EUSIPCO 2013 1569742599

BLOCKMODEL MODULE IDENTIFICATION IN PROTEIN INTERACTION NETWORKS THROUGH MARKOV RANDOM WALK

Yijie Wang and Xiaoning Qian

Department of Computer Science and Engineering, University of South Florida

ABSTRACT

To identify biologically meaningful modules in large-scale biological networks, general blockmodel network clustering algorithms have recently attracted much attention to search for groups of molecules that have similar interaction patterns in networks. However, existing blockmodel module identification algorithms suffer from the problems of prohibitive computational complexity and being trapped at local optima due to its inherent combinatorial complexity. In this paper, we propose a novel blockmodel module identification formulation based on Markov random walk to address those problems by finding high quality approximate solutions. A new convex optimization problem is formulated to find the low conductance (LC) sets as potential modules based on the two-hop transition matrix of Markov random walk on networks. We further propose a spectral approximate algorithm to find high quality modules in large-scale networks. The experimental results on two real-world PPI (protein-protein interaction) networks demonstrate that our method outperforms the state-of-the-art blockmodel module identification algorithms in terms of the accuracy measured by the F-measure based on curated annotations such as GO (Gene Ontology) and KOG (EuKaryotic Orthologous Groups) categories.

Index Terms— PPI (protein-protein interaction) networks, Blockmodel module identification, Markov random walk, Spectral method

1. INTRODUCTION

With increasingly large amounts of high-throughput protein-protein interaction (PPI) measurements, one of critical challenges in computational systems biology is to understand underlying cellular functional mechanisms by appropriate analysis of genome-scale PPI networks. One promising direction to analyze PPI networks is to first identify potential functional modules by network clustering. However, there is still no widely accepted definition of functional modules to guide functional module identification. Many existing clustering algorithms focus on identifying topologically densely connected modules [6], which may not be adequate for analyzing PPI networks since proteins do not only work together through dense connections. There are other topological structures in PPI networks that may carry essential cellular functionalities. For example, receptors in signal transduction cascades rarely interact with themselves but tend to interact with similar types of cytoplasmic proteins as well as with extracellular ligands [8]. In order to detect functional modules with more general and diverse topological structures, blockmodel module identification algorithms recently have been actively investigated [10, 5, 8, 14, 15].

To name a few blockmodel module identification algorithms, power graph (PG) [10] achieves significant edge reduction by greedily grouping topological similar nodes in the same module; Graph summarization (GS) [5] collects the nodes with similar interaction patterns by compressing a given network using a minimum description length principle. However, both PG and GS follow the bottom-up module identification procedure and are solved based on greedy algorithms; therefore the obtained results by PG and GS do not guarantee the global optimality. To deal with data reliability and dynamic changes in biological networks, blockmodel module identification based on probabilistic models [3, 7] has been proposed but these methods also converge to local optima. Simulated annealing (SA) [8] has been implemented to map the original network to an image graph and solve the resulting non-convex optimization problem with high solution quality. Unfortunately, SA is very time-consuming to guarantee the global optimality. Although several algorithms [14, 15] have been proposed to reduce the computational time with different heuristics, the non-convex optimization formulation for blockmodel module identification is still a barrier for analyzing large-scale PPI networks.

In this paper, we propose a novel formulation to solve the blockmodel module identification problem based on Markov random walk on a given network. By finding the low conductance (LC) sets based on the two-hop transition matrix of the random walk, we propose a convex optimization formulation to detect both densely and sparsely connected modules simultaneously. A spectral approximate algorithm is derived to solve the problem. We apply our method for functional module identification in two large-scale PPI networks. The experimental results demonstrate that our new algorithm outperforms two state-of-the-art blockmodel identification algorithms PG and GS with comparable computational efficiency.

2. TERMINOLOGY

Let $G = (V, E)$ represent a PPI network, in which V ($|V| = n$) denotes the set of nodes (corresponding to proteins) and

E indicates the set of edges (interactions between proteins). Here, G is an undirected graph excluding self-connections and we further assume that G is connected such that there is always at least one path connecting any pair of two nodes in G. The corresponding adjacency matrix $\mathbf{A}$ with each entry $\mathbf{A}_{ij} = 1$ when nodes i and j have interactions and $\mathbf{A}_{ij} = 0$, otherwise.

A random walker on G uniformly randomly selects one of the neighbors to walk next at any given node. Hence, the transition matrix $\mathbf{P}$ for the underlying Markov chain of the random walk can be computed by $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$, where $\mathbf{D}$ is an $n \times n$ diagonal matrix formed with the corresponding node degrees ($d_i = \sum_j \mathbf{A}_{ij}, i = 1, ..., n$) as its diagonal entries. As G is connected, the underlying Markov chain of the random walk on G is irreducible and ergodic and therefore there exists a stationary distribution satisfying $\mathbf{P}^T\pi = \pi$, where $\pi_i = d_i/M, M = \sum_{i=1}^{n} d_i$ [4]. As defined in [4], we similarly define the conductance of a proper subset of states S (subnetwork for potential module) on G as

$$\Phi_{\mathbf{P}}(S, \bar{S}) = \frac{\sum_{i \in S, j \in \bar{S}} \pi_i \mathbf{P}_{ij}}{\sum_{i \in S} \pi_i}, S \cup \bar{S} = V, \qquad (1)$$

which measures how "well-knit" the subnetwork S is to G.

3. BLOCKMODEL BY RANDOM WALK

3.1. LC by Two-hop Transition Matrix $\mathbf{P}^2$

As discussed in [4], network clustering can be formulated as the problem of finding low conductance (LC) sets on G based on the random walk transition matrix $\mathbf{P}$:

$$\min \sum_{i=1}^{k} \Phi_{\mathbf{P}}(S_i, \bar{S}_i) \; s.t. \bigcup_{i=1}^{k} S_i = V; S_i \cap S_j = \varnothing, i \neq j. \quad (2)$$

However, this formulation can only detect densely connected modules. For example, as shown in Fig. 1, we detect the LC sets of three basic types of network motifs (stars, cliques, and bi-cliques) by solving (2). The second column in Fig. 1 illustrates the identified modules marked by the red dash lines. Clearly, the densely connected modules (cliques in Fig. 1) can be correctly partitioned. But seeking LC sets may not detect different types of nodes with different interaction patterns for star and bi-clique modular structures.

In order to discover meaningful modules with more general and diverse topological structures, we propose to acquire the LC sets in G by designing a new conductance definition based on the two-hop transition matrix $\mathbf{P}^2$ ($\mathbf{P}^2 = \mathbf{P} \times \mathbf{P}$) of the random walk. The motivation is that in biological systems, molecules interacting with similar neighbors tend to carry similar cellular functionalities. Hence, nodes with similar interaction patterns (no matter whether densely connected or sparsely connected) should belong to the same functional module. By random walk on G, when two nodes have similar neighbors, they are more likely to transit from each other after two random walk hops compared to the nodes with very different neighbors. Furthermore, the authors of GS [5] also have discovered that the positive cost reduction only exists for

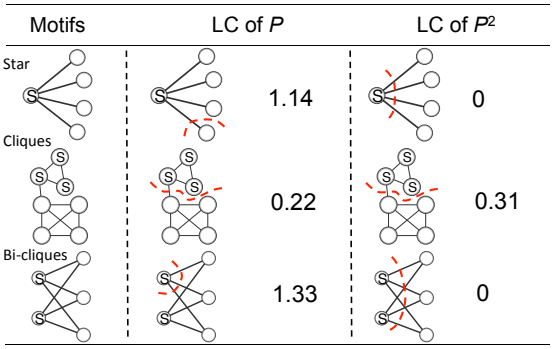

Fig. 1. Module identification results obtained by using $\mathbf{P}$ and $\mathbf{P}^2$. The second and third columns show the clustering results and its corresponding LC value computed by Eq.(2) and Eq.(3) respectively. Red dash line denotes the module dividing line.

pairs of nodes that are two hops away from each other. This further consolidates that it is reasonable to consider two-hop transition matrix $\mathbf{P}^2$ instead of $\mathbf{P}$ for identifying meaningful functional modules in PPI networks. Based on these observations, we propose a novel blockmodel module identification algorithm. Similar to (2), we can detect the LC sets based on the conductance defined by the two-hop transition matrix $\mathbf{P}^2$:

$$\min \sum_{i=1}^{k} \Phi_{\mathbf{P}^2}(S_i, \bar{S}_i) \quad s.t. \bigcup_{i=1}^{k} S_i = V; S_i \cap S_j = \varnothing, i \neq j. \qquad (3)$$

3.2. Convex Optimization for Blockmodel Modules

In this section, we explicitly derive the mathematical formulation for searching for LC sets based on $\mathbf{P}^2$ as potential blockmodel modules.

Proposition 1 $\Phi_{\mathbf{P}^2}(S, S) + \Phi_{\mathbf{P}^2}(S, \bar{S}) = 1.$

Proof:

$$\begin{aligned} & \Phi_{\mathbf{P}^2}(S, S) + \Phi_{\mathbf{P}^2}(S, \bar{S}) \\ = & \frac{\sum_{i \in S, j \in S} \pi_i \mathbf{P}^2_{ij}}{\sum_{i \in S} \pi_i} + \frac{\sum_{i \in S, j' \in \bar{S}} \pi_i \mathbf{P}^2_{ij'}}{\sum_{i \in S} \pi_i} \\ = & \frac{\sum_{i \in S} \pi_i \sum_{j \in V} \mathbf{P}^2_{ij}}{\sum_{i \in S} \pi_i} = 1. \end{aligned} \qquad (4)$$

The last equation is due to the fact that $\mathbf{P}^2$ is a stochastic matrix. □

Proposition 1 allows us to transfer the minimization problem (3) into to a maximization problem for the given number of LC sets k:

Proposition 2 The following optimization problem is equivalent to the optimization problem (3).

$$(P0) \begin{cases} \max & trace\left(\frac{\mathbf{X}^T \mathbf{A} \mathbf{D}^{-1} \mathbf{A} \mathbf{X}}{\mathbf{X}^T \mathbf{D} \mathbf{X}}\right) & \\ s.t. & \mathbf{X} 1_k = 1_n & (a) \\ & \mathbf{X} \in \{0, 1\}^{n \times k} & (b) \end{cases} \qquad (5)$$

where 1_k and 1_n are two all one vectors with k and n elements respectively.

Proof: Based on Proposition 1, we find the following equivalence with the fixed number of modules k.

$$\min \sum_{i=1}^{k} \Phi_{\mathbf{P}^2}\left(S_i, \bar{S}_i\right) \Leftrightarrow \max \sum_{i=1}^{k} \Phi_{\mathbf{P}^2}\left(S_i, S_i\right) \tag{6}$$

Based on (1), we have

$$\begin{aligned}\sum_{i=1}^{k} \Phi_{\mathbf{P}^2}\left(S_i, S_i\right) &= \sum_{i=1}^{k} \frac{\sum_{i\in S_i, j\in S_i} \pi_i \mathbf{P}_{ij}^2}{\sum_{i\in S_i} \pi_i} \\ &= \sum_{i=1}^{k} \frac{\sum_{i\in S_i, j\in S_i} \pi_i \sum_{l=1}^{n} \mathbf{P}_{i,l}\mathbf{P}_{l,j}}{\sum_{i\in S_i} \pi_i} \\ &= \sum_{i=1}^{k} \frac{\sum_{i\in S_i, j\in S_i} \sum_{k=1}^{n} \mathbf{A}_{ik}\mathbf{P}_{kj}}{\sum_{i\in S_i} d_i}\end{aligned}$$

as we recall $\pi_i = d_i/M$ and $\pi_i p_{ij} = \pi_j p_{ji} = \mathbf{A}_{i,j}/M$. We now introduce a binary module assignment matrix X with the corresponding column vector x_i as the indicator vector for the node set $S_i, \forall i$ indicating a potential module. To identify k modules, $\mathbf{X}$ is an $n \times k$ matrix. With that, the above equation can be transformed as

$$\begin{aligned}\sum_{i=1}^{k} \Phi_{\mathbf{P}^2}\left(S_i, S_i\right) &= \sum_{i=1}^{k} \frac{x_i^T \mathbf{A}\mathbf{P} x_i}{x_i^T \mathbf{D} x_i} = \sum_{i=1}^{k} \frac{x_i^T \mathbf{A}\mathbf{D}^{-1}\mathbf{A} x_i}{x_i^T \mathbf{D} x_i} \\ &= trace\left(\frac{\mathbf{X}^T \mathbf{A}\mathbf{D}^{-1}\mathbf{A}\mathbf{X}}{\mathbf{X}^T\mathbf{D}\mathbf{X}}\right).\end{aligned}$$

Combining this with the non-overlap module constraints (5a), (5b), we can derive that the original minimization problem in (3) is equivalent to (5). □

We can further transform the problem as:

$$(P)\begin{cases}\max & trace\left(\mathbf{Y}^T\mathbf{W}\mathbf{Y}\right) \\ s.t. & \mathbf{Y}^T\mathbf{Y} = \mathbf{I}_k,\end{cases} \tag{7}$$

where $\mathbf{Y} = \mathbf{D}^{1/2}\mathbf{X}\left(\mathbf{X}^T\mathbf{D}\mathbf{X}\right)^{-1/2}$ denotes the relaxed clustering indicator matrix and $\mathbf{W} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1/2}$. Note that this relaxed problem (P) is a convex optimization problem as $\mathbf{W}$ is positive semidefinite and the constraint sets forms a convex hull.

Algorithm 1 Spectral Algorithm for (P)

Input: Adjacency matrix $\mathbf{A}$, diagonal degree matrix $\mathbf{D}$ and number of modules k

Output: Module assignment matrix X

1. Compute $\mathbf{W} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1/2}$
2. Find largest k eigenvalues and their corresponding eigenvectors $[\mathbf{E}, \mathbf{V}\mathbf{E}] = eig(\mathbf{W}, k)$
3. Obtain the final module assignment by k-means: $[\mathbf{X}] = kmeans(\mathbf{V}\mathbf{E}, k)$

3.3. Spectral Approximate Algorithm

The procedure of solving (P) is in fact similar to the general spectral clustering problem [16] as shown in Algorithm 1, in which Line 2 uses the truncated eigenvectors of $\mathbf{W}$ (top k eigenvectors) to represent the original data; Line 3 uses the standard k-means method to approximately solve blockmodel module identification based on the top k eigenvectors. The time complexity of our algorithm is dominated by the time to compute eigenvectors, which is in the order $O(n^3)$.

4. RESULTS

In this section, we compare our algorithm with the state-of-the-art algorithms on two real-world PPI networks to show that our algorithm is superior to other existing blockmodel module identification algorithms. Also, we give several functional modules obtained by our algorithm, both densely and sparsely connected, to demonstrate the potential of our method for functional module identification.

4.1. Data and Metrics

We have run the experiments on two PPI networks, which are the *Saccharomyces cerevisiae* (*Sce*) PPI network obtained from DIP (Database of Interacting Proteins) [11] and the *Homo sapiens* (*Hsa*) PPI network extracted from HPRD (Human Protein Reference Database) [9], respectively. Because we do not have the ground truth of the functional organization of these two networks, we evaluate our module identification performance based on GO (Gene Ontology) terms [1] and KOG (euKaryotic Orthologous Groups) categories [13]. GO terms is a set of hierarchical annotations, which covers three domains: molecular function (F), biological process (P) and cellular component (C). In this paper, we only use the GO terms whose information content (IC) is higher than 2 to make sure we do not compare with large sets of proteins at too coarse levels. The IC of a GO term g is defined as IC $= -\log\left(|g|/|root|\right)$ [12], where "$root$" is the corresponding root GO term (either F, P or C) of g. KOG is another annotation which classifies the proteins from seven eukaryotic genomes into 25 functional categories (such as T, K, R, A ...) [13].

Table 1. Information of the two real-world PPI networks

Network	$\|V\|$	$\|E\|$	$\|V \in GO\|$			$\|V \in KOG\|$
			F	P	C	
Hsa	9270	36917	5231	6603	3865	4775
Sce	4490	21911	2364	3333	2562	3005

In order to evaluate the performance based on GO terms, we first compute the neighborhood affinity $NA(a, b)$ between an identified module a containing a set of proteins (V_a) and a given GO term b annotated to another set of proteins V_b: $NA(a, b) = \frac{|V_a \cap V_b|^2}{|V_a| \times |V_b|}$. If $NA(a, b) > \omega$, then the module a and the GO term b are considered to be a match at the level of ω. Let $C = \{c_1, c_2, ..., c_k\}$ denote the identified modules and $G = \{g_1, g_2, ..., g_l\}$ denote the selected GO terms. We can calculate the number of predicted modules that match at least one GO term, denoted by N_{cp}: $N_{cp}^{GO} = |\{c_i \in C | NA(c_i, g_j) > \omega, \exists g_j \in G\}|$. Similarly, we can get the number of GO terms that match at least one identified module N_{cb}: $N_{cg}^{GO} = |\{g_i \in G | NA(c_i, g_j) > \omega, \exists c_i \in C\}|$. In this paper, we set $\omega = 0.2$ as in [12]. Based on these numbers, we can further compute the precision and recall: precision $= \frac{N_{cp}^{GO}}{|C|}$, recall $= \frac{N_{cg}^{GO}}{|G|}$. The final F-measure [12] for performance evaluation is the harmonic mean of precision and recall: $F = 2 \times$ precision $\times$ recall$/$(precision $+$ recall).

Similarly, F-measure of KOG categories can be computed as following. Given 25 KOG categories $KOG = \{kog_1, ..., kog_{25}\}$ and $C = \{c_1, ..., c_k\}$, let T_{ij} indicate the number of proteins annotated by the ith KOG category in the jth module. We consider a match between a given module and any specific KOG category if the majority of the proteins in the module are annotated by that KOG category. Mathematically, we have $N_{cp}^{KOG} = |\{c_j \in C \mid T_{ij} > |c_j|/2\}|$, $N_{cg}^{KOG} = |\{kog_i \in KOG \mid T_{ij} > |c_j|/2\}|$.The precision and recall can be computed by precision $= \frac{N_{cp}^{KOG}}{|C|}$, recall $= \frac{N_{cg}^{KOG}}{|KOG|}$.

4.2. Comparison with Other Blockmodel Algorithms

We compare our method with two state-of-the-art blockmodel module identification algorithms—PG [10] and GS [5], which are representative open source methods so that we can have a fair and objective comparison. Both PG and GS are hierarchical bottom-up algorithms, which do not have tuning parameters while the number of modules k, which controls the module granularity, is the only parameter in our algorithm. We can obtain fine-grained modules with large k and coarse-grained modules when k is small. In order to demonstrate that the number of modules k has little influence on the performance of our algorithms, we compare both coarse-grained and fine-grained results from our method with the results obtained by PG and GS.

Table 2. Performance comparison for the *Sce* PPI network

Method	#modules	coverage	N_{cp}^{GO}	N_{cg}^{GO}	N_{cp}^{KOG}	N_{cg}^{KOG}
Ours (400)	207	3862	58	144	42	15
Ours (900)	376	3431	87	187	73	17
PG	314	2076	21	34	30	14
GS	370	1645	4	6	36	12

Tables 2 and 3 give the detailed comparative results of the three methods in the *Sce* PPI network and the *Hsa* PPI network, respectively. For the *Sce* PPI network, we set $k = 400$ to get coarse-grained modules and $k = 900$ for fine-grained modules. For the *Hsa* PPI network, we set $k = 800$ to get coarse-grained modules and $k = 1700$ for fine-grained modules. For each method, we have listed the number of modules discovered (# modules, which contain at least 3 proteins), the number of proteins covered by legitimate modules (coverage), N_{cp}^{GO}, and N_{cg}^{GO}. We also have listed N_{cp}^{KOG} and N_{cg}^{KOG} for the evaluation based on KOG categories.

Table 3. Results comparison in the *Hsa* PPI network

Method	#modules	coverage	N_{cp}^{GO}	N_{cg}^{GO}	N_{cp}^{KOG}	N_{cg}^{KOG}
Ours (800)	659	9063	116	225	39	15
Ours (1700)	874	8153	162	314	75	18
PG	806	5175	33	60	41	15
GS	1166	4717	11	18	19	7

According to GO term results from Tables 2 and 3, our method predicts more specific GO terms (N_{cp}) than PG and GS in both PPI networks. Furthermore, the GO terms found by PG and GS are all covered by the modules detected by our method. Our method with larger k (fine-grained) has the best performance. However, the number of modules k does not affect the conclusions that our method dominates PG and GS. We observe the same trend for the evaluation based on KOG categories in Tables 2 and 3. Our method predicts more KOG annotated modules and covers more KOG categories than PG and GS especially when k is large (fine-grained) for both PPI networks. Our results are robust when we set k between the current selected values.

Fig. 2 shows the overall comparison results of the three blockmodel algorithms in terms of F-measures for both PPI networks. It is obvious that our method consistently outperforms PG and GS in both evaluation criteria based on GO terms and KOG categories.

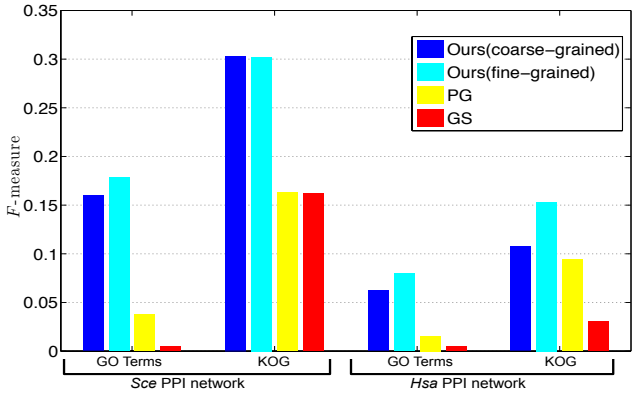

Fig. 2. Performance comparison for three blockmodel algorithms in terms of F-measures for both PPI networks.

For computational efficiency, our method depends on the speed of computing eigenvectors of the matrix W. On a standard desktop with 2.4GHz CPU and 6GB RAM, the computational time of our algorithm is competitive to PG and GS. Take the *Sce* PPI network for example, both PG and GS finish in around one minute and our algorithm takes around two minutes to get the results.

4.3. Module Examples

In this section, we illustrate several modules identified by our method. Fig. 3 illustrates two induced subnetworks from the *Sce* PPI network. All these modules in the subnetworks are statistically significantly enriched with corresponding GO terms as given in Table 4 computed by GOTermFinder [2]. As shown in Fig. 3, our method can detect both dense and sparse modules within which proteins have similar interaction patterns.

Fig. 4 illustrates two induced subnetworks from the *Hsa* PPI network. Again, the identified modules based on their interaction patterns share statistically significant functional similarity as annotated by GO terms as shown in Table 5.

5. CONCLUSIONS

In this paper, we propose a new approach to solve blockmodel module identification in PPI networks based on ran-

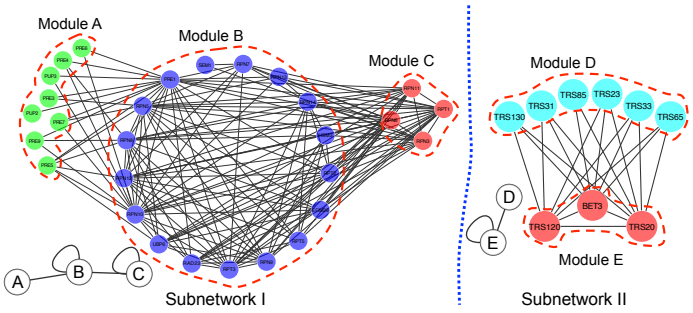

Fig. 3. Two subnetworks with modules detected by our method in the *Sce* network. Each node is annotated by its gene name.

Table 4. GO enrichment information of modules in Fig. 3

Modules	Enriched GO terms	GO level	p-value
A	proteasome core complex	[+3, -1]	6.42e-21
B	proteasome complex	[+3, -1]	4.30e-32
C	proteasome regulatory particle	[+3, -1]	3.81e-9
D	TRAPP complex	[+5, 0]	1.73e-16
E	TRAPP complex	[+5, 0]	1.00e-7

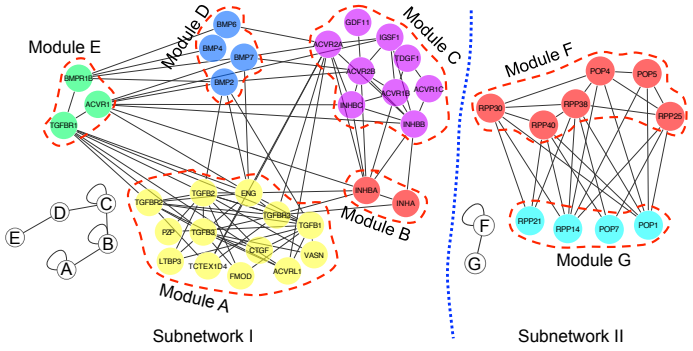

Fig. 4. Two subnetworks with modules detected by our method in the *Hsa* network. Each node is annotated by its gene name.

Table 5. GO enrichment information of modules in Fig. 4

Modules	Enriched GO terms	GO level	p-value
A	transforming growth factor beta	[+7, -1]	4.30e-7
B	hemoglobin biosynthetic process	[+5, 0]	6.51e-5
C	transmembrane receptor protein serine	[+6, -1]	9.03e-9
D	fibroblast growth factor	[+4, 0]	1.15e-7
E	transforming growth factor	[+7, -1]	7.06e-8
F	ribonuclease P activity	[+7, 0]	1.52e-18
G	ribonuclease P activity	[+7, 0]	2.86e-8

dom walk on graphs. The experimental results have proven that our method is superior to other existing state-of-the-art blockmodel algorithms. Furthermore, we have demonstrated that our method can identify biologically meaningful modules, specifically these with a sparse modular structure, which have important cellular functionalities.

6. REFERENCES

[1] M Ashburner et al. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nat Genet*, 25(1):25–29, 2000.

[2] E Boyle, I Elizabeth, S Weng, J Gollub, H Jin, D Botstein, J.M Cherry, and G Sherlock. GO::TermFinder—open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. *Bioinformatics*, 20:3710–3715, 2004.

[3] R. Guimera and M. Sales-Pardo. Missing and spurious interactions and the reconstruction of complex networks. *Proc Natl Acad Sci USA*, 106(52):22073–22078, 2009.

[4] J. King. Conductance and rapidly mixing markov chains. Technical report, University of Waterloo, 2003.

[5] S. Navlakha, M.C. Schatz, and C. Kingsford. Revealing biological modules via graph summarization. *J. Comp. Biol.*, 16(2):253–264, 2009.

[6] MEJ Newman and M Girvan. Finding and evaluating community structure in networks. *Phys Rev E*, 69:026113, 2004.

[7] Y. Park, C. Moore, and J. S. Bader. Dynamic networks from hierarchical bayesian graph clustering. *PLoS ONE*, 5(1):e8118, 2010.

[8] S Pinkert, J Schultz, and J Reichardt. Protein interaction networks: More than mere modules. *PLoS Comput Biol*, 6:e1000659, Jan 2010.

[9] TSK Prasad and et al. Human Protein Reference Database—2009 update. *Nucleic Acids Research*, 37:D767–D772, 2009.

[10] L. Royer, M. Reimann, et al. Unraveling protein networks with power graph analysis. *PLoS Comput Biol*, 4(7):e1000108, 2008.

[11] L Salwinski, CS Miller, AJ Smith, FK Pettit, JU Bowie JU, and D Eisenberg. The Database of Interacting Proteins: 2004 update. *Nucleic Acids Research*, 32:D449–D451, 2004.

[12] Y. Shih and S. Parthasarathy. Identifying functional modules in interaction networks through overlapping markov clustering. *bioinformatics*, 28:i473–i479, 2012.

[13] R. Tatusov et al. The cog database: an updated version includes eukaryotes. *BMC Bioinformatics*, 4:41, 2003.

[14] Y. Wang and X. Qian. Functional module identification by block modeling using simulated annealing with path relinking. In *ACM Conference on Bioinformatics, Computational Biology and Biomedicine 2012*, 2012.

[15] Y. Wang and X. Qian. A novel subgradient-based optimization algorithm for block model functional module identification. In *Asia Pacific Bioinformatics Conference 2013*, 2013.

[16] E. Xing and M. Jordan. On semidefinite relaxation for normalized k-cut and connections to spectral clustering. Technical report, UC. Berkeley, 2003.