

Research Article

Hierarchical Modular Structure Identification with Its Applications in Gene Coexpression Networks

Shuqin Zhang

Center for Computational Systems Biology, School of Mathematical Sciences, Fudan University, Shanghai 200433, China

Correspondence should be addressed to Shuqin Zhang, zhangs@fudan.edu.cn

Received 2 November 2012; Accepted 25 November 2012

Academic Editors: R. Jiang, W. Tian, J. Wan, and X. Zhao

Copyright © 2012 Shuqin Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Network module (community) structure has been a hot research topic in recent years. Many methods have been proposed for module detection and identification. Hierarchical structure of modules is shown to exist in many networks such as biological networks and social networks. Compared to the partitional module identification methods, less research is done on the inference of hierarchical modular structure. In this paper, we propose a method for constructing the hierarchical modular structure based on the stochastic block model. Statistical tests are applied to test the hierarchical relations between different modules. We give both artificial networks and real data examples to illustrate the performance of our approach. Application of the proposed method to yeast gene coexpression network shows that it does have a hierarchical modular structure with the modules on different levels corresponding to different gene functions.

1. Introduction

Networks are widely applied to model complex systems, including biological systems, social organizations, World-Wide-Webs, and so on. In a network, the nodes (vertices) represent the members in the system, while the edges represent the interactions among the members. If two nodes have interactions in a network, there will be an edge connecting them. With such a representation, the complex systems can be analyzed by computational methods.

Module (community) structure is a common property of many different types of networks. Modules are the dense subgroups of a network, where the nodes in the same module are more likely to connect each other than the nodes in other modules. In general, the members in the same module share some common properties or play similar roles. For example, in a gene coexpression network, the genes in the same module may belong to the same functional category such as lipid metabolism and acute-phase response [1]. Since the paper published by [2], module detection and identification becomes a hot research topic in several different areas such as computer science, physics, and statistics. A large number of related works have been published with the physicists making the most contributions [3–12]. Several recent review

papers provide details and comparisons of the module identification methods [6, 9, 13]. Reference [13] compares the performance of several existing methods for both computation time and output. Reference [6] is a thorough, more recent discussion. Reference [9] contrasts different perspectives of the methods and sheds light on some important similarities of several methods. A recent comparison of some popular methods is shown in [14]. Among the compared methods, the method by maximizing the average degree within modules and minimizing the average connections between different modules outperforms other methods in identification accuracy. Its computational speed is also competitive [14]. Besides these computational methods, theoretical analysis on module identifications is presented very recently. Bickel and Chen gave the first statistical analysis on the properties of modules [15]. There based on the stochastic block model, they gave the sufficient conditions for a modularity to be a consistent estimator of modules and presented a new consistent modularity. However, the computation of maximizing this modularity is very time consuming.

Although so many related works are published, how to choose an appropriate number of modules keeps being an open problem. Different methods output different solutions when they are applied to the same network. In reality, all

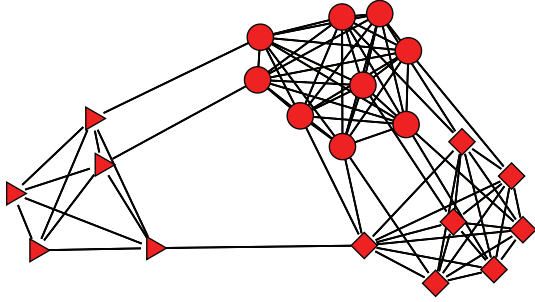


FIGURE 1: Example of hierarchical modular network structure.

of the different choices may be reasonable because different choices of this number may correspond to the modules on different levels. As explained in [16], some modular networks may have hierarchical structure. For example, in a friendship network, on the large scale, the modules may correspond to people from different countries. On the smaller scales, people in the same module may graduate from the same university, grow up in the same community, or even be born in the same family. Such hierarchical modular structure appears in different kinds of networks. For example, Meunier and colleagues gave an example on hierarchical modular structures in human brains [17]. Figure 1 shows an example of hierarchical modular network. There are two levels of the modules. We can identify three modules corresponding to different shapes of nodes on the lowest level or two modules with nodes represented by cubes and circles being combined together on the higher level.

Compared to the module identification in a partitional way (all the modules are on the same level), there are much fewer works on computational methods for hierarchical modular structure analysis [18–20]. Although these papers present some methods to construct the hierarchical modular structure, they do not give a clear picture on how these modules are organized and what the relationship among the modules is. In this paper, we mainly consider the problem of hierarchical modular structure in unweighted networks. Based on the module identification method presented in [14], we give the method on how to construct the hierarchical structure from all the possible modules in Section 2. Numerical experiments for both simulated networks and real data networks are presented to show the performance of our proposed method in Section 3. The application of the proposed method to yeast gene coexpression network shows that it does have a hierarchical structure, which corresponds to the different levels of gene functions. Conclusion remarks are given finally. By constructing the hierarchical structure, we aim to explore the functions of modules on different levels and explain why the number of modules may differ for different identification methods.

2. Methodology

Before going to the details on how to construct the hierarchical structure, we give its definition first. We consider a network $G(V, E)$ with n nodes, where V denotes the set of

nodes and E denotes the set of edges. The adjacency matrix is denoted as A with each entry being 0 or 1. The hierarchical structure of a network is defined based on the stochastic block model, which is a direct extension of the Erdős-Rényi random graph model [21]. The network is obtained by starting with a set of n nodes and adding edges between them in a probabilistic fashion. The presence of an edge between any two nodes is a Bernoulli event where the probability may be vertex-pair dependent. At the beginning, we assume there are K modules in the network. The network is generated in two steps. First, any node is assigned to a module M_i with a probability μ_i , where $\mu = (\mu_1, \mu_2, \dots, \mu_K)$ satisfies $\sum_{i=1}^K \mu_i = 1$. Then any two nodes $u, v \in V$ and $u \in M_i, v \in M_j$ are connected with probability $P_{i,j}$ depending on M_i, M_j , and P is symmetric. If there is the modular structure in the network, then $P_{i,j} < \min\{P_{i,i}, P_{j,j}\}$. With this model, the hierarchical structure of a network can be defined recursively. For any three modules M_i, M_j , and M_k , if $P_{i,j} > \max\{P_{i,k}, P_{j,k}\}$, we say there is hierarchical structure among these three modules and M_i, M_j can be combined to a new module parallel to M_k .

To construct the hierarchical structure, we use the bottom-up strategy. We first find all the possible modules on the lowest level and then build the hierarchical structure. We use the method presented in [14] to find all the possible modules. Suppose K is given first. We let N_k denote the number of nodes in subnetwork V_k , L_{kk} denote twice the total number of edges in subnetwork V_k , and L_{kl} denote the total number of connections between the subnetworks V_k and V_l , where $k, l = 1, 2, \dots, K$. The module identification problem is formulated as

$$\max_{\mathbf{P}} \Phi(\mathbf{P}) = \sum_{k=1}^K \frac{L_{kk}}{N_k} - \sum_{k=1}^K \sum_{l \neq k} \frac{L_{kl}}{N_k}, \quad (1)$$

where \mathbf{P} is a partition of the network.

In matrix form, if we let

$$S_{ik} = \begin{cases} 1, & \text{if node } i \in V_k \\ 0, & \text{otherwise} \end{cases} \quad i = 1, 2, \dots, n, \quad (2)$$

the problem is formulated as

$$\begin{aligned} \max \Psi(S) &= \sum_{k=1}^K \frac{S_{\cdot,k}^T A S_{\cdot,k}}{S_{\cdot,k}^T S_{\cdot,k}} - \sum_{k=1}^K \sum_{l \neq k} \frac{S_{\cdot,k}^T A S_{\cdot,l}}{S_{\cdot,k}^T S_{\cdot,k}} \\ &= \sum_{k=1}^K \frac{S_{\cdot,k}^T (2A - D) S_{\cdot,k}}{S_{\cdot,k}^T S_{\cdot,k}} \end{aligned} \quad (3)$$

$$\text{s.t. } S_{i,j} \in \{0, 1\} \quad \text{for } i, j = 1, 2, \dots, K,$$

$$\sum_{k=1}^K S_{\cdot,k} = \mathbf{1}.$$

Here $\mathbf{1}$ is a vector with all elements being 1.

The objective function aims to both maximize the average degree within each module and minimize the average connections between different modules. We expect to achieve a good balance of the module size and make correct inference

on the modules. The problem (3) is solved with an approximate method similar to the spectral clustering. We first compute the K eigenvectors of the matrix $2A - D$. By clustering these K eigenvectors as a matrix of n objects with K dimensions, we get the assignment of the n nodes into K modules.

Now, we discuss how to determine the lowest level of all the possible modules K . For any node $i \in V$, the degree can be written as

$$d_i = \sum_{k=1}^K d_i(V_k), \quad (4)$$

where

$$d_i(V_k) = \sum_{j \in V_k} A_{ij}, \quad (5)$$

which defines the connections that node i has in the subnetwork V_k . To determine the number of possible modules, we compare the average connectivity within a subnetwork and the average connectivity between it and any other subnetwork. If the average connectivity within a subnetwork is greater, we take it as a module, that is,

$$\frac{\sum_{i \in V_k} d_i(V_k)}{N_k} > \frac{\sum_{i \in V_k} d_i(V_l)}{N_k}, \quad l \neq k. \quad (6)$$

Alternatively, it can also be written as

$$L_{kk} > L_{kl}, \quad (7)$$

if we multiply both sides with N_k . This condition is very weak, thus with it, we hope we find all the modules as on the lowest level. We do the clustering for K increasing from two until the condition (6) does not hold and get all the possible modules. The efficiency of the above algorithm can be seen in [14].

Based on the above results, we construct the hierarchical structure in an agglomerative way (bottom-to-up). We directly use connection probability, which is computed from the clustering results through maximum likelihood estimation, to measure the distance between different modules. This connection probability matrix is denoted as \hat{P}^0 . First the maximum connection probability between different modules is found, and we assume it is \hat{P}_{i_0, j_0}^0 with the corresponding two modules i_0, j_0 being recorded. The second largest connection probability for these two modules i_0, j_0 are also found, and we assume they are \hat{P}_{i_0, k_0}^0 and \hat{P}_{j_0, l_0}^0 with the corresponding modules being k_0 and l_0 . To determine whether there is a hierarchical structure for these modules, we use Fisher exact test to see whether the connection probabilities \hat{P}_{i_0, k_0}^0 and \hat{P}_{j_0, l_0}^0 are the same as \hat{P}_{i_0, j_0}^0 . That is, we need to test $\hat{P}_{i_0, j_0}^0 = \hat{P}_{i_0, k_0}^0$ and $\hat{P}_{i_0, j_0}^0 = \hat{P}_{j_0, l_0}^0$. Here we take a P value threshold to be 0.05. Three different cases may occur for these two relations. (1) Both of these two null hypotheses are rejected. In this case, there is hierarchical structure and the modules i_0, j_0 are on the lower level than k_0 and l_0 . We combine the two modules i_0 and j_0 and take them as one module. (2) Only one of $\hat{P}_{i_0, j_0}^0 = \hat{P}_{i_0, k_0}^0$ and $\hat{P}_{i_0, j_0}^0 = \hat{P}_{j_0, l_0}^0$ is accepted. The

corresponding modules having the same connection probability are combined together. We look for the next largest connection probability for these three modules and test the relationship again. If two modules are tested to have the same connection probability, they are combined into one group, and the same step is implemented again. (3) Both of these two null hypotheses are accepted. These modules are taken as on the same level and combine together. We search the next largest connection probability to these four modules and do the statistical test until the hierarchical structure occurs or all the modules are combined together. After the above steps are finished, the connection probability between different modules is recalculated and recorded as \hat{P}^1 . The above search and test steps are repeated for \hat{P}^1 . Such steps are implemented recursively until all the modules are combined into one big module/network. For the statistical tests, we can also use t -test to test the relations between the connection probabilities if the distribution of the connections between different modules can be approximated by normal distribution. With this method, we can efficiently combine the modules with the same connection probability into the same level.

3. Numerical Experiments

In this section, we evaluate the performance of our proposed method through its application to several examples. We first start with two artificial networks having comparatively clear module structures. We then apply our method to two real networks to evaluate its performance. The first real network is the well-known karate club network and the second one is a yeast gene coexpression network.

3.1. Artificial Networks

3.1.1. A Network Composed of Cliques. We consider a network with 200 nodes, which is composed of 4 cliques. The sizes of the cliques are 90, 30, 40, and 40. The connections between different cliques are randomly generated with the following probability:

$$P = \begin{pmatrix} 1.000 & 0.200 & 0.002 & 0.003 \\ 0.200 & 1.000 & 0.005 & 0.010 \\ 0.002 & 0.005 & 1.000 & 0.030 \\ 0.003 & 0.010 & 0.030 & 1.000 \end{pmatrix}. \quad (8)$$

The pattern of the adjacency matrix is shown in Figure 2(a). From upper-left to lower-right, we denote the four modules as M_1, M_2, M_3 , and M_4 , which correspond to the position in the connection probability matrix. We can see the hierarchical structure of the network from the adjacency matrix. We apply our proposed method to this network. The condition (6) is satisfied until $K = 4$. The estimated connection probability matrix is

$$\hat{P} = \begin{pmatrix} 1.000 & 0.205 & 0.003 & 0.003 \\ 0.205 & 1.000 & 0.006 & 0.009 \\ 0.003 & 0.006 & 1.000 & 0.029 \\ 0.003 & 0.009 & 0.029 & 1.000 \end{pmatrix}. \quad (9)$$

We apply statistical tests to the corresponding modules, and finally we get the hierarchical structure as shown in

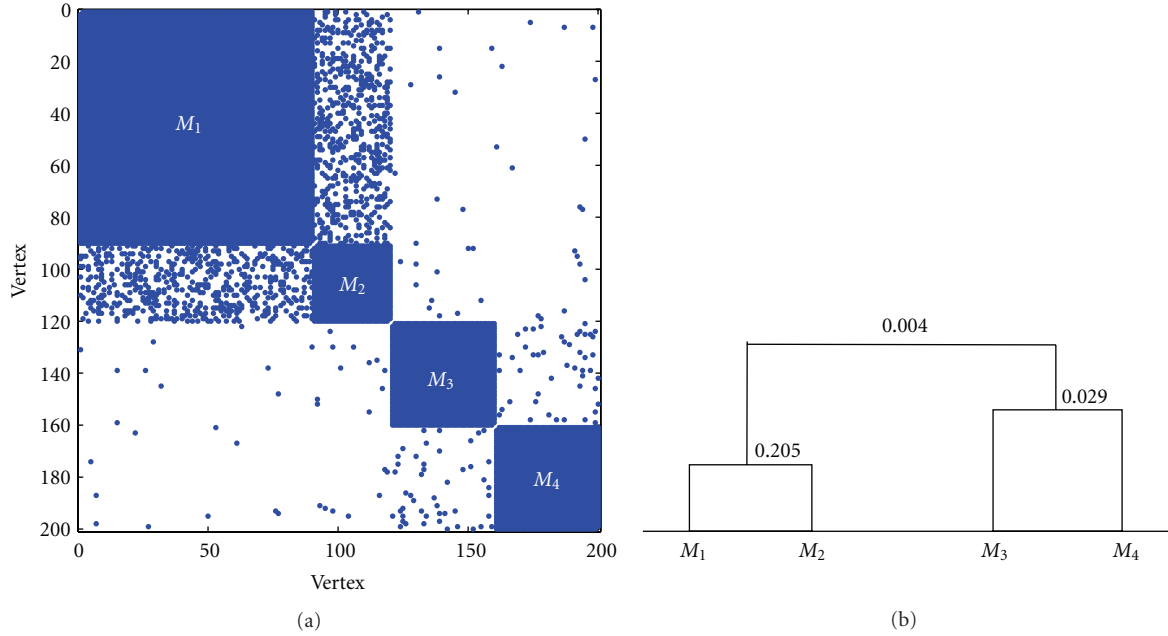


FIGURE 2: Example of hierarchical modular network structure. (a) Pattern of the adjacency matrix; (b) the hierarchical structure of the network.

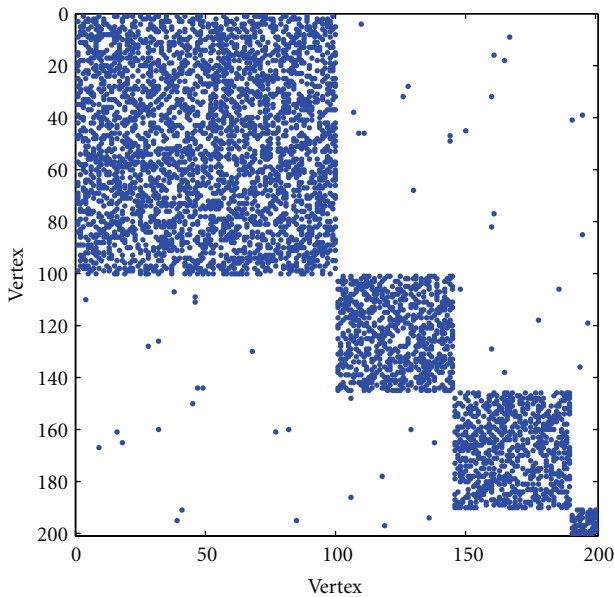


FIGURE 3: Pattern of the adjacency matrix for the randomly generated network.

Figure 2(b). The values on the hierarchical tree is the estimated connection probability of the corresponding modules. On the lowest level, there are four modules. If the tree is cut between 0.205 and 0.029, there are three modules while if the cutoff is greater than 0.029, there are only two modules. These results are consistent with the network generation strategy.

3.1.2. A Randomly Generated Network. In this example, we also consider a network with 200 nodes and 4 modules.

The size of each module is 10, 45, 45, and 100. We set the degree of each node within its module to be 6, 15, 15, and 30. Then the connections between different nodes are randomly generated. We keep all the edges generated for each node. So finally the average degree within each module is greater than the prespecified number. The connection probability between different modules is 0.002. The pattern of the adjacency matrix is shown in Figure 3. From upper-left to lower-right, the four modules are M_1 , M_2 , M_3 , and M_4 , respectively. With our proposed method, the network is partitioned into four modules correctly on the lowest level and the estimated connection probability is

$$\hat{p} = \begin{pmatrix} 0.298 & 0.002 & 0.002 & 0.003 \\ 0.002 & 0.328 & 0.002 & 0.004 \\ 0.002 & 0.002 & 0.321 & 0.000 \\ 0.003 & 0.004 & 0.000 & 0.560 \end{pmatrix}. \quad (10)$$

By using the statistical tests, these four modules are determined as parallel modules, which is the same as that in our network generation strategy.

3.2. Karate Club Network. We consider the Zachary's network of karate club members [22] in this example. There are 34 nodes in this network corresponding to the members in a karate club. This dataset has been applied as a benchmark to test many module identification algorithms since the true modules are known in this network. The people in the club were observed for a period of three years. The edges represent connections of the individuals outside the activities of the club. At some point, the administrator and the instructor of the club broke up due to a conflict between them. The club was separated into two groups supporting the administrator and the instructor. Figure 4 shows the network. Originally, there are two modules, which have 16 nodes (squares and

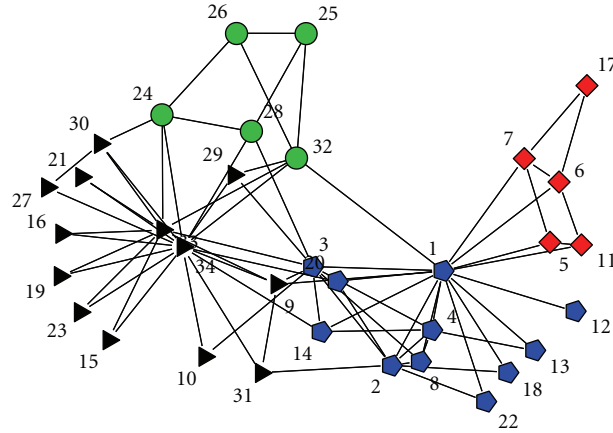


FIGURE 4: Zachary's karate club network. Different shapes show the modules. M_1 : pentagon, M_2 : square, M_3 : triangle, M_4 : circle.

pentagons in the figure) and 18 nodes (circles and triangles in the figure), respectively.

We apply our proposed method to this network. The criterion (6) is satisfied until $K = 4$. The result is shown in Figure 4, with different shapes of the nodes denoting different modules. The estimated connection probability matrix is

$$\hat{P} = \begin{pmatrix} 0.364 & 0.073 & 0.056 & 0.036 \\ 0.073 & 0.480 & 0.000 & 0.000 \\ 0.056 & 0.000 & 0.237 & 0.108 \\ 0.036 & 0.000 & 0.108 & 0.480 \end{pmatrix}. \quad (11)$$

From this matrix, it is easy to see that M_3 and M_4 are more likely to connect each other. With statistical tests, we can get that the connection probability among M_3 , M_4 , and M_1 is the same. Although M_2 has no connections to M_3 and M_4 , it has a larger connection probability to M_1 than M_3 , M_4 to M_1 . Thus these four modules are on the same level. In [19], the authors considered constructing the hierarchical modular structure of this network too. At first, they also found four modules on the lowest level. Then they found that this network has two modules with some nodes (3, 9, 10, 14, 31) belonging to both of them. We did not consider the overlapping nodes in this article. However, we can see that because these overlapping nodes belong to both M_1 and M_3 , and they connect both parts closely, our method detect M_1 and M_3 , M_3 and M_4 as having the same connectivity.

3.3. Hierarchical Modular Structure in Yeast Gene Coexpression Network. In this section, we apply our proposed approach to analyze a gene coexpression network of yeast. The data set we use was generated by Brem and Kruglyak from a cross between two distinct isogenic strains BY and RM [23]. As described in [23], a total of 5740 ORFs were obtained after data preprocessing. In our analysis, we only use the 1,800 most differentially expressed genes as input to construct coexpression network and derive modules. When constructing the adjacency matrix of the network, we use the hard thresholding, that is: if the absolute value of Pearson correlation coefficient between two genes is greater than some given value, we assign an edge between them; otherwise, there is no edge. We compute the linear regression

coefficient between the frequency of degree d ($\log_{10}(f(d))$) and the \log_{10} transformed degree d ($\log_{10}(d)$), and choose the threshold that leads to approximately scale free property of the network as described in [24]. Finally, the threshold is set to be 0.705, \hat{R} is about 0.75. By such a setting, this gene coexpression network is divided into 690 unconnected parts with the largest part of size 788. Here, we only analyze the hierarchical modular structure of the largest connected network.

Starting from $K = 2$, we apply the method in [14] to this network, and the condition (6) holds until $K = 10$. To make the solution more accurate, we do a global maximization by changing the module index of boundary nodes starting from the approximate solution. Since the approximate solution is already good, this step is very fast. The structure of the network is shown in Figure 5(a), with different colors and shapes denoting different modules as described in Table 1. Then we construct the hierarchical modular structure as shown in Figure 5(b). On the lowest level, there are ten modules, while on the highest level, there are four modules.

Since coexpressed genes tend to be coregulated and possibly have similar functions, genes in the same module are expected to be enriched for some function categories. In order to understand the biological basis of the network modules, we consider each identified module for enrichment of annotations from gene ontology (GO) [25]. In our analysis, the enrichment analysis was performed by GO stats from Bioconductor. For each module, the statistically most significant GO categories are analyzed. Table 1 shows the enrichment results for the ten modules. "M-size" and "G-size" are the size of both the modules and the GO categories, respectively. "Overlap" is the overlap size of the module and the GO category. Table 2 shows the enrichment results for the modules on different levels. From the tables, it is easy to see that different gene function categories are enriched most on different levels. For example, module M_2 enriches the GO category "translation" most significantly, while the combined module M_2, M_8 enriches "Ribonucleoprotein complex biogenesis" most significantly, with M_2 containing 42 genes having this function. The combined module M_2, M_8, M_4 , and M_1 also enriches this function, while M_4 itself enriches

TABLE 1: GO enrichment analysis results of the gene modules on the lowest level.

Module	Color, shape	M -size	Enriched GO category	P value	G -size	Overlap
M_1	White, square	190	Cellular carbohydrate metabolic process	3.23×10^{-9}	60	35
M_2	White, circle	126	Translation	4.70×10^{-59}	101	80
M_3	Grey, triangle	135	Organic acid biosynthetic process	5.41×10^{-35}	89	64
M_4	Grey, pentagon	62	Cellular respiration	4.13×10^{-27}	36	28
M_5	Black, circle	12	Amino acid catabolic process to alcohol via Ehrlich pathway	1.76×10^{-7}	5	4
M_6	Black, circle	13	Steroid biosynthetic process	2.20×10^{-15}	13	9
M_7	White, pentagon	19	Branched chain family amino acid metabolic process	4.37×10^{-8}	11	6
M_8	Grey, triangle	209	Ribonucleoprotein complex biogenesis	5.94×10^{-39}	149	106
M_9	Grey, square	11	Protein targeting to membrane	8.91×10^{-6}	4	3
M_{10}	White, square	11	Regulation of translational termination	1.55×10^{-4}	2	2

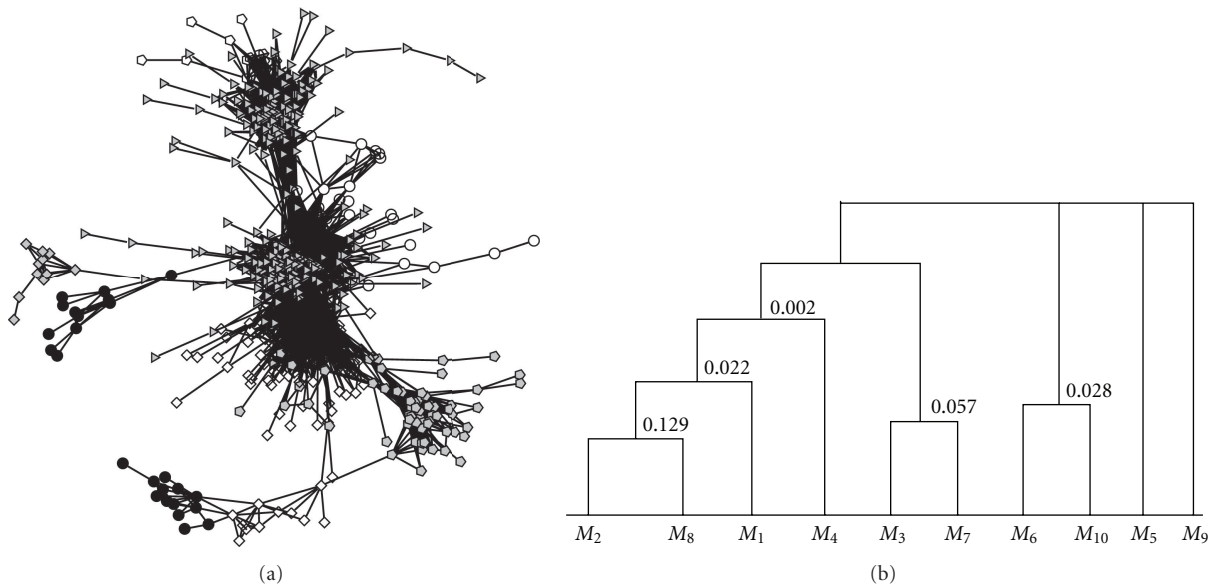


FIGURE 5: Yeast gene coexpression network. (a) The network structure, (b) the hierarchical structure.

“cellular respiration” significantly. On the uppermost level, the module composed of M_2 , M_8 , M_1 , M_4 , M_3 , and M_7 enriches four GO function categories most significantly, and all the genes are overlapped. Three (“cellular component biogenesis,” “cellular component biogenesis at cellular level,” and “ribosome biogenesis”) of them are different from the most enriched gene functions for each of these six modules. All these results indicate that hierarchical modular structure does exist in gene coexpression networks and different gene functions are enriched most on different levels.

We use the software REViGO to check the hierarchical structure of the enriched GO categories [26]. We consider the enriched GO categories in Tables 1 and 2 except the category “regulation of translational termination” because its G -size is very small and the P value is comparatively large. Figure 6 shows the tree map of the most enriched GO categories. The subgraph that we do not mark with the module corresponds to the combined module M_1 , M_2 , M_3 , M_4 , M_7 , M_8 . Here the modules M_6 , M_9 and other modules are parallel to each other, which is consistent with our results. M_3 and M_7 belong to a large category, which is “branched chain family amino

acid metabolic process”. This large category is different from the most enriched category for the combined module M_3 and M_7 . This may come from the fact that since M_7 is very small, it does not cover a large part of its enriched category. M_1 and M_4 are parallel to each other which is also consistent with our analysis. All these results show that our proposed method can explain some of the hierarchical structure of the GO categories. Due to the network size, we did not handle all the genes of yeast. This may be a reason why some of our computational results are not consistent with the GO function tree map.

4. Conclusion

Module identification problem has attracted much attention from different fields and it continues being a hot research topic. How to determine the number of modules in a modular network has been an open problem during the study of module identification methods. This problem may come from the hierarchical structure of modular networks. The different numbers correspond to the different levels of

TABLE 2: GO enrichment analysis results of gene modules on the upper level.

Module	M-size	Enriched GO category	P value	G-size	Overlap
M_2, M_8	335	Ribonucleoprotein complex biogenesis	4.02×10^{-66}	149	148
M_1, M_2, M_8	525	Ribonucleoprotein complex biogenesis	1.33×10^{-29}	149	148
M_1, M_2, M_4, M_8	587	Ribonucleoprotein complex biogenesis	6.04×10^{-23}	149	149
M_3, M_7	154	Organic acid biosynthetic process	9.22×10^{-40}	89	71
M_1, M_2, M_3, M_4	741	Cellular component biogenesis	4.01×10^{-6}	175	175
M_7, M_8		Cellular component biogenesis at cellular level	1.84×10^{-5}	156	156
		Ribonucleoprotein complex biogenesis	3.19×10^{-5}	149	149
		Ribosome biogenesis	3.44×10^{-5}	148	148
M_6, M_{10}	24	Steroid biosynthetic process	2.36×10^{-19}	13	12

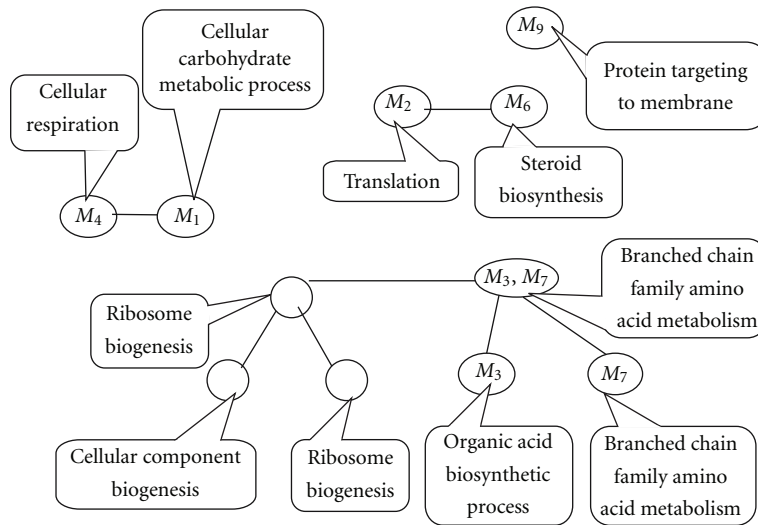


FIGURE 6: Tree map of the enriched GO categories in yeast gene coexpression network.

the hierarchical structure and they may be all reasonable. In this paper, we proposed a method for constructing the hierarchical modular structure of networks. With statistical tests, we can identify both the parallel modules and the hierarchical structure. According to different cutoffs of the hierarchical tree, different numbers of modules can be identified. This may solve the problem of the number of network modules to some extent. Several examples are given to demonstrate the efficiency of our method. Application of this method to gene coexpression networks shows that there are hierarchical modules in yeast gene coexpression network. On different levels of such networks, the genes in the module belong to different gene functions most. Thus studying the gene function through constructing the hierarchical modular structure instead of specifying the number of modules should perform better. Application of such algorithms to other kinds of networks may also contribute to other research fields.

Acknowledgments

This work was supported in part by NSFC Grants 10901042, 10971075, and 91130032. The primary version of this paper has appeared in IEEE ISB 2012.

References

- [1] R. Guimerà and L. A. N. Amaral, "Functional cartography of complex metabolic networks," *Nature*, vol. 433, no. 7028, pp. 895–900, 2005.
- [2] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [3] A. Arenas, J. Borge-Holthoefer, S. Gómez, and G. Zamora-López, "Optimal map of the modular structure of complex networks," *New Journal of Physics*, vol. 12, Article ID 053009, 2010.
- [4] J. Dong and S. Horvath, "Understanding network concepts in modules," *BMC Systems Biology*, vol. 1, article 24, 2007.
- [5] E. Estrada and N. Hatano, "Communicability in complex networks," *Physical Review E*, vol. 77, no. 3, Article ID 036111, 2008.
- [6] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3–5, pp. 75–174, 2010.
- [7] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, Article ID 036104, 2006.
- [8] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 23, pp. 8577–8582, 2006.

- [9] M. A. Porter, J. P. Onnela, and P. J. Mucha., “Communities in networks,” *Notices of the American Mathematical Society*, vol. 56, no. 9, pp. 1082–1102, 2010.
- [10] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Paris, “Defining and identifying communities in networks,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 9, pp. 2658–2663, 2004.
- [11] Z. Li, S. Zhang, R.-S. Wang, X.-S. Zhang, and L. Chen, “Quantitative function for community detection,” *Physical Review E*, vol. 77, no. 3, Article ID 036109, 2008.
- [12] M. Rosvall and C. T. Bergstrom, “An information-theoretic framework for resolving community structure in complex networks,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 18, pp. 7327–7331, 2007.
- [13] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, “Comparing community structure identification,” *Journal of Statistical Mechanics*, no. 9, Article ID 09008, pp. 219–228, 2005.
- [14] S. Zhang and H. Zhao, “Community identification in networks with unbalanced structure,” *Physical Review E*, vol. 85, no. 6, Article ID 066114, 2012.
- [15] P. J. Bickel and A. Chen, “A nonparametric view of network models and Newman-Girvan and other modularities,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 50, pp. 21068–21073, 2009.
- [16] A. Arenas, A. Díaz-Guilera, and C. J. Pérez-Vicente, “Synchronization reveals topological scales in complex networks,” *Physical Review Letters*, vol. 96, no. 11, Article ID 114102, 2006.
- [17] D. Meunier, R. Lambiotte, and E. T. Bullmore, “Modular and hierarchically modular organization of brain networks,” *Frontiers Neuroscience*, vol. 4, no. 200, 2010.
- [18] H. Shen, X. Cheng, K. Cai, and M.-B. Hu, “Detect overlapping and hierarchical community structure in networks,” *Physica A*, vol. 388, no. 8, pp. 1706–1712, 2009.
- [19] A. Lancichinetti, S. Fortunato, and J. Kertész, “Detecting the overlapping and hierarchical community structure in complex networks,” *New Journal of Physics*, vol. 11, Article ID 033015, 2009.
- [20] E. Ravasz, “Detecting hierarchical modularity in biological networks,” *Computational Systems Biology*, vol. 54, pp. 145–160, 2009.
- [21] P. Erdős and A. R. Rényi, “On random graphs. I,” *Publicationes Mathematicae*, vol. 6, pp. 290–297, 1959.
- [22] W. W. Zachary, “An information flow model for conflict and fission in small groups,” *Journal of Anthropological Research*, vol. 33, pp. 452–473, 1977.
- [23] R. B. Brem and L. Kruglyak, “The landscape of genetic complexity across 5,700 gene expression traits in yeast,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 5, pp. 1572–1577, 2005.
- [24] B. Zhang and S. Horvath, “A general framework for weighted gene co-expression network analysis,” *Statistical Applications in Genetics and Molecular Biology*, vol. 4, no. 1, article 17, 2005.
- [25] M. Ashburner, C. A. Ball, J. A. Blake et al., “Gene ontology: tool for the unification of biology, the gene ontology consortium,” *Nature Genetics*, vol. 25, no. 1, pp. 25–29, 2000.
- [26] F. Supek, M. Bošnjak, N. Škunca, and T. Šmuc, “REVIGO summarizes and visualizes long lists of gene ontology terms,” *PLoS ONE*, vol. 6, no. 7, Article ID 21800, 2011.