a tree to find community structures in an unweighted and undirected network. As the original G-N algorithm does not include a clear definition of module, it does not formally determine which parts of the tree are modules. Radicchi *et al*. (2004) combined the G-N division process with two new module definitions and gave a new self-contained algorithm to identify modules from a network. However, their module definitions do not capture some topologies that appear in natural module (see detail in section 2). Newman (2004) and Guimera and Amaral (2005) very recently proposed global optimization algorithms to partition the network into modules. Each of these algorithms requires complete information about the whole network. However, as many of the currently available biological networks derived from high-throughput experimentation are incomplete, global optimization algorithms may not be applicable for existing biological networks.

In this paper, we extend the concept of degree from vertices to sub-graphs and propose a new formal definition of a module in a network. By combining this new module definition with the relative edge order generated by the G-N algorithm, we proposed a new agglomerative algorithm to identify modules in the network. This approach has been implemented in a JAVA-based application termed MoNet.

2 DEFINITION OF MODULE

Several module definitions based on different criteria have been proposed (Wasserman and Faust, 1994; Newman, 2004; Radicchi *et al*., 2004). Generally, a module in a network is a sub-network that has more internal edges than external edges. Figure 1A show a sample network containing three modules that have very different topologies, and yet each appears to be a module, intuitively. A module definition must be able to capture these topological distinctions.

A protein interaction network can be described by a graph G = (V, E), where the set V of vertices represents proteins and the set E of edges represents interaction between proteins. In the context of this paper, the graph is synonymous with the network. In graph theory, the degree of a vertex, namely number of edges connected to it, has been commonly used to quantify the connectivity of the vertex. Radicchi *et al*. (2004) modified the degree definition of vertices in an undirected graph and proposed two module definitions: strong modules and weak modules. They defined the indegree of a vertex in an undirected graph as the number of edges connecting it to other vertices in the same module and the outdegree of a vertex in an undirected graph as the number of edges connecting it to the vertices that do not belong to the same module. For a strong module, each vertex in the module has higher indegree than outdegree. For a weak module, the sum of indegree value of all vertices in the sub-graph is greater than the sum of outdegree values of all vertices. However, these definitions limit the types of module topologies. For example, the determination of the strong module can be strongly influenced by the degree of a single vertex. As shown in Figure 1A, all three sub-graphs are not strong modules because each sub-graph has at least one vertex that does not have more indegree edges than outdegree edges. Only one strong module, Module 2 in Figure 1B, can be identified by removing the highly connected peripheral node from the Module 1 in Figure 1A. On the other hand, in the weak module definition, the edges inside a sub-graph have been counted multiple times. Even if a sub-graph has the same number of external edges as internal edges, or even more external edges than internal edges, it may still be considered to be a weak module due to this duplication in counting (e.g. Module 3 in Fig. 1B).

To overcome these limitations, we have extended the concept of degree from the individual vertex to the sub-graph in order to characterize the connectivity of a sub-graph within a graph:

DEFINITION 1. *Given a graph G, let U be a sub-graph of G ($U \subset G$). The number of edges within U is defined as the* ***indegree*** *of U, ind(U). The number of edges that connect U to the remaining part of G (G-U) is defined as the* ***outdegree*** *of U, outd(U).*

Definition 2. *The* ***modularity*** *M of a sub-graph U in a given graph G is defined as the ratio of its indegree, ind(U), and outdegree, outd(U)*:

$$M_U = ind(U)/outd(U)$$

This modularity definition will let us easily define whether a sub-graph is a module:

Definition 3. *Given a graph G, a sub-graph* $U \subset G$ *is a* ***module*** *if* $MB_U > 1$.

A more general version of the module definition could be $M_U > S$, $S \geq 1$. However, it is difficult to determine a general best value of S, which may vary according to different networks. Here, we just choose 1 as it is the smallest value that satisfies the general understanding of modules in a network. To further distinguish different levels of modules in the network, we define the complex module and the simple module as following:

Definition 4. *A sub-graph module is a* ***complex module*** *if it can be separated into at least two modules by removing edges inside it using the G-N algorithm. Otherwise, it is a* ***simple module***.

Although this module definition is simple, it directly captures the general understanding of the module concept. Similar to the approach of Radicchi *et al*. (2004), this definition is based on the relationship between insider links and outsider links, rather than relying on insider links only (Spirin and Mirny, 2003). This module definition is stricter than the weak definition of Radicchi *et al*. A sub-graph being module by this definition is also a weak module. However, there is no relationship between this module definition and the strong definition of Radicchi *et al*. Figure 1C shows an example of a sub-graph that would be a strong module, but is not a module based on this new definition. Furthermore, this new definition, based on the connectivity of sub-graphs, makes it possible to define the adjacency relationship between modules:

Definition 5. *Given two modules* $U, V \subset G$, *U and V are adjacent if* $U \cap V = \emptyset$ *and there are edges in G directly connecting vertices in U and V*.

3 ALGORITHM

In this section, a new agglomerative algorithm is presented to identify simple modules within a protein interaction network, which has been implemented in a JAVA application, MoNet. The theoretical foundations for this algorithm are as follows.

Theorem 1. *Given two modules* $U, V \subset G$, *If U, V are adjacent, the sub-graph* $W = U \cup V$ *is also a module*.

The proof of Theorem 1 is straightforward. And from the definition of complex module and simple module:

Corollary 1. *The separation of a simple module into two sub-graphs using the G-N algorithm can at most generate one module*.

Theorem 1 suggests that merging two adjacent modules will generate a complex module. Corollary 1 implies that a simple module can be created by either merging two non-modules or by merging a module and a non-module.

Similar to conventional agglomerative algorithms, our agglomerative algorithm initially puts each vertex into a singleton sub-graph. All of these singleton sub-graphs have no internal edge and at least one external edge. Then, the sub-graphs are gradually merged to find the simple modules in the network. There are two important characteristics of our agglomerative algorithm—the occurrence of merging and the order of merging.

Based on the above theorems, only two kinds of mergence are allowed: the mergence between two non-modules; and the mergence between a non-module and a module. As a key goal of this approach is to identify simple modules within the network, the mergence between two modules is prevented. Note that the network itself as a whole is a module by Definition 4. If the network is a complex module, the agglomerative algorithm should generate all simple modules inside the network. On the other hand, if the network itself is a simple module, the agglomerative algorithm will finally recover the whole network.

As mentioned above, the order of edge deletion based on the betweenness value in the G-N algorithm reflects the relative relationship between edges inside modules and edges between modules in the network. The later the edge is deleted (i.e. the lower the betweenness value), the more likely it is an edge inside a module. In MoNet, the edge deletion order generated by the G-N algorithm is reversed and used as the merging order in the agglomerative algorithm. By gradually adding edges to the sub-graphs in the reverse order of deletion by the G-N algorithm, MoNet assembles the singleton sub-graphs into simple modules. This merging scheme distinguishes different levels of modules, and generates simple modules as large as possible without merging with more loosely related sub-graphs.

The agglomerative algorithm implemented in MoNet is summarized as follows:

(1) The G-N algorithm is run on the network and the order of edge deletion is obtained.
   - (a) The betweenness scores for each edge in the network are calculated.
   - (b) The edge with the highest betweenness is identified and removed from the network.
   - (c) Step 1 is repeated until no edges remain in the network.

(2) An edge list is created in the reverse order of edge deletion in Step 1.

(3) The agglomerative algorithm is initialized by setting each vertex as a singleton sub-graph with no edges. All singleton sub-graphs are labeled as mergable.

(4) An edge is removed from the top of the edge list.

(5) If the edge connects vertices in the same sub-graph, it is added to the sub-graph.

(6) If the edge connects vertices in two different sub-graphs:
   - (a) If both sub-graphs are mergable, the two sub-graphs are evaluated based on the module definition.
     - (i) The edge is retained if merging occurs:
       - between two non-modules, or
       - between a non-module and a module
     - (ii) Otherwise, the two sub-graphs are set as non-mergable.
   - (b) If one of the sub-graphs is non-mergable, the other sub-graph is set as non-mergable.

(7) Repeat steps 4–7 until no edges are left in the edge list.

The computational complexity of the MoNet algorithm is $M^2N + M \approx O(M^2N)$, where M^2N is the time complexity for the calculation of edge betweenness; M is the number of edges and N is the number of vertices.

4 EXPERIMENTS AND RESULTS

4.1 Identification of modules from yeast protein interaction network

The yeast core protein interaction network downloaded from the DIP database (version ScereCR20041003) (Xenarios *et al*., 2002) was generated by filtering the large high-throughput protein interaction data using two different computational methods—the Expression Profile Reliability Index and the Paralogous Verification Method—to improve reliability of the interaction data (Deane *et al*., 2002). After removal of all self-connecting links, this final core protein interaction network included 2609 yeast proteins and 6355 interactions, and consists of a single large component network of 2440 interconnected proteins (6241 links) and 65 small components with sizes no more than 7 interconnected proteins. We applied MoNet to analyze the large component of the yeast core protein interaction network.

MoNet identified 86 simple modules with size larger than three from the large component of the yeast core protein interaction network (see Supplementary Table 1). The largest module has 201 proteins. All 86 simple modules together include 1651 of the 2440 proteins in the large component. Supplementary Figure 1. shows all 86 MoNet modules. The topologies of modules are diverse, including linear, star, highly connected and others.

4.1.1 Evaluation of MoNet modules using Gene Ontology To gain insights on the shared underlying biological processes of the modules, we utilize Gene Ontology annotations. We first captured the annotation of each protein in the network using Gene Ontology downloaded from the *Saccharomyces* Genome Database (SGD) (Cherry *et al*., 1998). Manual inspection of the annotations showed that most modules appeared to be enriched for proteins related to similar biological processes (Supplementary Table 1). For example, all 14 proteins in module 34 are related to vacuolar acidification, including 10 components of the hydrogen-translocating V-type ATPase complex; all 12 proteins of module 40 belong to the anaphase-promoting complex.

To further substantiate the biological significance of MoNet modules, we quantified GO biological process term co-occurrences using the SGD GO Term Finder (Hong *et al*., http://www.yeastgenome.org/). The results show that most modules (only one exception) demonstrate statistical over-representation of GO terms beyond what would be expected by chance (see Supplementary Table 2). The GO Term Finder calculates a *P*-value that reflects the probability of observing the co-occurrence of proteins with a given GO annotation in a certain module by chance based on a binomial distribution. The lower the *P*-value of a GO term, the more statistically significant a module is enriched in the GO term. The lowest *P*-values of GO term of the 86 modules range from 2.81E-2 to 5.87E-69 with an average 2.978E-17. In this multiple comparisons test for each module, the *P*-value cutoff, namely the alpha level, is chosen by dividing 0.05 (5% chance of committing a Type I error) by the number of hypotheses that were tested in the module. As a result, only Module #77 does not have a significant GO term co-clustering. The frequencies of GO terms, which are defined as the percentage of proteins in the module annotated with indicated GO term out of the total number of proteins in that module, range from 5.7 to 100% in 86 modules with an average 63.07%. The modules with lower GO term frequency usually have a star topology, which appear to represent interconnections among related biological processes.

4.1.2 Robustness of MoNet modules In comparison with shortest path approaches, ‘tie in proximity’ in betweenness values calculated using G-N algorithm are relative rare. For the yeast core protein interaction network, there are 2978020 distances based on all shortest paths that range from 1 to 13. Clearly there are a lot of identical distances between vertices. Furthermore, two connected proteins will have the same distance, 1, no matter if they are in the same module or different modules. Distances based on shortest paths thus cannot distinguish these differences, which strongly affects the results of clustering algorithms based simply on the distance scores. On the other hand, the betweenness scores of 6241 edges of yeast core protein interaction network range from 2 to 82869.09, which implies that identical edge betweenness scores are much more limited. In fact, the largest number of identical betweenness values is 669. Because MoNet picks an edge from the edges with the same betweeness values at random, different runs may yield different order of edge deletion lists and thus result in different modules. However, it is important to point out that edges inside a module will have higher chance to tie in betweenness values than edges between modules. A robust module definition should generate modules that are rarely affected by the tie in betweenness problem following different runs.

To determine the effect of ties, we have run MoNet on the largest component of yeast core protein interaction network 10 times. For each pair of proteins in the modules, the fraction of times that they are assembled in the same modules can be used to evaluate the robustness of the results to the effects of ties in betweenness. In Supplementary Figure 2, the fraction of times that proteins are grouped in the same modules for the all of the, 10 largest and 50 smallest modules obtained by 10 runs is plotted. As shown, the MoNet modules are consistent and robust; most of these reproducibility values in MoNet modules are equal to 1 (purple boxes in Supplementary Figure 2). 97.65% proteins belong to the same modules in each of the 10 runs. The coefficients of variation (standard deviation over mean) of the number of nodes, the number of edges and the number of modules (size larger than 3) are only 0.0036, 0.0017 and 0.0056, respectively. These results suggest that the module definition used in the MoNet algorithm is robust to ties in betweenness for the identification of functional modules in this biological network.

4.2 Comparison with Radicchi’s module definitions

To test the effectiveness of this new module definition in comparison with other approaches, we applied the same agglomerative algorithm using the weak and strong module definitions proposed by Radicchi *et al*. (2004) to the same large component of yeast core protein interaction network. A total of 30 strong modules with size larger than 3 were obtained. The largest strong module has 22 proteins. In total, the 30 strong modules include only 252 of the

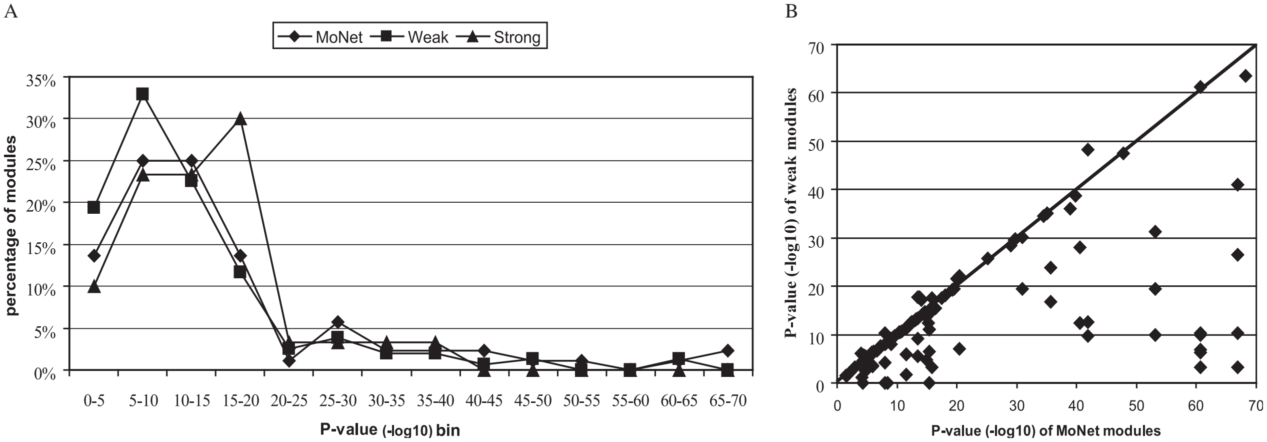

Fig. 2. A comparison of GO term clustering in modules using different definitions. (**A**) A comparison of the distribution of lowest *P*-value GO terms of modules with size larger than 3 obtained using different module definitions, grouped in different *P*-value bins. (**B**) *P*-values of modules obtained based on the definition implemented in MoNet plotted against the *P*-values of the corresponding weak modules composed of similar protein sets. The diagonal line is a plot of y = x.

2440 large component proteins. At the same time, 155 weak modules with size larger than 3 were identified. The largest weak module has 75 proteins. There are a total of 1986 proteins in all 155 weak modules. Thirty-eight of the 155 weak modules are identical to simple MoNet modules; 75 weak modules have corresponding MoNet simple modules (see detail in next paragraph); and 42 modules are obtained using the weak definition only. Similarly, we used the SGD GO Term Finder to quantify GO term over-representation testing strong and weak modules. All 30 strong modules showed significant co-occurrence of specific GO terms. However, the average of the lowest *P*-values of GO terms for the 30 strong modules is 3.37E-15, which is higher than the average of the lowest *P*-values for the 86 simple modules obtained by MoNet (2.98E-17).

Three weak modules did not show significant GO terms co-occurrence. The average of the lowest *P*-values of GO terms for the 155 weak modules is 3.82E-13, and is again higher than of the average of the 86 MoNet simple modules. The average of the lowest *P*-values of 113 weak modules that have corresponding or identical MoNet simple modules is 1.02E-14, while the average of 42 modules obtained using the weak definition only is 6.62E-9. The frequency of the lowest *P*-values of GO term of the 30 strong modules ranges from 29% to 100%, with an average of 82%. The frequency of GO term with lowest *P*-values of 155 weak modules ranges from 7% to 100%, with an average of 62.72%. The average frequency of MoNet modules is lower than strong modules, and slightly higher than weak modules.

Figure 2A shows the distribution of the lowest *P*-values of GO terms for modules with sizes larger than 3 obtained by the different module definition methods, grouped into different *P*-values bins. There are fewer high *P*-value modules and more low *P*-value modules obtained by MoNet than in the weak module and strong module groups. Supplementary Table 3 lists corresponding relationship between 113 weak modules and 86 MoNet modules. There are two types of cases in which a MoNet module is defined as corresponding to a weak module. In the first type, one weak module may include one MoNet module. Because the weak module definition is looser than the MoNet module definition, a sub-graph that is a MoNet module can be merged with more non-module sub-graphs and still be a weak module. For example, 35 of 45 proteins in MoNet Module #8 belong to 'rRNA processing' (lowest *P*-value GO term) with a *P*-value of 1.87E-48. On the other hand, the corresponding weak module 3 contains 57 proteins, whose lowest *P*-value GO term is still 'rRNA processing' (38 out of 57 proteins), but now with a *P*-value of 3.85E-48. In the second case, one MoNet module may include several weak modules. Because sub-graphs that are weak modules are not necessarily MoNet modules, the agglomerative algorithm will continue the merging process until the formation of a MoNet simple module. For example, 14 of 16 proteins in MoNet Module #28 belong to GO term 'protein biosynthesis' with a *P*-value of 3.3E-12. This MoNet module corresponds to two weak modules #75 and #79. Each of these two weak modules has 8 proteins, 7 of them belonging to 'protein biosynthesis', with *P*-values of 1.33E-6 for each. For the lowest *P*-values of GO term for each weak module, Supplementary Table 3 lists the *P*-value of that GO term in the corresponding MoNet modules. There are only two MoNet modules that do not have corresponding weak modules. Figure 2B plots the lowest *P*-values of each weak module against the lowest *P*-values of the corresponding MoNet modules. In this plot, most of points lie below the y = x line, indicating that most of the MoNet modules have lower *P*-values than the corresponding weak modules. This comparison suggests that using the over-representation of GO terms as a measurement for biological implications, our module definition outperforms the weak or strong module definition of Radicchi *et al.* (2004).

Because the coverage of proteins identified based on our module definition is less than those based on weak module definition, it is important to evaluate whether this loss is compensated by other gains. Unless we have a complete knowledge of each module, it is, however, difficult to establish a gold standard for determination the gain. We proposed to evaluate the significant of modules based on their confidence level. The loss of coverage by MoNet may not be important if only the less confidence modules are lost. We applied the classification scores developed d by of Qi *et al.* (2006) as an independent criterion to evaluate significance of the MoNet modules and weak modules. Qi *et al.* (2006) used the support vector machine to classification all possible Yeast protein–protein interaction (PPI) based on 162 features. A PPI with higher classification score will have higher likelihood of being a true positive interaction.

After establishing a score threshold, Qi *et al*. provided scores for top 26205 PPIs. In our case, the score of a PPI was set to 0 if its score is not provided by Qi *et al*. We then used the average of classification scores of all PPIs inside a module as the confidence score of the module. A module with higher score will have higher likelihood of being a true biological module. The average confidence scores of MoNet modules, weak modules with and without corresponding MoNet modules are 0.249345, 0.241069 and 0.173791, respectively. Two tails T test of the confidence scores of MoNet modules vs. those of weak modules with corresponding MoNet modules was 0.74353. In contrast, two tails T test of confidence scores of MoNet modules versus those of weak modules without corresponding MoNet modules was 0.010714; two tails T test of confidence scores of weak modules with corresponding MoNet modules versus those of weak modules without corresponding MoNet modules was 0.019869. These results suggested that there is a significant difference between the confidence level of MoNet modules and those of the weak modules without corresponding MoNet modules. The sub-graphs that do not follow our module definition are actually less significant than those following our module definition. These results implied that our module definition can identify modules with higher confident level; and the loss of proteins (1651 versus 1986) in MoNet modules is appropriate.

To further compare the coverage of proteins in the same biological process, we examed the coverage of two biological process GO terms—‘mRNA metabolism’ and ‘Golgi vesicle transport’, which are relatively far away from each other in the GO tree (Supplementary Figure 5), in the MoNet modules and weak modules. Supplementary Tables 4 and 5 list both MoNet and weak modules that are significant in these two GO terms. For ‘mRNA metabolism’, MoNet modules cover 105 proteins comparing to 101 proteins covered by weak modules. There are no weak modules without corresponding MoNet module. For ‘Golgi vesicle transport’, MoNet modules cover only 71 proteins compared to 81 proteins covered by weak modules. This is because that there are two weak modules (totally 13 out of 21 proteins) that do not have corresponding MoNet modules. In summary, the coverage of proteins by Monet and Radicchi’s modules for these two biological processes is very similar. As there are 42 weak modules that do not have corresponding MoNet modules, it is not surprising that the coverage of protein in weak modules will be higher. However, the trade off in increase coverage is offset by the increase in false positive cluster membership as judged by the increase in co-clustering P-values.

To test the robustness of weak and strong modules, we assembled the strong and weak modules using the edge deletion list generated by 10 MoNet runs used before. Supplementary Figure 3 plots the fractions of times proteins are grouped in the same modules in all 10 largest and 50 smallest weak modules in 10 runs. Supplementary Figure 4 plots the fractions of times that proteins are grouped in the same modules in the 10 largest strong modules in 10 runs, respectively. As shown, the weak and strong modules that are generated following multiple runs are less consistent than those generated with the MoNet modules definition. Only 91.92% proteins belong to the same weak modules and 74.44% proteins belong to the same strong modules in 10 runs. Furthermore, the fractions of proteins in many modules are not close to 1. In comparison with result of MoNet modules in Supplementary Figure 2, the weak and strong modules are more dramatically affected by the tie in betweenness value and are less consistent between multiple runs. The coefficients of variation of the number of nodes, the number of edges and the number of modules (size larger than 3) in weak modules are 4.28, 7.12, 1.96 times greater than those of MoNet modules. The coefficients of variation of the number of nodes, the number of edges and the number of modules (size larger than 3) in strong modules are 11.03, 22, 5.07 times greater than those of MoNet modules. This observation thus suggests that weak and strong module definitions are not as robust to the tie in betweenness issue as the MoNet module definition when using the G-N algorithm to separate the network.

4.3 Interconnection between modules

In order to gain insight into how the modules obtained by MoNet relate to each other within the cellular system, we assembled an interconnection network of the 86 MoNet modules from the large component of the yeast core protein interaction network. The network of modules was constructed as follows: for each adjacent module pair, the edge that is deleted last by the G-N algorithm was selected from all the edges that connect two modules to represent the link between two modules. A total of 82 of 86 modules are connected to each other. The network of modules obtained is highly connected, which suggests that a yeast cell is a complex web of highly interconnected functional modules. To facilitate discussion, we show the 30 modules with lowest P-values in Figure 3. The width and grayscale of edges reflects the order of the deletion in the G-N algorithm, which also represent relative relationships among modules (Fig. 3). The wider and darker the line is, the later the edge it represents is deleted by the G-N algorithm, which implies closer relationships between the modules that are linked together. Among these links, some of them are known to be functionally close. For example, two ribosome biogenesis modules (#4, #8) connect to an mRNA catabolism module (#35) with very heavy and wide links. Other links connect modules that are functionally related. For example, the nucleocytoplasmic transport module (Module #1) serves as a hub connecting the protein biogenesis modules, RNA metabolism and vesicle-mediated transport modules. Finally, some of these connections are novel and currently unknown from the available experimental data. In summary, although these relative relationships between modules are based on the structural information in the network (edge betweenness scores), many of them appear to reflect true biological interconnected functionality.

Furthermore, to gain insights into the relationship among modules in the same biological process, we sampled two large GO categories, ‘mRNA metabolism’ and ‘Golgi vesicle transport’, and assembled a network for each of these two GO categories. As shown in Supplementary Figures 6, 5 of 8 modules in ‘Golgi vesicle transport’ are highly significant in over-representation of the GO term and are also highly connected together. Similarly, 4 of 5 modules that are highly significant in over-representing ‘mRNA metabolism’ are also highly connected together.

5 CONCLUSION AND DISCUSSION

Identifying separable modules within biological systems is essential for the understanding of the high-level organization of the cell. In this study, we extended the concept of degree from the vertex to the sub-graph and proposed a new formal definition of module within a network based on the degree definition of the sub-graph. A new agglomerative algorithm was designed to assemble simple modules

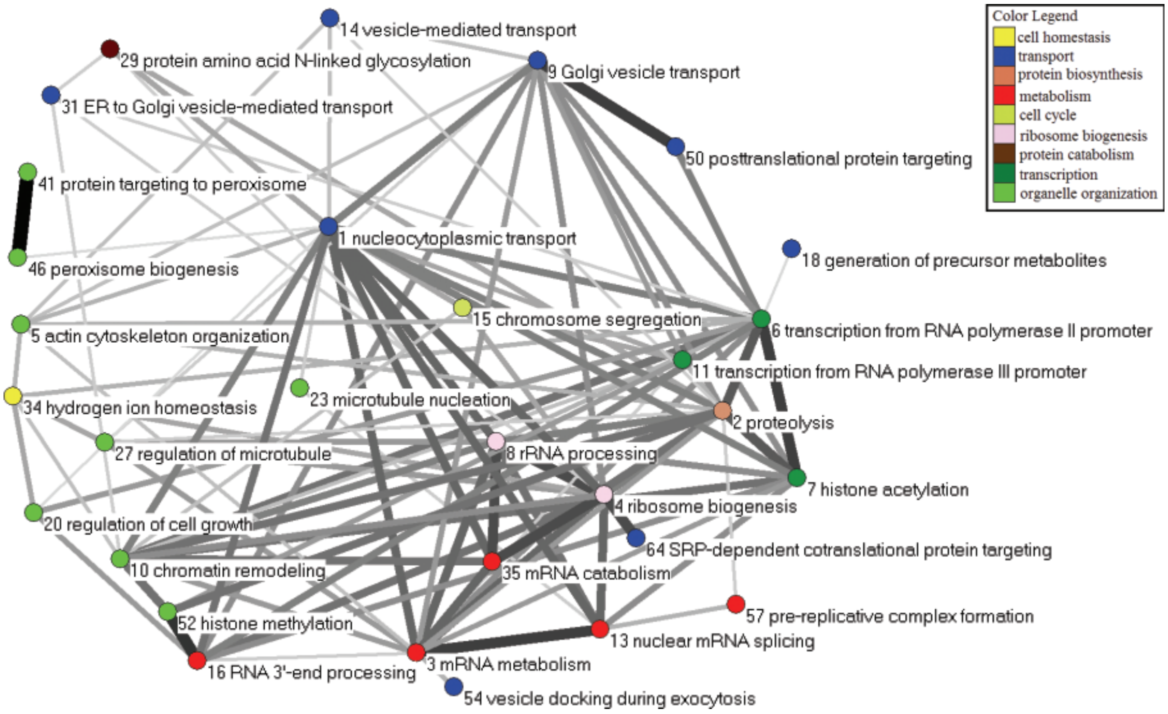
Color Legend
cell homestasis
transport
protein biosynthesis
metabolism
cell cycle
ribosome biogenesis
protein catabolism
transcription
organelle organization
14 vesicle-mediated transport
29 protein amino acid N-linked glycosylation
9 Golgi vesicle transport
31 ER to Golgi vesicle-mediated transport
50 posttranslational protein targeting
41 protein targeting to peroxisome
1 nucleocytoplasmic transport
18 generation of precursor metabolites
46 peroxisome biogenesis
15 chromosome segregation
6 transcription from RNA polymerase II promoter
5 actin cytoskeleton organization
11 transcription from RNA polymerase III promoter
34 hydrogen ion homeostasis
23 microtubule nucleation
2 proteolysis
27 regulation of microtubule
8 rRNA processing
7 histone acetylation
4 ribosome biogenesis
20 regulation of cell growth
64 SRP-dependent cotranslational protein targeting
10 chromatin remodeling
35 mRNA catabolism
57 pre-replicative complex formation
52 histone methylation
13 nuclear mRNA splicing
16 RNA 3'-end processing
3 mRNA metabolism
54 vesicle docking during exocytosis

Fig. 3. Interaction network of modules. A section of the interconnected module network with 30 functional modules that have the lowest *P*-values of most significant GO term in each module. Each node denotes a module. Each edge denotes a connection between two modules. Width and gray scale of edges reflects the order of deletion in the G-N algorithm. The wider and darker the line is, the later the edge it represents is deleted by the G-N algorithm, which implies closer relationships between the modules that are linked together. Modules (nodes) are assigned to biological process categories by summarizing the GO annotation of their constituent proteins. The graph was produced using Pajeck (Batagelj and Mrvar, http://vlado.fmf.uni-lj.si/pub/networks/pajek/).

from protein interaction networks, using the relative order of edges based on the betweenness values generated by the G-N algorithm as the merging order. Our new approach is based solely on the topological characteristics of the network, without transformation of the network into a weighted graph.

Although the choice of $S = 1$ in our module definition is simple, it captures the general understanding of a module and of module topologies. Moderately increasing S will lead to the identification of tighter modules with lower average *P*-values. However, the modules identified with higher S have similar confidence levels. For example, two tails T test of the confident scores of original MoNet modules versus those of modules identified by MoNet based upon module definition with $S = 2$ is 0.874908. On the other hand, larger value of S may lead to the merging of large modules in the agglomerative algorithm. In the extreme case, the agglomerative algorithm will ultimately identify the whole network as one module, which has all inside links and no outside links. Supplementary Figure 7 shows how the number of proteins in the largest module changes with different S values. The number has a jump when $S \geq 1$; keeps stable in the range $1 \leq S < 2$ and increase dramatically after $S > 2$.

Application of MoNet to the large component of DIP yeast core protein interaction network generated 86 modules significantly enriched for functional Gene Ontology terms. Comparison with the weak and strong module definitions of Radicchi *et al*. (2004) showed that the *P*-values of MoNet modules obtained are in general lower, while maintaining similar frequencies of proteins. Tests showed that the MoNet modules have significantly higher confidence levels than sub-graphs that do not follow our module definition. Furthermore, the membership of MoNet modules are shown to be more robust than the weak and strong modules following multiple runs and are only slightly affected by the tie in betweenness value.

Another advantage of the MoNet approach is that it facilitates the description of the network of modules, allowing for the construction of a network of adjacent modules. Furthermore, the relative order of linking edges between adjacent modules by the G-N algorithm captures the relative relationship between the modules—the later the linking edge is deleted by the G-N algorithm, the closer the adjacent modules are linked together. Evaluation of the module connections suggests that the relative interactions between MoNet modules may represent actual links between biological pathways.

Although the yeast protein interaction network used in this study has been filtered by two computation algorithms, there may still be false positive interactions in the dataset, which would be treated equally with true positive interactions by MoNet. This may be one of the reasons that some of the modules produced by MoNet contain proteins associated with different biological processes. Another limitation of MoNet is the dependency on the G-N algorithm. Computation by the G-N is resource intensive. In addition, the performance of the G-N algorithm is dependent on the completeness of the network organization to generate the order of edges. Even though the agglomerative assembly portion of MoNet identifies modules locally, the incompleteness of the protein interaction data may still affect the final membership of the resulting modules because of the global characteristics of G-N algorithm.

In conclusion, the approach for modular decomposition of protein interaction networks implemented in MoNet provides an objective approach to the understanding of the organization and interactions of biological processes. With the increasing amount of protein interaction data available and the development of high-throughput approaches for defining genetic interaction networks (e.g. Basso *et al*., 2005), MoNet may facilitate the construction of a more complete view of the composition and interconnection of functional modules and the understanding of the organization of the whole cellular system.

ACKNOWLEDGEMENTS

The authors would like to thank Shankar Subramaniam, Trey Ideker, Nick Grishin and Stuart Johnson for helpful advice and critical review of this manuscript. The authors are also thankful to the anonymous reviewers for their great suggestion. The authors are thankful to Yanjun Qi for providing the confidence level data. This research was supported by the National Institutes of Health contracts N01-AI40076 and N01-AI40041, and by The United States Department of Energy under the Genomics: GTL program through Shewanella Federation, Office of Biological and Environmental Research, Office of Science. Oak Ridge National Laboratory is managed by University of Tennessee-Battelle LLC for the Department of Energy under contract DE-AC05-00OR22725. F.L. is supported by NSF EPSCoT grant EPS-0447660.

Conflict of Interest: none declared.

REFERENCES

Arnau,V. *et al*. (2005) Iterative cluster analysis of protein interaction data. *Bioinformatics*, **21**, 364–378.

Bader,G.D. and Hogue,C.W.V. (2003) An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, **4**, 2.

Barabasi,A.-L. and Oltvai,Z.N. (2004) Network biology: understanding the cell's functional organization. *Nat. Rev*., **5**, 101–114.

Basso,K. *et al*. (2005) Reverse engineering of regulatory networks in human B cells. *Nat. Genet*., **37**, 382–390.

Batagelj,V. and Mrvar,A. Pajek—Program for large network analysis.

Bu,D. *et al*. (2003) Topological structure analysis of the protein–protein interaction network in budding yeast. *Nucleic Acids Res*., **31**, 2443–2450.

Chen,J. and Yuan,B. (2006) Detecting functional modules in the yeast protein–protein interaction network. *Bioinformatics*, **22**, 2283–2290.

Cherry,J.M. *et al*. (1998) SGD: *Saccharomyces* Genome Database. *Nucleic Acids Res*., **26**, 73–79.

Deane,C.M. *et al*. (2002) Protein interactions: two methods for assessment of the reliability of high-throughput observations. *Mol. Cell. Proteomics*, **1**, 349–356.

Enright,A.J. and Ouzounis,C.A. (2001) BioLayout—an automatic graph layoutalgorithm for similarity visualization. *Bioinformatics*, **17**, 853–854.

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

Guimera,R. and Amaral,L.A.N. (2005) Functional cartography of complex metabolic networks. *Nature*, **433**, 895–900.

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

Hong,E.L., Balakrishnan,R., Christie,K.R., Costanzo,M.C., Dwight,S.S., Engel,S.R., Fisk,D.G., Hirschman,J.E., Livstone,M.S., Nash,R., Oughtred,R., Park,J., Skrzypek,M., Starr,B., Theesfeld,C.L., Andrada,R., Binkley,G., Dong,Q., Lane,C.D., Hitz,B.C., Miyasato,S., Schroeder,M., Weng,S., Wong,E.D., Dolinski,K., Botstein,D., Cherry,J.M. *Saccharomyces* Genome Database.

MacCuish,J. *et al*. (2001) Ties in proximity and clustering compounds. *J. Chem. Inform. Comp. Sci*., **41**, 134–146.

Newman,M.E.J. (2004) Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, **69**, 066133.

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

Qi,Y.J. *et al*. (2006) Evaluation of different biological data and computational classification methods for use in protein interaction prediction. *Proteins: Struct. Func. Bioinformatics*, **63**, 490–500.

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

Ravasz,E. *et al*. (2002) Hierarchical organization of modularity in metabolic networks. *Science*, **297**, 1551–1555.

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

Snel,B. *et al*. (2002) The identification of functional modules from the genomic association of genes. *Proc. Natl Acad. Sci. USA*, **99**, 5890–5895.

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

Wasserman,S. and Faust,K. (1994) *Social Network Analysis*. Cambridge University Press, Cambridge, UK.

Wilhelm,T. *et al*. (2003) Physical and functional modularity of the protein network in Yeast. *Mol. Cell. Proteomics*, **2**, 292–298.

Xiong,H. *et al*. (2005) Identification of functional modules in protein complexes via hyperclique pattern discovery. *Pac. Symp. Biocomp*., **10**, 221–232.

Xenarios,I. *et al*. (2002) DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res*., **30**, 303–305.