A Graph-Theoretic Method for Mining Overlapping Functional Modules in Protein Interaction Networks*

Min Li[1], Jianxin Wang[1,**], and Jianer Chen[1,2]

[1] School of Information Science and Engineering,
Central South University, Changsha 410083, P.R. China
[2] Department of Computer Science,
Texas A&M University, College Station, TX 77843, USA
limin@mail.csu.edu.cn,jxwang@mail.csu.edu.cn,chen@cs.tamu.edu
http://netlab.csu.edu.cn

Abstract. Identification of functional modules in large protein interaction networks is crucial to understand principles of cellular organization, processes and functions. As a protein can perform different functions, functional modules overlap with each other. In this paper, we presented a new algorithm OMFinder for mining overlapping functional modules in protein interaction networks by using graph split and reduction. We applied algorithm OMFinder to the core protein interaction network of budding yeast collected from DIP database. The experimental results showed that algorithm OMFinder detected many significant overlapping functional modules with various topologies. The significances of identified modules were evaluated by using functional categories from MIPS database. Most importantly, our algorithm had very low discard rate compared to other approaches of detecting overlapping modules.

Keywords: protein interaction network, functional module, graph.

1 Introduction

Proteins are central components of cell machinery and life [1]. Large-scale interaction detection methods have resulted in a large amount of protein-protein interaction data. Such data can be naturally represented in the form of networks. System level analysis and understanding of protein interaction networks is one of the most fundamental challenges in post-genomic era. Accumulating evidence suggests these protein interaction networks are organized by functional modules, which are cellular entities performing certain biological functions [2,3,4,5,6].

* This research was supported in part by the National Natural Science Foundation of China under Grant Nos. 60433020 and 60773111, the Program for New Century Excellent Talents in University No. NCET-05-0683, the Program for Changjiang Scholars and Innovative Research Team in University No. IRT0661.

** The corresponding author.

I. Măndoiu, R. Sunderraman, and A. Zelikovsky (Eds.): ISBRA 2008, LNBI 4983, pp. 208–219, 2008.

Identification of functional modules is crucial in understanding the principles of cellular organization and unveiling functional and evolutionary mechanisms.

A wide range of graph clustering algorithms have been developed to identify functional modules from protein interaction networks. All these methods can be categorized into three groups: partitional clustering, hierarchical clustering and density-based clustering.

Partitional clustering approaches partition a network into multi separated sub-networks. As a typical example, the Restricted Neighborhood Search Clustering (RNSC) algorithm [7] explores the best partition of a network using a cost function. It starts with randomly partitioning a network, and iteratively moves a node from one cluster to another to decrease the total cost of clusters. It can get the best partition by running multi-times. However, it needs the number of clusters as prior knowledge and its results depend heavily on the quality of initial clustering.

Hierarchical clustering approaches have been applied widely for identifying functional modules [6,8,9,10,11]. Hartuv and Shamir use minimum cut set to divide network recursively [8]. Girvan and Newman decompose a network based on the graph theoretical concept of betweenness centrality [9]. Luo and colleagues also use betweenness and develope an agglomerate algorithm named MoNet [6]. Several approaches have been proposed for weighting protein-protein interactions. Pereira-Leal and colleagues propose an approximate solution to weight a protein interaction based on the number of experiments that support the interaction [10]. Another method is to weight the distance between two proteins by the length of the shortest path between them [11]. However, the method usually generates many identical distances and leads to a "tie in proximity" problem during hierarchical clustering [6].

As a disadvantage, partitional clustering approaches and hierarchical clustering approaches can only generate separated functional modules. In fact, functional modules overlap with each other, since a protein can be included in several different functional modules to perform different functions [12,13].

Density-based clustering approaches focus on detecting highly connected subnetworks. An extreme example is to identify all fully connected subgraphs [14]. Mining fully connected subgraphs only is too strict to be used in real biological networks. A variety of alternative methods have been proposed to detect dense subgraphs by using a density threshold [15,16,17]. Recently, several density-based clustering approaches have attempted to detect overlapping functional modules [12,18]. However, such methods of detecting highly connected subnetworks neglect many peripheral proteins that connect to the core protein clusters with few links, even though these peripheral proteins may represent true interactions. In addition, biologically meaningful functional modules that do not have highly connected topologies are ignored by these approaches [6].

To mine overlapping functional modules with various topologies, we present a new graph-theoretic-based algorithm, named OMFinder. Recent results of analyzing biological networks show that highly connected proteins in the networks play an important role in evolution and likely participate in multiple biological

progresses [19,20,21,22,23]. Based on this fact, we divide the proteins into two classes of high-degree and low-degree nodes and constrain only the high-degree nodes can belong to multiple functional modules. We split the original graph G into three subgraph G_h, G_l and G_b, where G_h is a subgraph representing the relations between high-degree nodes, and G_l is a subgraph representing the relations between low-degree nodes, and G_b is a subgraph representing the relations between high-degree nodes and low-degree nodes. Each operation is only in one separated subgraph, which improves the efficiency of the algorithm effectively. We apply algorithm OMFinder to the core protein interaction network of budding yeast in DIP database. The experiment results show that our algorithm OMFinder can detect many significant functional modules effectively. Most of the identified modules overlap with each other.

2 Methods

The protein interaction network can be represented as an undirected, un-weighted graph $G(V, E)$ with proteins as a set of nodes V and interactions as a set of edges E. As the protein interaction networks are scale-free, and they are dominated only by a few nodes known as hubs. We proposed a new graph-theoretic-based algorithm for detecting overlapping functional modules in protein interaction networks. Distinguishing from other methods, we defined the graph G as a superposition of three subgraph G_h, G_l and G_b. According to the three subgraphs, we constructed a reduced graph for G. And, we constrained that only the informative nodes could belong to more than one functional modules. Based on the graph split and reduction, we developed a new algorithm, named OMFinder.

2.1 Informative Proteins Selection

Recently, the small world effect and scale-free property of protein interaction networks have been investigated extensively [19,20,21,22]. The small world is characterized by small average length of the shortest paths and large clustering coefficient. The scale-free networks follow a power law degree distribution, the probability of a node in which has a degree k is approximated by $P(k) \approx \alpha k^{-r}$ with $1 < r < 3$. The scale-free of protein interaction networks shows that only a few nodes (known as hubs) have very large degrees, while most other nodes have very few interactions. Genome-wide studies show that deletion of a hub protein is more likely to be lethal than deletion of a non-hub protein [20,21,23]. Thus, we select the nodes with large degrees as informative proteins from the protein interaction networks.

2.2 Graph Split and Reduction

The nodes in the protein interaction networks can be divided into two classes, namely informative and non-informative proteins. More precisely, we define a graph G with node set $V(G)$ that is composed of two disjoint subsets $V_h \subset V(G)$

and $V_l \subset V(G)$, where V_h is the set of high-degree nodes, and V_l is the set of low-degree nodes, and $|V_h| + |V_l| = |V(G)|$. Then the edge $e(u, v)$ in the graph G can be grouped into three classes: $e_h(u, v \in V_h)$, $e_l(u, v \in V_l)$, and $e_b(u \in V_h, v \in V_l$ or $u \in V_l$, $v \in V_h)$. Let $E_h = \{e_h\}$, $E_l = \{e_l\}$, and $E_b = \{e_b\}$. A graph G can be viewed as a superposition of three subgraph $G_h(V_h, E_h)$, $G_l(V_l, E_l)$ and $G_b(V_b, E_b)$. A simple example is illustrated in Fig. 1. Suppose we select two nodes of the highest degrees, which are marked in black in the original graph G, as informative nodes. Then the graph G can be separated into three subgraph G_h, G_l and G_b. The subgraph G_l is divided into three separated subgraphs S_1, S_2, and S_3. This is a common phenomenon known as centrality-lethality rule in protein interaction networks. If the subgraphs of G_l are reduced as nodes, named S-nodes, then the original graph G can be rebuilt as Fig. 2(a). If four nodes are selected as informative nodes from the original graph G, then graph G can be reduced as Fig. 2(b).

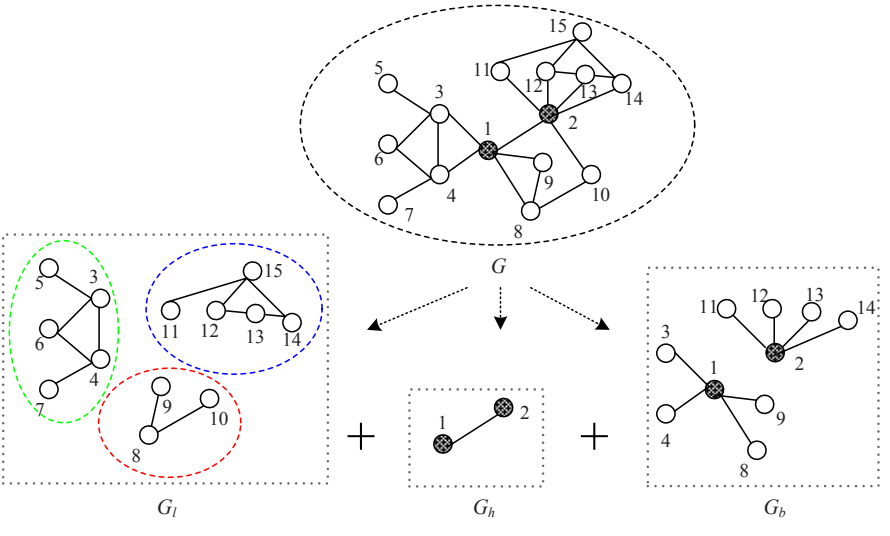

Fig. 1. An example for that a graph G is a superposition of three subgraph G_h, G_l and G_b. Node 1 and node 2 are two informative nodes of graph G, whose degrees are largest. Graph G_l is divided into three separated sub-graphs S_1, S_2, and S_3.

In Fig.2, the solid edge (e_h) connects two high-degree nodes and the dashed edge connects a high-degree node and a S-node. We construct a dashed edge between a high-degree node and a S-node, if there is one interaction between the low-degree nodes (in subgraph S) and the high-degree node in G_b. To measure how strongly the S-nodes connect to the informative nodes, we define the weight of the dashed edge as:

$$w_{hS} = \frac{|E_{hS}|}{|V_S|} \tag{1}$$

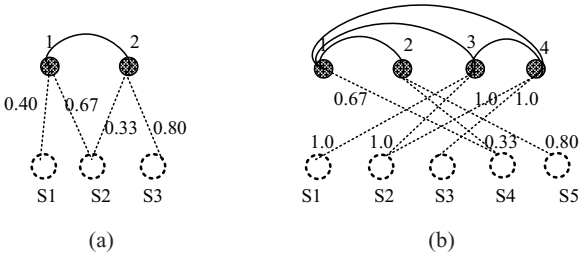

Fig. 2. The reduced graph of G. The arc edge connects two high-degree nodes, and the dashed edge connects a high-degree node and a S-node. (a) node 1 and node 2 are informative nodes, S_1={3,4,5,6,7}, S_2={8,9,10},S_3={11,12,13,14,15}; (b) node 1, node 2, node 3 and node 4 are informative nodes, S_1={5}, S_2={6},S_3={7}, S_4={8,9,10}, S_5={11,12,13,14,15}.

Here, $|E_{hS}|$ is the total number of edges between the high-degree node and the low-degree nodes in subgraph S, and $|V_S|$ is the number of nodes in subgraph S.

In biological networks, the high-degree nodes act as hubs and are essential to the networks. Jeong and colleagues analyzed the topologies and functions of 43 metabolic networks of different organisms. They found that all the 43 metabolic networks were scale-free and were dominated by the same highly connected substrates, while less connected substrates preferentially served as the educts or products of species-specific enzymatic activities [19]. Most of the substrates were only concerned with one or two metabolic reactions, only a few of substrates were concerned with multiple metabolic reactions. For protein interaction network, it is also a scale-free network and its highly connected nodes have the same property. We obtained protein lethality data from the MIPS database [24]. For the essential proteins annotated in FunCat [25], more than 80% have two or more annotations. Most of the highly connected nodes in protein interaction networks are essential. Thus, in the protein interaction networks, it is more likely for the highly connected proteins having multiple functions than the less connected proteins. In the reduced graph model, we constrained the S-nodes could only be separated into one module, and the high-degree nodes could be separated into multiple modules. Since several highly connected proteins may be concerned with the same biological progress together, we first group them by enumerating all the fully connected subgraphs in G_h. Then the predigested graphs in Fig. 2 can be reduced to the bipartite graphs showed in Fig.3.

We define the node reduced from the fully connected subgraph as C-node. The weight of the relation between a C-node and a S-node is defined as:

$$W_{CS} = \sum_{h \in C} w_{hS} \tag{2}$$

A S-node may be relate to several C-nodes. To constrain each S-node belongs to one functional module, we only construct an edge between the S-node and a C-node when the weight of the relation between them is maximum. Then, each S-node has only one C-node connecting to it. In contrast, a C-node may have

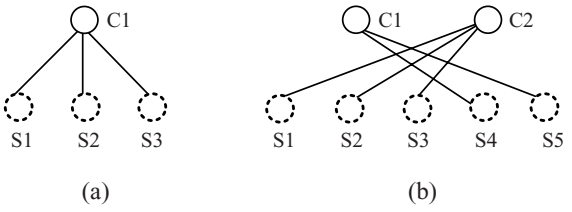

Fig. 3. The bipartite graph H reduced from Fig.2. The high-degree nodes are grouped by enumerating the fully connected subgraphs in G_h. (a) Node 1 and node 2 are informative nodes, C_1={1,2}; (b) Node 1, node 2, node 3 and node 4 are informative nodes, C_1={1,2}, C_2={1,3,4}.

several S-nodes connecting to it. Then, the separated subgraphs in the bipartite graph H are the functional modules.

3 Experiments and Results

We downloaded the budding yeast core protein interaction network (version ScereCR20070107) from DIP, the Database of Interacting Proteins [26]. We removed all the self-connecting interactions and the repeated interactions from the original network. The final core protein interaction network includes 2528 yeast proteins and 5734 interactions. We use a parameter PI (***P***ercentage of ***I***nformative proteins) to control the number of the informative nodes selected.

3.1 Identification of Overlapping Modules

We implemented OMFinder to analyze the core protein interaction network. By changing the values of parameter PI from 20% to 40%, we achieved five different output sets of modules from the protein interaction networks. As shown in Table 1, the number of identified modules with $size \geq 3$ was increasing with the increase of PI. On the contrary, the number of identified modules with $size \geq 8$ was decreasing as PI increased. The average size of the identified modules and the size of the biggest module were both decreased as PI increased. This showed the modules identified by OMFinder became more and smaller when PI increased.

Most of the identified modules shared common proteins. To evaluate their overlapping rate, we counted the number of the appearances across different modules for each protein. The average overlapping rates of identified modules with different values of PI were shown in Table 1. As PI increased, the average overlapping rate was slightly increased.

Cho, Hwang, and their colleagues showed that discarding the sparsely connected proteins could be a fatal decision which might loose the important biological information [27,28]. To evaluate how many proteins neglected by the identified modules, we define the discard rate (Dr), as shown in formula (3).

Table 1. The effect of parameter PI on clustering

Parameter	Number of the identified modules			Average	Max	Overlapping
PI	$size \geq 3$	$size \geq 5$	$size \geq 8$	size	size	rate
$PI = 20\%$	746	319	121	5.68	94	1.58
$PI = 25\%$	886	346	107	5.14	45	1.73
$PI = 30\%$	1024	345	95	4.74	39	1.84
$PI = 35\%$	1143	344	68	4.44	39	2.13
$PI = 40\%$	1263	323	55	4.22	37	2.05

$$Dr = \frac{|V| - |\cup M_i|}{|V|} \tag{3}$$

where $|V|$ was the total number of proteins in the network, $|\cup M_i|$ was the number of proteins included in all the identified modules with size larger than a given threshold. Since a module with $size = 2$ only represents one interaction with little information, a significant functional module should include at least 3 proteins. The discard rates of the identified modules generated by OMFinder using different values of parameter PI were shown in Fig.4. As shown in Fig.4, our method OMFinder had a very low discard rate, which was lower than 10% for the identified modules with size equal or larger than 3. However, CFinder and Maximal Clique both had a very high discard rate of more than 50%. If only the modules with $size \geq 5$ were considered, there were approximatively 90% proteins neglected by Maximal Clique.

3.2 Statistical Assessment and Functional Annotation

The P-value from hypergeometric distribution was often used to estimate whether a given set of proteins was accumulated by chance. It has been used as a criteria to assign each identified module a main function [7,16,22]. Here, we also calculated P-value for each identified module and assigned a function category to it when the minimum P-value occurred. The computing formula of P-value [7,16,22] was defined as:

$$P = 1 - \sum_{i=0}^{k-1} \frac{\binom{|F|}{i}\binom{|V|-|F|}{|M|-i}}{\binom{|V|}{|M|}} \tag{4}$$

where $|M|$ was the number of proteins in an identified module, $|F|$ was the number of proteins in a reference function, and k was the number of common proteins between the functional group and the identified module. Low P-value indicated that the module closely corresponded to the function because the network had a lower probability to produce the module by chance [13].

The functional classification of proteins used in this paper was collected from the MIPS Functional Catalog (FunCat) database. FunCat [25] was an annotation scheme of tree-like structure for the functional description of proteins. There

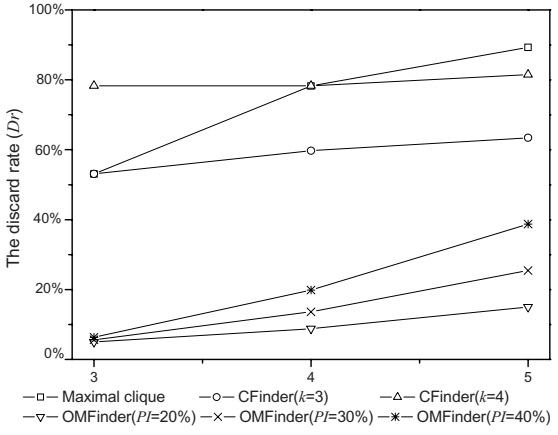

Fig. 4. The comparison of discard rates of OMFinder and other two methods: CFinder and Maximal Clique

were up to 6 levels of increasing specificity and 1360 functional categories in FunCat. We obtained 215, 219, 205, 181 and 159 modules with $size \geq 6$ when using PI=20%, 25%, 30%, 35%, and 40%. The number of the identified modules ($size \geq 6$) with $P < 0.01$ and with $P < 0.001$ generated by different values of PI was shown in Table 2.

Table 2. The number of modules ($size \geq 6$) generated by OMFinder using different values of PI with $P < 0.01$ and $P < 0.001$, respectively

Parameter	$PI = 20\%$	$PI = 25\%$	$PI = 30\%$	$PI = 35\%$	$PI = 40\%$
Number of all modules	215	219	205	181	159
Number of modules($P < 0.01$)	212	214	200	176	153
Number of modules($P < 0.001$)	189	195	178	153	113

For all the identified modules generated with different values of PI, there were more than 96.2% and 83% modules matching well with known functional categories with $P < 0.01$ and $P < 0.001$, respectively. Table 3 showed annotations for some identified modules ($size \geq 10$) with $P < 1.0 \times 10^{-10}$, where PI=25% was used.

3.3 Accuracy Analysis

Recall and precision are two important aspects to estimate the performance of algorithms for detecting functional modules. Recall is the fraction of the true-positive predictions out of all the true predictions, and precision is the fraction of the true-positive predictions out of all the positive predictions. The calculation formulae [13] of recall and precision are:

Table 3. Annotations of the identified modules ($size \geq 10$) with $P < 1.0 \times 10^{-10}$. All the identified modules are generated by using $PI = 25\%$.

ID	Size	P-value	Function	Unknown proteins
1	21	$< 1.00 \times 10^{-30}$	mitochondrial transport	YJL064W
2	20	$< 1.00 \times 10^{-30}$	rRNA processing	-
3	15	$< 1.00 \times 10^{-30}$	electron transport	YBR281C;YGR210C
4	11	1.11×10^{-16}	chromosome condensation	-
5	17	2.22×10^{-16}	rRNA synthesis	YIL141W;YJR087W
6	25	1.55×10^{-15}	general transcription activities	YLR123C;YMR102C;YHL023C
7	13	3.33×10^{-15}	microtubule cytoskeleton	-
8	25	4.44×10^{-15}	DNA repair	YJL043W; YFL042C
9	16	2.67×10^{-13}	enzymatic activity regulation /enzyme regulator	YLR190W
10	16	6.31×10^{-13}	proteasomal degradation (ubiquitin/proteasomal pathway)	-
11	13	9.06×10^{-13}	metabolism of energy reserves (e.g. glycogen, trehalose)	-
12	11	4.80×10^{-12}	cell wall	YFR044C
13	17	7.22×10^{-12}	regulation of nitrogen utilization	YIL152W;YDR078C;YLR376C; YHL006C
14	21	6.79×10^{-11}	splicing	YGR021W; YPL105C
15	15	7.28×10^{-11}	vacuole or lysosome	-
16	10	7.65×10^{-11}	perception of nutrients and nutritional adaptation	Q06966

$$recall = \frac{|M \cap F_i|}{|F_i|} \tag{5}$$

$$precision = \frac{|M \cap F_i|}{|M|} \tag{6}$$

Here, F_i is a functional category mapped to module M. The proteins in functional category F_i are considered as true predictions, the proteins in module M are considered as positive predictions, and the common proteins between F_i and M are considered as true positive predictions. It is obvious that the larger module is likely to have higher recall and lower precision. If we generate all the proteins in one module, then its recall will be equal to 1. In contrast, the module with smaller size tends to have higher precision and lower recall. As an extreme case, if we generate a single protein as one module, then we have the maximum value of precision. In general, f-measure is used as a harmonic mean of precision and recall. The f-measure [13] is defined as formula (7).

$$f - measure = \frac{2 * precision * recall}{precision + recall} \tag{7}$$

For each identified module, we calculated its f-measure to assess its accuracy. As shown in Fig.5, for the same f-measure, the number of the identified modules generated by OMFinder was all more than that generated by CFinder, the former

was about five times more than the latter. Though the number of the identified modules generated by Maximal Clique was close to those generated by OMFinder for the same f-measure, Maximal Clique discard too many proteins.

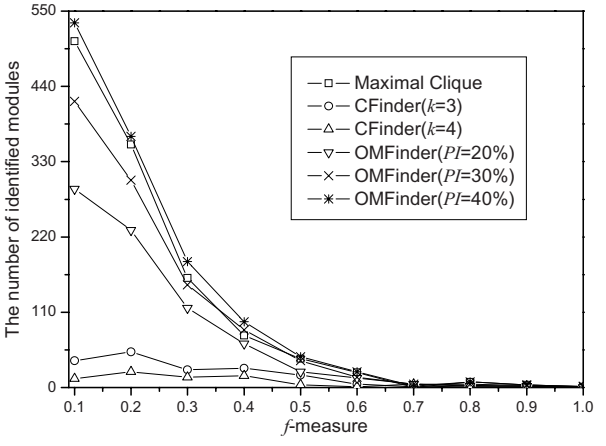

Fig. 5. The number of identified modules with respect to $f-meaure \geq 0.1, 0.2, \cdots, 1.0$

4 Conclusions

Functional modules play a special role in biological networks, which are relatively independent units performing certain biological functions. Many graph clustering methods have been developed to detecting functional modules in protein interaction networks. However, most of the previous methods can not detect the overlapping functional modules by generating separate subraphs. And, a few existed methods for identifying overlapping modules focused on detecting highly connected subgraphs, which neglected many peripheral proteins.

In this paper, we present a new graph-theoretic-based algorithm for identifying overlapping functional modules in protein interaction networks. We divide the proteins into two classes, namely high-degree and low-degree nodes, respectively. Based on the fact that highly connected proteins in biological networks play an important role in evolution and likely participate in multiple biological progresses, we constrain that only the high-degree nodes can belong to multiple functional modules. We split the original graph G into three subgraph G_h, G_l and G_b. Each operation is only in one separated subgraph, which improves the efficiency of the algorithm effectively. Our algorithm OMFinder is implemented in C++. We applied algorithm OMFinder to the core protein interaction network of budding yeast in DIP database. Many significant functional modules were detected. Of all the 219 identified modules with $size \geq 6$ ($PI = 25\%$), more than 96.2% corresponded to $P < 0.01$, and more than 86.8% corresponded to $P < 0.001$. We predicted functions for previous unknown proteins by assigning the identified modules a main function with the lowest P-value. We identified

more overlapping functional modules with high recall and precision than previous methods CFinder. Most importantly, our algorithm OMFinder can cover most of the proteins in the network, which neglect few peripheral proteins. As a new graph-theoretic method, we think that it will be helpful to detect functional modules and to analyze the topologies of biological networks.

Acknowledgments. The authors wish to thank Adamcsek B., Palla G., Farkas I., Derenyi I., and Vicsek T for sharing their program of CFinder.

References

1. Asur, S., Ucar, D., Parthasarathy, S.: An ensemble framework for clustering protein-protein interaction networks. ISMB/ECCB 23, 29–40 (2007)
2. Hartwell, L.H., et al.: From molecular to modular cell biology. Nature 402, 47–52 (1999)
3. Barabasi, A.L., Oltvai, Z.N.: Network biology: understanding the cell's functional organization. Nat. Res. 5, 101–114 (2004)
4. Chen, J.C., Yuan, B.: Detecting functional modules in the yeast protein-protein interaction network. Bioinformatics 22, 2283–2290 (2006)
5. Rives, A.W., Galitski, T.: Modular organization of cellular networks. Proc. Natl. Acad. Sci. 100, 1128–1133 (2003)
6. Luo, F., et al.: Modular organization of protein interaction networks. Bioinformatics 23, 207–214 (2007)
7. King, A.D., Pržulj, N., Jurisica, I.: Protein complex prediction via cost-based clustering. Bioinformatics 20, 3013–3020 (2004)
8. Hartuv, E., Shamir, R.: A clustering algorithm based graph connectivity. Information Processing Letters, 175–181 (2000)
9. Girvan, M., Newman, M.E.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. 99, 7821–7826 (2002)
10. Pereira-Leal, J.B., et al.: Detection of functional modules from protein interaction networks. Proteins: Struct. Func. Bioinformatics 54, 49–57 (2004)
11. Arnau, V., et al.: Iterative cluster analysis of protein interaction data. Bioinformatics 21, 364–378 (2005)
12. Adamcsek, B., Palla, G., Farkas, I., Derenyi, I., Vicsek, T.: CFinder: locating cliques and overlapping modules in biological networks. Bioinformatics 22, 1021–1023 (2006)
13. Cho, Y.R., Hwang, W., Ramanathan, M., Zhang, A.: Semantic integration to identify overlapping functional modules in protein interaction networks. BMC Bioinformatics, 8–265 (2007)
14. Spirin, V., Mirny, L.A.: Protein complexes and functional modules in molecular networks. Proc. Natl. Acad. Sci., 12123–12128 (2003)
15. Bader, G.D., Hogue, C.W.: An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinformatics 4(1) (2003)
16. Altaf-Ul-Amin, M., et al.: Development and implementation of an algorithm for detection of protein complexes in large interaction networks. BMC Bioinformatics, 7(207) (2006)
17. Pei, P., Zhang, A.: A seed-refine algorithm for detecting protein complexes from protein interaction data. IEEE Transactions on Nanobioscience 6, 43–50 (2007)

18. Palla, G., et al.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435, 814–818 (2005)
19. Jeong, H., Tombor, B., Albert, R., Oltvai, Z.N., Barabasi, A.L.: The large-scale organization of metabolic networks. Nature 407, 651–654 (2000)
20. Jeong, H., Mason, S., Barabási, A., Oltvai, Z.: Lethality and centrality in protein networks. Nature 411, 41–42 (2001)
21. Yook, S., Oltvai, Z., Barabasi, A.: Functional and topological characterization of protein interaction networks. Proteomics 4, 928–942 (2004)
22. Pržulj, N., Wigle, D.A., Jurisica,I.: Functional topology in a network of protein interactions. Bioinformatics 20(3), 340–348 (2004)
23. Ucar, D., Asur, S., Catalyurek, U., Parthasarathy, S.: Improving functional modularity in potein-protein interactions graphs using Hub-induced subgraphs. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, Springer, Heidelberg (2006)
24. Mewes, H.W., et al.: MIPS: analysis and annotation of proteins from whole genome in 2005. Nucleic Acid Research 34, 169–172 (2006)
25. Ruepp, A., et al.: The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. Nucleic Acid Research 32, 5539–5545 (2004)
26. `http://dip.doe-mbi.ucla.edu/`
27. Hwang, W., Cho, Y.R., Zhang, A., Ramanathan, M.: A novel functional module detection algorithm for protein- protein interaction networks. Algorithms for Molecular Biology 12, 1–24 (2006)
28. Cho, Y.R., Hwang, W., Zhang, A.: Identification of overlapping functional modules in protein interaction networks: information flow-based approach. In: Perner, P. (ed.) ICDM 2006. LNCS (LNAI), vol. 4065, Springer, Heidelberg (2006)