BIOINFORMATICS ORIGINAL PAPER

Vol. 30 no. 1 2014, pages 81–93
doi:10.1093/bioinformatics/btt569

Systems biology

Advance Access publication October 1, 2013

Functional module identification in protein interaction networks by interaction patterns

Yijie Wang[1,*] and Xiaoning Qian[1,2,*]

[1]Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620, USA and [2]Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843, USA

Associate Editor: Martin Bishop

ABSTRACT

Motivation: Identifying functional modules in protein–protein interaction (PPI) networks may shed light on cellular functional organization and thereafter underlying cellular mechanisms. Many existing module identification algorithms aim to detect densely connected groups of proteins as potential modules. However, based on this simple topological criterion of 'higher than expected connectivity', those algorithms may miss biologically meaningful modules of functional significance, in which proteins have similar interaction patterns to other proteins in networks but may not be densely connected to each other. A few blockmodel module identification algorithms have been proposed to address the problem but the lack of global optimum guarantee and the prohibitive computational complexity have been the bottleneck of their applications in real-world large-scale PPI networks.

Results: In this article, we propose a novel optimization formulation LCP^2 (low two-hop conductance sets) using the concept of Markov random walk on graphs, which enables simultaneous identification of both dense and sparse modules based on protein interaction patterns in given networks through searching for LCP^2 by random walk. A spectral approximate algorithm SLCP^2 is derived to identify non-overlapping functional modules. Based on a bottom-up greedy strategy, we further extend LCP^2 to a new algorithm (greedy algorithm for LCP^2) GLCP^2 to identify overlapping functional modules. We compare SLCP^2 and GLCP^2 with a range of state-of-the-art algorithms on synthetic networks and real-world PPI networks. The performance evaluation based on several criteria with respect to protein complex prediction, high level Gene Ontology term prediction and especially sparse module detection, has demonstrated that our algorithms based on searching for LCP^2 outperform all other compared algorithms.

Availability and implementation: All data and code are available at http://www.cse.usf.edu/~xqian/fmi/slcp2hop/.

Contact: yijie@mail.usf.edu or xqian@ece.tamu.edu

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

Received on April 16, 2013; revised on September 23, 2013; accepted on September 25, 2013

1 INTRODUCTION

Advances in high-throughput profiling techniques have enabled researchers to produce large-scale protein–protein interaction (PPI) datasets (Phizicky and Fields, 1995; Raman, 2010). Systematic analysis of these large-scale interactomic datasets based on their graph representations, in which nodes stand for proteins in species of interest and edges represent interactions between proteins, has the potential to yield a better understanding of protein functions computationally (Rivas and Fontanillo, 2010). One way to chart out the underlying cellular functional organization is to identify functional modules in these networks by grouping the proteins sharing similar biological functions into the same modules (Navlakha *et al*., 2009; Nepusz *et al*., 2012; Pinkert *et al*., 2010; Royer *et al*.,2008).

Intuitively, based on interactomic data, if two proteins interact with each other, they are more likely to share the same cellular functionalities than proteins that do not interact. Thus, densely connected groups or subnetworks of proteins in a given network can be viewed as potential functional modules. Based on this idea, many modularity-based algorithms (Newman, 2006; Newman and Girvan, 2004) have been successfully applied to identify functional modules in PPI networks by detecting 'higher than expected connectivity' subnetworks. Several algorithms based on Markov random walk on graphs also have been proposed recently. For example, Markov CLustering (MCL) algorithm is one of such module identification algorithms for biological network analysis by iteratively implementing 'Expand' and 'Inflation' operations on the transition matrix of the underlying Markov chain of random walk on the given network (van Dongen, 2000). Regularized MCL (RMCL) (Satuluri and Parthasarathy, 2009; Satuluri *et al*., 2010) further extends the original MCL algorithm to penalize the large module size at each iteration to obtain more balanced modules with a similar number of nodes within them. Other formulations based on Markov random walk, including finding low conductance (LC) sets (Voevodski *et al*., 2009), also can be applied in module identification, which is similar to normalized cut problems (Xing and Jordan, 2003) in graph partitioning to minimize the normalized cut size across modules. Recently, several overlapping module identification methods have been developed to detect densely connected modules that may overlap with each other in networks. For example, Cluster One (ClusterOne) (Nepusz *et al*., 2012) can be viewed as the overlapping version of normalized cut. Link community (LinkComm) (Ahn *et al*., 2010) formulates the overlapping module identification in an innovative framework to implement the hierarchical clustering on edge graph representations, which reveals hierarchical and overlapping organization of networks.

In addition to densely connected modules in PPI networks, such as protein complexes, there are other topological structures

*To whom correspondence should be addressed.

in PPI networks that may possess important cellular functionalities. Again, based on interactomic data, the proteins that interact with similar sets of other proteins in a given network also intuitively have a higher probability of sharing the similar functionalities compared with the proteins that do not share any interacting partners or neighbors (Morrison *et al.*, 2006; Pinkert *et al.*, 2010; Royer *et al.*, 2008). These proteins may not directly interact with each other but they still work toward similar cellular functionalities and hence should belong to the same modules. Take transmembrane proteins for example. It is well known that transmembrane proteins, such as receptors in signal transduction cascades, tend to interact with cytoplasmic proteins as well as with extracellular ligands, but rarely interact with themselves (Pinkert *et al.*, 2010). To identify such types of functional modules, many state-of-the-art blockmodel module identification algorithms have been proposed recently. For example, Power Graph (PG) (Royer *et al.*, 2008) greedily collects topologically similar nodes into the same module based on Jaccard index similarity. Graph Summarization (GS) (Navlakha *et al.*, 2009, 2008) uses the minimum description length principle to group nodes with similar interaction patterns. However, both PG and GS are solved by greedy algorithms, which can not guarantee the global optimality. Additionally, they tend to over segment the network to get relatively small modules based on our empirical experience. A Bayesian framework (Hofman and Wiggins, 2008) based on a stochastic blockmodel formulation has been developed to identify modules as well as the optimal number of modules. However, the algorithm only guarantees to converge to local optima. Reichardt (2009) has proposed to solve blockmodel module identification by optimally mapping the given network to an image graph using simulated annealing, and several optimization strategies also have been proposed to accelerate the original simulated annealing algorithm (Wang and Qian, 2012, 2013). But those algorithms suffer from prohibitive computational complexity due to the inherent combinatorial complexity of the blockmodel problem.

In this article, we propose a novel formulation to solve the functional module identification problem, which simultaneously identifies the previously described dense and sparse modules with similar interaction patterns. The article is organized as follows: In Section 2, we first introduce the new optimization formulation by searching for the low two-hop conductance sets (LCP^2) based on the two-hop transition matrix of the underlying Markov chain of the random walk on a given network. Then, we derive the corresponding mathematical programing problem and propose an algorithm SLCP^2 (spectral algorithm for LCP^2), which solves LCP^2 to search for non-overlapping modules by a spectral approximate method with a close-to-optimal solution. We also present an extended algorithm (greedy algorithm for LCP^2) GLCP^2, which solves LCP^2 to search for overlapping modules by a bottom-up greedy strategy. In Section 3, we evaluate and compare our methods with other state-of-the-art algorithms for functional module identification on four large-scale PPI networks: the *Saccharomyces cerevisiae* PPI network extracted from the Database of Interacting Proteins (DIP) (Salwinski *et al.*, 2004) (*Sce*DIP); the corresponding network from the BioGRID database (Breitkreutz *et al.*, 2008; Stark *et al.*, 2006) (*Sce*BioGRID); the *Homo sapiens* (*Hsa*HPRD) PPI network collected from the Human Protein Reference Database (HPRD version 9) (Prasad *et al.*, 2009); and the human PPI network *Hsa*BioGRID obtained from BioGRID (Breitkreutz *et al.*, 2008; Stark *et al.*, 2006). The experimental results of protein complex prediction show that non-overlapping SLCP^2 outperforms most of the non-overlapping state-of-the-art algorithms and performs competitively with the more recent RMCL algorithm (Satuluri and Parthasarathy, 2009; Satuluri *et al.*, 2010). When we compare GLCP^2 with the other algorithms for overlapping modules, our experiments show that GLCP^2 outperforms ClusterOne (Nepusz *et al.*, 2012) and LinkComm (Ahn *et al.*, 2010). High level GO (Gene Ontology) term (Ashburner *et al.*, 2000) prediction results further demonstrate that SLCP^2 is superior to other non-overlapping algorithms, whereas GLCP^2 and LinkComm perform equally well. Furthermore, we present a few identified functional sparse modules to illustrate that SLCP^2 and GLCP^2 have the advantage in detecting functional sparse modules compared with the other state-of-the-art algorithms in the last part of Section 3. In Section 4, we draw our conclusions by briefly summarizing the differences between our new SLCP^2 and GLCP^2 algorithms and other existing module identification algorithms.

2 METHOD

2.1 Preliminaries

Without loss of generality, let $G = (V, E)$ represent a connected PPI network, in which there exists at least one path connecting any pair of nodes. Let V denote the set of nodes in G corresponding to n proteins and E is the edge set denoting interactions among all n proteins. A is the adjacency matrix of G, of which the element $A_{ij} = 1$ when node i 'interacts' with node j; $A_{ij} = 0$ when there is no interaction between node i and node j (there are no self edges and $A_{ii} = 0$).

For random walk on G, its underlying Markov chain can be characterized by a transition matrix $P = D^{-1}A$, where $D = Diag(d_1, d_2, \dots, d_n)$ is an $n \times n$ diagonal matrix with the corresponding node degrees ($d_i = \sum_j A_{ij}, i = 1, \dots, n$) on its diagonal. As G is connected, the underlying Markov chain of the random walk is irreducible and ergodic. Therefore there exists a stationary distribution satisfying $P^T\pi = \pi$, where $\pi_i = d_i/M, M = \sum_{i=1}^{n} d_i$. The conductance of a subset of nodes S in G can be defined as (King, 2003)

$$\Phi_P(S, \bar{S}) = \frac{\sum\limits_{i\in S, j\in \bar{S}} \pi_i P_{ij}}{\sum\limits_{i\in S} \pi_i}, S \cup \bar{S} = V. \qquad (1)$$

Finding k LC sets in the network G based on this conductance definition involves partitioning the node set V into k subsets $(S_1, S_2, \dots, S_k)$, which can be formulated as the following optimization problem:

$$\min \sum_{h=1}^{k} \Phi_P(S_h, \overline{S_h}) \qquad s.t. \bigcup_{h=1}^{k} S_h = V; S_h \cap S_l = \emptyset, \forall h \neq l. \qquad (2)$$

We call this method LC sets defined by P (LCP) for simplicity, and LCP is equivalent to the formulation of normalized k-cut in Xing and Jordan (2003).

2.2 Interaction patterns and transition matrix P^2

Considering Markov random walk on the given network G, its corresponding transition matrix P describes the transition probability that the random walker walks from one node to another in one step. With two directly interacting nodes ($A_{ij} = 1$), the corresponding transition probability is uniformly random among all the direct neighbors: $P_{ij} = \frac{A_{ij}}{d_i}$, denoting the probability of walking from node i to j in one step. Clearly, nodes without connections have no chance to reach each other in one step. The conductance definition in (1) extends to the transition probabilities between two complement partitions S and $\bar{S}$ in the given network. Hence, finding LCP tends to find densely connected modules as it aims to minimize the transition probabilities between potential modules to the rest of the network, which are dependent on the corresponding cut size or the number of edges across potential modules.

However, in addition to densely connected modules, functional module identification in PPI networks desires to detect other meaningful modules with nodes having similar interaction patterns in networks. The star and biclique motifs in Figure 1 show that nodes with similar interaction patterns may be sparsely connected or even have no interactions among them. For example, nodes marked by 'S' and 'T', which should be grouped into two respective modules, all have the same interaction patterns based on the network structure. But because there are no interactions among them, existing algorithms for densely connected modules, including LCP, rarely cluster them into the corresponding modules correctly. The second column in Figure 1 lists the random walk transition matrix P of each motif and the module dividing lines by LCP derived based on P. The third column in Figure 1 gives the objective function values computed by LCP (2). Based on the analysis of the three basic motifs, we confirm that LCP only focuses on detecting dense modules, which may not be adequate for functional module identification in PPI networks.

To identify modules of more diverse topology based on interaction patterns, we propose to search for LC sets defined by a two-hop transition matrix $P^2 = P \times P$ (LCP^2). Intuitively, nodes with similar interaction patterns (no matter whether densely connected or sparsely connected) are more likely to transit back to the nodes in the same module after two steps of random walk. Therefore, we redefine the conductance by replacing P with P^2, which captures more meaningful modular structures in PPI networks. The fourth and fifth columns in Figure 1 show P^2 transition matrices and module dividing lines for three basic motifs and LC values computed by P^2, respectively. From P^2 in Figure 1, we find that the nodes with the same interaction patterns have higher probabilities to walk to each other in two random walk steps. Therefore, the correct module identification of star and biclique motifs can be achieved by finding LC sets defined by the two-hop transition matrix P^2. For the clique motif, the nodes in cliques still have the same interaction patterns though the LC value computed by P^2 increases. Therefore, the corresponding cliques can still be correctly identified by LCP^2 as potential modules. The example of these three motifs demonstrates that dense modules like cliques and sparse modules such as stars and bicliques can be identified simultaneously through searching for LC sets based on P^2.

Based on these motivating examples, finding LC sets using P^2 has the promising potential to discover biologically meaningful modules consisting of the nodes with similar interaction patterns. We now provide the mathematical formulation and the optimization algorithm to solve LCP^2.

Similar to LCP, we aim to solve the following minimization problem LCP^2 by using the two-hop transition matrix P^2:

$$\min \sum_{h=1}^{k} \Phi_{P^2}(S_h, \overline{S_h}) \qquad s.t. \bigcup_{h=1}^{k} S_h = V;\ S_h \cap S_l = \emptyset, \forall\, h \neq l, \tag{3}$$

in which $\Phi_{P^2}(S_h, \overline{S_h})$ is the new conductance based on P^2. Note that P^2 is still a stochastic matrix and its stationary distribution is also π $(P^T P^T \pi = P^T \pi = \pi)$. We can derive that $\Phi_{P^2}(S, S) + \Phi_{P^2}(S, \bar{S}) = 1$ (proof is provided in Supplementary Materials). With these, the aforementioned problem (3) can be transformed to an equivalent formulation:

$$\max \sum_{h=1}^{k} \Phi_{P^2}(S_h, S_h) \qquad s.t. \bigcup_{h=1}^{k} S_h = V;\ S_h \cap S_l = \emptyset, \forall\, h \neq l. \tag{4}$$

As the underlying Markov chain is ergodic given a connected network, we have $\pi_i P_{ij} = \pi_j P_{ji} = A_{ij}/M$ and $\pi_i = d_i/M$. By expanding the objective function in (4), we can further derive

$$\begin{aligned}
&\sum_{h=1}^{k} \Phi_{P^2}(S_h, S_h)\\
&= \sum_{h=1}^{k} \frac{\sum_{i,j \in S_h} \pi_i P_{ij}^2}{\sum_{i \in S_h} \pi_i} = \sum_{i=1}^{k} \frac{\sum_{i,j \in S_h} \pi_i \sum_{l=1}^{n} P_{il} P_{lj}}{\sum_{i \in S_h} \pi_i}\\
&= \sum_{h=1}^{k} \frac{\sum_{i,j \in S_h} \sum_{l=1}^{n} A_{il} P_{lj}}{\sum_{i \in S_h} d_i} = \sum_{h=1}^{k} \frac{x_h^T A P x_h}{x_h^T D x_h}\\
&= \sum_{h=1}^{k} \frac{x_h^T A D^{-1} A x_h}{x_h^T D x_h} = trace\left(\frac{X^T A D^{-1} A X}{X^T D X}\right),
\end{aligned}$$

where x_h denotes the hth column of the $n \times k$ module assignment matrix X, which lies in the space:

$$\mathfrak{F}_k = \{X : X 1_k = 1_n,\ x_{ij} \in \{0, 1\}\}, \tag{5}$$

in which 1_k and 1_n are vectors with all of their elements equal to 1.

Combining the transformed objective function and the constraint set (5), we can express LCP^2 as the following optimization problem:

$$(F)\begin{cases} \max: & trace\left(\frac{X^T A D^{-1} A X}{X^T D X}\right) \\ s.t. & X \in \mathfrak{F}_k \end{cases} \tag{6}$$

2.3 Module identification by interaction patterns

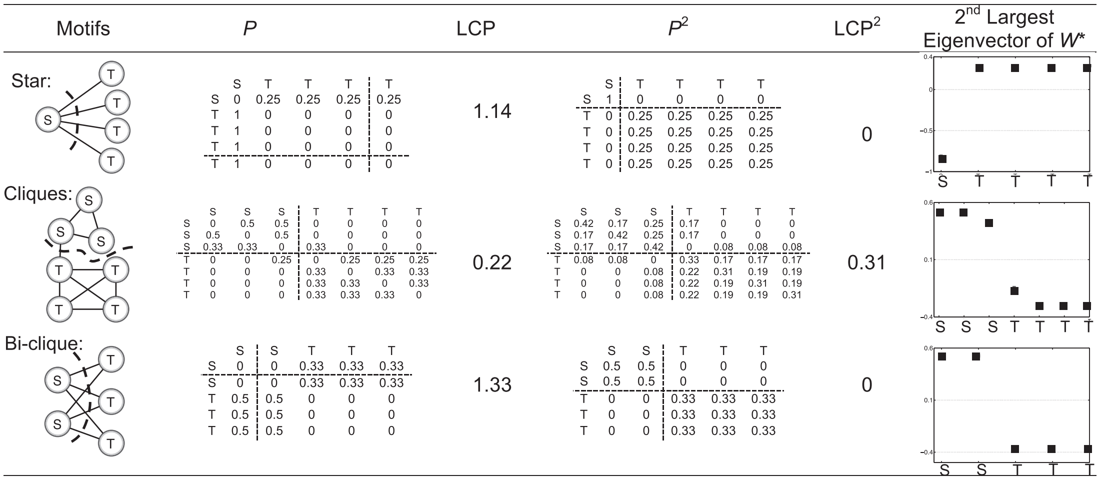
2.3.1 Non-overlapping algorithm We can further transform the problem (F) to the following relaxed optimization problem (detailed explanation is given in Supplementary Materials):

$$(F1)\begin{cases} \max & trace(Y^T W Y) \\ s.t. & Y^T Y = I_k, \end{cases} \tag{7}$$

where $W = D^{-1/2} A D^{-1} A D^{-1/2}$, and $Y = D^{1/2} X (X^T D X)^{-1/2}$ denotes the relaxed assignment matrix, which is orthonormal. Let $H = D^{-1/2} A D^{-1/2}$. We can rewrite $W = HH^T$ as the inner-product of H. Taking each column of H as the normalized interaction pattern of the corresponding node, this Gram matrix W measures the interaction similarity among different nodes (we note that the inner-product can be replaced by a general Mercer kernel if needed). According to this inner-product form of W, nodes in dense modules have high similarities as they share the same interaction pattern, which is to interact with each other within modules. At the same time, similarities among nodes in sparse modules are high because they interact with similar neighbors in the rest of the network. Consequently, similarities among nodes with similar interaction patterns (no matter whether in dense or sparse modules) are higher. Therefore, nodes that play identical roles in the network can be grouped together.

We note that a formulation similar to ours has also been independently presented in Satuluri and Parthasarathy (2011). The authors in Satuluri and Parthasarathy (2011) have proposed to use a symmetrization strategy AA^T to detect interaction patterns of nodes. In our new LCP^2 formulation, module identification depends on the different form HH^T, which can be viewed as the normalized version of AA^T. As shown in previous results obtained by normalized cuts, we expect that this new formulation depending on the normalized version HH^T may yield more balanced modules that may lead to biologically meaningful functional module identification results.

To derive the solution strategy for LCP^2, we relax Y to be an orthonormal matrix and it turns out that $(F1)$ has a closed-form solution based on *Ky Fan Theorem.*



	S	T	T	T	T
S	0	0.25	0.25	0.25	0.25
T	1	0	0	0	0
T	1	0	0	0	0
T	1	0	0	0	0
T	1	0	0	0	0

	S	T	T	T	T
S	1	0	0	0	0
T	0	0.25	0.25	0.25	0.25
T	0	0.25	0.25	0.25	0.25
T	0	0.25	0.25	0.25	0.25
T	0	0.25	0.25	0.25	0.25

	S	S	S	T	T	T	T
S	0	0.5	0.5	0	0	0	0
S	0.5	0	0.5	0	0	0	0
S	0.33	0.33	0	0.33	0	0	0
T	0	0	0.25	0	0.25	0.25	0.25
T	0	0	0	0.33	0	0.33	0.33
T	0	0	0	0.33	0.33	0	0.33
T	0	0	0	0.33	0.33	0.33	0

	S	S	S	T	T	T	T
S	0.42	0.17	0.25	0.17	0	0	0
S	0.17	0.42	0.25	0.17	0	0	0
S	0.17	0.17	0.42	0	0.08	0.08	0.08
T	0.08	0.08	0	0.33	0.17	0.17	0.17
T	0	0	0.08	0.22	0.31	0.19	0.19
T	0	0	0.08	0.22	0.19	0.31	0.19
T	0	0	0.08	0.22	0.19	0.19	0.31

	S	S	T	T	T
S	0	0	0.33	0.33	0.33
S	0	0	0.33	0.33	0.33
T	0.5	0.5	0	0	0
T	0.5	0.5	0	0	0
T	0.5	0.5	0	0	0

	S	S	T	T	T
S	0.5	0.5	0	0	0
S	0.5	0.5	0	0	0
T	0	0	0.33	0.33	0.33
T	0	0	0.33	0.33	0.33
T	0	0	0.33	0.33	0.33

Fig. 1. Different module identification results obtained by using P and P^2. The first column displays three basic motifs (star motif, clique motif and biclique motif) [used by Royer *et al.* (2008)] and the dashed lines show the natural partitions. The second column gives the P of three basic motifs and the dashed lines denote the module dividing lines obtained by LCP. The third column gives the minimum objective function values by (2). The fourth column gives the P^2 of three basic motifs and the dashed lines indicate the identified modules by LCP^2. The fifth column shows the minimum objective function values based on (3). The last column illustrates the second largest eigenvector of W^* used in Algorithm 1

Theorem 1. *(Ky Fan Theorem) Let T be a symmetric matrix with eigenvalues* $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ *and the corresponding eigenvectors* $U = [u_1, \ldots, u_n]$. *Then* $\sum_{i=1}^{k} \lambda_i = \max_{X^TX=I_k} trace(X^TTX)$. *Moreover, the optimal* X^* *is given by* $X^* = [u_1, \ldots, u_k]Q$ *with Q being an arbitrary orthogonal matrix.*

Following this theorem, we can use the largest k eigenvectors of the Gram matrix W to approximate the module assignment matrix Y. Therefore, we propose our module identification algorithm SLCP^2 in Algorithm 1.

```
Algorithm 1 (Non-overlapping): Spectral algorithm for LCP$^2$ (SLCP$^2$)
Input: Adjacency matrix $A$ and the number of modules $k$
Output: Module assignment matrix $X_n$
1. Add self loops to $A = A + I_n$
2. Compute $W = D^{-1/2}AD^{-1}AD^{-1/2}$
3. $W^* = W - Diag(diag(W))$
4. Find the largest $k$ eigenvalues and their corresponding eigenvectors
   $[E, V_k] = eig(W^*, k)$
5. Obtain the approximated module assignment $R$ by pivoted QR decomposition:
   $V_k^T P = Q[R11,\ R12]$, then $R = [I_k\ R_{11}^{-1}R_{12}]P^T$
6. The module membership of each node is determined by the row index
   of the largest element in the absolute values of the corresponding column of $R$
```

The first step in the algorithm aims to compute the interaction similarity more accurately by considering the self connection. Adding self loops can make dense modules more distinguishable and avoid impairing the dense modular structure by considering interaction patterns. The second step computes W. The third step removes the diagonal part of the Gram matrix W to get rid of the influence of self similarity because proteins tend to be clustered into single node modules when they have large self similarities. To obtain modules of appropriate size, removing self similarity is necessary. The fourth step obtains the k largest eigenvectors of W^*. Steps 5 and 6 use the pivoted QR decomposition to approximate the module assignment matrix X (Zha *et al.*, 2001). The pivoted QR decomposition is a better option than the classic k-means method. It is well known that the performance of k-means heavily depends on its initialization. However, when dealing with a large-scale network that may have thousands of potential modules, it is difficult for k-means to find good initializations. Using the pivoted QR decomposition avoids the initialization step, therefore better performance can be achieved. As illustrated, the last column of Figure 1 exhibits the second largest eigenvector of W^*, from which we can easily distinguish the two different modules in the three motifs in Figure 1.

2.3.2 *Overlapping algorithm*

Based on the previously derived Gram matrix W, which contains the information of interaction similarity among all the nodes in the given network, we can further derive a bottom-up greedy algorithm to identify overlapping functional modules. The procedure of the greedy algorithm is illustrated in Algorithm 2. The idea of adopting the greedy strategy is similar to the one used in ClusterOne (Nepusz *et al.*, 2012) to grow each module from each single protein as a seed. For each iteration, we add proteins to modules to acquire the most gain in the weight density of a module h, which can be computed as

$$W_d(S_h) = \frac{\sum_{i,j \in S_h} W^*_{i,j}}{|S_h|^2}, \qquad (8)$$

where $W_{i,j}$ measures the interaction similarity between proteins i and j. We keep adding proteins to potential modules until there is no increase of the weight density.

```
Algorithm 2 (Overlapping): Greedy algorithm for LCP$^2$ (GLCP$^2$)
Input: Gram matrix $W$
Output: Module assignment matrix $X_o$
1. Assign each protein in its own module
2. Compute the average weight density $Q = \frac{\sum W_d(S_h)}{N}$
3. while($Q > \xi$)
4.     Shuffle protein list $V$
5.     for $i = 1 : |V|$
6.         Add the protein $V_i$ to existing module $h$ to achieve the largest
7.         positive weight density gain.
8.     endfor
9.     Re-compute the average weight density $Q$.
10. endwhile
11. Post-processing the obtained modules.
```

The post-processing step in Line 11 of Algorithm 2 aims to remove low-quality modules and merge highly overlapped modules. Because our

LCP^2 formulation can detect both densely connected modules and sparsely connected modules (the sparsely connected modules contain proteins with similar interaction to the rest of the network), we use two quality functions to evaluate the obtained modules. One quality function is $qf_d = \text{edgedensity} \times \text{sqrt(size)}$, which has been similarly adopted in Shih and Parthasarathy (2012) to identify high-quality dense modules. The other quality function is $qf_s = \#.\text{sharedproteins}/\text{size}$ for sparse modules. We remove the modules when $qf_d < \alpha$ and $qf_s < \beta$, where α and β are two user-specified thresholds. With larger α and β, we may remove a larger number of low-quality modules by qf_d and qf_s. After removing low-quality modules, we merge highly overlapped modules based on $NA(a, b) = \frac{|V_a \cap V_b|^2}{|V_a| \times |V_b|}$, where *a, b* are two modules with V_a and V_b proteins respectively. If $NA(S_i, S_j) > p$, we merge modules S_i and S_j together. Here, p is another tuning parameter and we typically set it over 0.9 to guarantee that only highly overlapped modules are merged.

3 EXPERIMENTS

We first introduce how we implement the algorithms that we take for performance comparison, where we obtain the PPI networks and protein complex golden standard sets, and what criteria we use to evaluate the performance of the selected algorithms. After that, we compare all algorithms on synthetic networks with both dense and sparse modular structures and show that both the non-overlapping and overlapping algorithms (SLCP^2 and GLCP^2, respectively) based on the two-hop transition matrix outperform all other state-of-the-art methods. Then, we analyze the performance of protein complex and high level GO term predictions to demonstrate the potential of predicting biologically meaningful modules by all compared algorithms. In the end, we illustrate that the algorithms based on our LCP^2 formulation are superior to the state-of-the-art algorithms in identifying sparse functional modules by displaying the module detection results for several specific biological functional sparse modules.

3.1 Algorithms, data and metric

3.1.1 Algorithms For algorithms that identify non-overlapping modules, we compare SLCP^2 with five state-of-the-art algorithms, which are LCP (Xing and Jordan, 2003), MCL (van Dongen, 2000), RMCL (Satuluri and Parthasarathy, 2009; Satuluri *et al.*, 2010), GS (Navlakha *et al.*, 2009) and PG (Royer *et al.*, 2008). Comparing with LCP aims to show that finding LC sets through P^2 is superior to LCP based on the conductance definition by P, as LCP only focuses on detecting dense modules. We also compare SLCP^2 with MCL and RMCL because they are widely used network clustering algorithms in biological network analysis and have been shown to give biologically meaningful results. Additionally, two other algorithms, GS and PG, are chosen as they search for modules based on interaction patterns and hence are also able to detect both dense and sparse modules as SLCP^2 does.

For overlapping module identification algorithms, we compare our GLCP^2 with two other recently proposed algorithms: ClusterOne (Nepusz *et al.*, 2012) and LinkComm (Link Community) (Ahn *et al.*, 2010). To distinguish non-overlapping and overlapping algorithms, we mark all the overlapping algorithms with a star (*) in all the figures in our experimental results.

As discussed earlier, LCP is equivalent to the normalized k-cut problem (Xing and Jordan, 2003). Therefore, we adopt the spectral method proposed in Xing and Jordan (2003) to solve LCP. The implementation of the k-means clustering algorithm used by LCP is based on the procedure proposed in Bisgin and Dalfes (2008). We have obtained the source code for MCL (http://www.micans.org/mcl), RMCL (http://www.cse.ohio-state.edu/satuluri/research.html), GS(https://open-innovation.alcatel-lucent.com/projects/gscode/), PG(http://www.biotec.tu-dresden.de/re-search/schroeder/powergraphs/), ClusterOne(http://www.paccanarolab.org/cluster-one/index.html) and LinkComm (https://github.com/bagrow/linkcom) from the web pages provided in the corresponding articles.

For non-overlapping module identification algorithms, SLCP^2 and LCP have one parameter k (the number of modules) and MCL also has one tuning parameter called 'Inflation' I_F. RMCL has two tuning parameters, which are 'balance' b and 'Inflation' I_F. For the number of modules k in SLCP^2 and LCP, we implement the grid search from $k = 500$–3000 with an interval of 100. For I_F in MCL, we similarly search from 1.2 to 5.0 with an interval of 0.1. For RMCL, we set b and I_F to 0.5 and 2.0, respectively, based on the suggestions in the articles (Satuluri *et al.*, 2010; Shih and Parthasarathy, 2012). Because both PG and GS are hierarchical bottom-up algorithms, they do not have any tuning parameter.

For overlapping module identification algorithms, LinkComm has one parameter t and GLCP^2 has three parameters α, β and p. For LinkComm, we set the threshold $t = 0.2$ as it yields the best results in our experiments. For GLCP^2, the parameters set (α, β, p) determines the quality of the results. From our experience, $(\beta, p) = (0.8, 0.9)$ gives good performance. As to α, it depends on the density of the original network. In the following experiments, we set $\alpha = 0.76$ for the *Sce* PPI networks and $\alpha = 0.7$ for the *Hsa* PPI networks because the *Hsa* PPI networks are more sparse than the *Sce* PPI networks.

3.1.2 Data We have run all these selected algorithms on four PPI networks. Two of them are *S.cerevisiae* (Sce) PPI networks obtained from the DIP (Salwinski *et al.*, 2004) (SceDIP) and BioGRID database (Breitkreutz *et al.*, 2008; Stark *et al.*, 2006) (SceBioGRID), respectively. The other two are the *H.sapiens* (Hsa) PPI networks extracted from HPRD (Prasad *et al.*, 2009) (HsaHPRD) and BioGRID database (Breitkreutz *et al.*, 2008; Stark *et al.*, 2006) (HsaBioGRID), respectively. We use the largest components of these four networks as the input of the algorithms.

We evaluate the complex prediction performance of the algorithms based on four protein complex golden standards. For *Sce* PPI networks, we use Munich Information Center for Protein Sequences (MIPS) (Mewes *et al.*, 2004) and Saccharomyces Genome Database (SGD) (Hong *et al.*, 2008) golden standards. For *Hsa* PPI networks, we adopt the Human Protein Complex Database with a Complex Quality Index (PCDq) (Kikugawa *et al.*, 2012) as well as CORUM (comprehensive resource of mammalian protein complexes) golden standards (Ruepp *et al.*, 2008) for our performance evaluation. We use all golden standard protein complexes with two or more proteins in all our experiments.

For examining whether the detected modules capture protein functional relationships other than just protein complexes, we use the high-level GO terms in all three domains (molecular function (F), biological process (P) and cellular component

(C)) as the golden standard for GO term prediction. Any GO term, whose information content (IC) (Shih and Parthasarathy, 2012) is >2, is considered as a high-level GO term. The definition of the IC of a GO term g is $\mathrm{IC} = -\log(|g|/|root|)$ as given in Shih and Parthasarathy (2012), where '*root*' is the corresponding root GO term (either F, P or C) of g. In addition, we remove GO terms that contain <2 proteins. The detailed information of the networks, complex golden standards and GO terms are listed in Table 1.

3.1.3 Metric To evaluate the performance for complex prediction, we use two independent quality measures [used by Nepusz *et al.* (2012)] to assess the similarity between the predicted complexes and the golden standard reference complexes. In our experiments, we set the minimum size of detected modules to three for fair comparison between all competing algorithms. The first measure counts the number of predicted modules matched to the golden standard reference modules. A predicted module a with V_a proteins is considered a match to a reference module b with V_b proteins when the neighborhood affinity $NA(a,b) = \frac{|V_a \cap V_b|^2}{|V_a| \times |V_b|} \geq 0.25$ (Li *et al.*, 2010; Nepusz *et al.*, 2012). The threshold of 0.25 is chosen because it represents the case when at least half of the complexes overlap if the two compared complexes are equally large. The second measure is the geometric mean of two other measures, which is the cluster-wise sensitivity (Sn) and the cluster-wise positive predictive value (PPV) (Li *et al.*, 2010). Given r predicted and s reference complexes, let t_{ij} denote the number of proteins that exist in both predicted complex i and reference complex j, and w_j represent the number of proteins in reference complex j. Then Sn and PPV can be defined as

$$Sn = \frac{\sum_{j=1}^{s} \max_{i=1,\ldots,r} t_{ij}}{\sum_{j=1}^{s} w_j}, \quad PPV = \frac{\sum_{i=1}^{r} \max_{j=1,\ldots,s} t_{ij}}{\sum_{i=1}^{r}\sum_{j=1}^{s} t_{ij}}.$$

Because Sn can reach its maximum by grouping all proteins in one module, whereas PPV can be maximized by putting each protein in its own module, we use their geometric mean as 'accuracy' to balance these two measures ($Acc = \sqrt{Sn \times PPV}$) (Li *et al.*, 2010; Nepusz *et al.*, 2012).

To investigate the functional significance of identified modules, we follow the same strategy in Shih and Parthasarathy (2012) to compute F measure based on high-level GO term prediction.

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

Network	Number of nodes	Number of edges	MIPS	SGD	PCDq	CORUM	$\|GO\|$
*Sce*DIP	4980	22 076	203	305	—	—	1166
*Sce*BioGRID	5640	59 748	203	305	—	—	1172
*Hsa*HPRD	9269	36 917	—	—	1204	1294	4452
*Hsa*BioGRID	14 283	87 397	—	—	1204	1294	4457

Note: The networks are the largest components of the original datasets. $|GO|$ is the number of GO terms whose IC is >2.

Let $C = \{c_1, c_2, \ldots, c_k\}$ denote the identified modules and $G = \{g_1, g_2, \ldots, g_l\}$ denote the selected GO terms. We can calculate the number of identified modules that match at least one GO term denoted by N_{cp}: $N_{cp} = |\{c_i \in C | NA(c_i, g_j) > 0.25, \exists g_j \in G\}|$. The number of GO terms that match at least one identified module can be computed: $N_{cg} = |\{g_i \in G | NA(c_i, g_j) > 0.25, \exists c_i \in C\}|$. Based on these numbers, we can further compute precision and recall: $\text{precision} = \frac{N_{cp}}{|C|}$, $\text{recall} = \frac{N_{cg}}{|G|}$. The final F-measure is the harmonic mean of precision and recall: $F = 2 \times \text{precision} \times \text{recall}/(\text{precision} + \text{recall})$.

Finally, all experiments illustrated in this article can be accomplished within 1 h on a 2.4 GHz quad-core CPU and 6 GB RAM computer. Except when identifying modules in the *Hsa*BioGRID PPI network, PG and SG fail to execute due to the large memory requirement from two algorithms for this large PPI network. Based on the simulation results, the run time of SLCP^2 and GLCP^2 is competitive with the other algorithms. For example, SLCP^2 only takes around 2 min for clustering the *Sce*DIP PPI network into $k = 1000$ modules and GLCP^2 needs <1 min for analyzing the *Sce*DIP PPI network.

3.2 Synthetic networks

To illustrate the performance difference of different algorithms, we first evaluate all the selected algorithms on synthetic networks with the known ground truth. The modular structure of synthetic networks is shown in Figure 2A. There are three dense modules of different sizes together with two sparse modules of the same size. To test statistical significance, we generate the null model by shuffling edges from an original synthetic network based on the Maslov–Sneppen procedure (Maslov and Sneppen, 2002). Figure 2B is one example of the random network after half of the original edges are permuted. The performance is evaluated by generalized normalized mutual information (GNMI) (Lancichinetti *et al.*, 2009) for both non-overlapping and overlapping module identification algorithms. GNMI ranges from 0 to 1 and it equals to 1 when the module identification result is the same as the ground truth.

Figure 2C shows the mean values and the standard deviations of GNMI obtained by all the algorithms on 100 random null networks. For non-overlapping algorithms, SLCP^2 is superior to LCP, MCL and RMCL. For PG and GS, although the obtained GNMI values are better than LCP and MCL, they may not provide useful biological information as their identified modules are fine grained (one or two nodes in each module). For overlapping module identification algorithms, GLCP^2 outperforms ClusterOne and LinkComm. Figure 2D plots the $-log(p - value)$ of the t-test scores of SLCP^2 compared with other non-overlapping algorithms as well as the comparison of GLCP^2 to ClusterOne and LinkComm. From Figure 2D, we find that both SLCP^2 and GLCP^2 are significantly better than other state-of-the-art algorithms on synthetic networks with the ground truth modular structure.

In addition, we estimate the statistical significance for each identified module in synthetic random networks for all nine algorithms. We annotate the dense modules in Figure 2A as D1, D2 and D3 and sparse modules as S1 and S2. Based on 100 random null networks, for each module, we can obtain the distribution of corresponding Acc scores based on the known

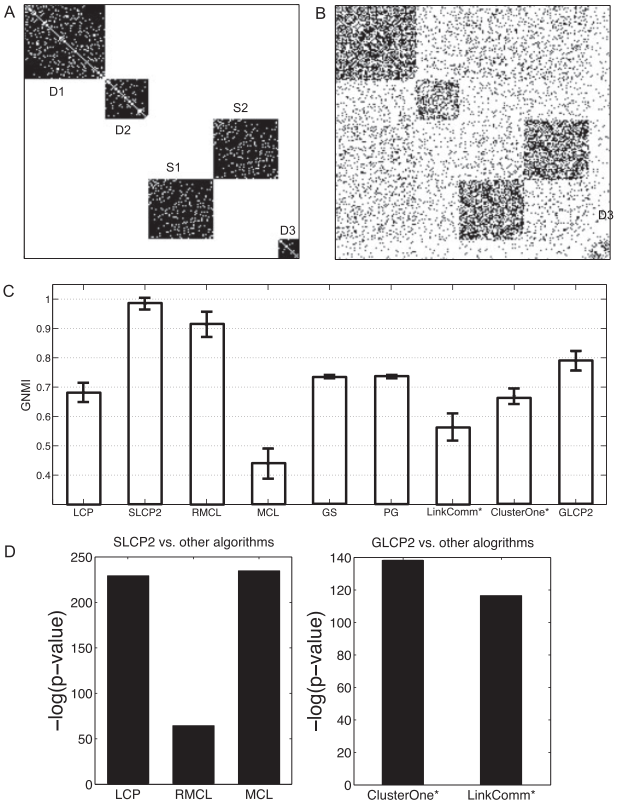


Fig. 2. Performance comparison on synthetic networks: (**A**) the adjacency matrix of the original network; (**B**) one example of the randomly shuffled network (obtained by shuffling half of the original edges); (**C**) GNMI comparison among all algorithms; (**D**) *t*-test results

ground truth. Figure 3A displays the mean values and the standard deviations of *Acc* scores produced by all the algorithms on every module in Figure 2A. For example, the first nine bars indicate the mean values and the standard deviations of *Acc* scores from all nine competing algorithms in detecting dense module D1 in Figure 2A. Based on the distributions of *Acc* scores, we can further compute the *P*-values of our proposed algorithms compared with other state-of-the-art algorithms. Figure 3B plots the $-log(p-value)$ of the *t*-test scores of SLCP^2 compared with other non-overlapping algorithms and the comparison of GLCP^2 to ClusterOne and LinkComm on all five modules, respectively. We consider our algorithms are significantly better when $-log(p-value) \geq 3$ $(p-value \leq 1.0e-3)$. From Figure 3B, we find that LCP and SLCP^2 are competitive in identifying dense module D1. For the rest of the modules and algorithms, the $-log(p-value)$ values shown in Figure 3B imply that our SLCP^2 and GLCP^2 achieve significantly better performance in detecting both dense and sparse modules. Furthermore, from Figure 3B, we find the bars for sparse modules (S1 and S2) are typically higher than those corresponding to dense modules, which further validates that the competing algorithms focus more on detecting dense modules, whereas our proposed algorithms can simultaneously detect both dense and sparse modules based on interaction patterns.

3.3 Complex prediction

We test the quality of a module identification algorithm by how well it can be applied to make predictions for protein complexes. We compare SLCP^2 with other state-of-the-art non-overlapping module identification algorithms, including LCP, RMCL, MCL, GS and PG, on four PPI networks. Also, to detect overlapping modules, we compare GLCP^2 with ClusterOne and LinkComm. The information of the module identification results and the optimal parameters used by each algorithm are reported in Tables 2 and 3.

For non-overlapping module identification algorithms, as shown in Tables 2 and 3, SLCP^2 and RMCL are competitive and outperform all the other non-overlapping algorithms. For the *Sce*DIP PPI network, SLCP^2 achieves better performance than RMCL because it predicts more matched protein complexes and has a higher *Acc* score. For other PPI networks, SLCP^2 and RMCL obtain competitive results, as SLCP^2 consistently predicts more matched protein complexes, whereas RMCL gets higher *Acc* scores. In addition, SLCP^2 has the best coverage with more proteins clustered into corresponding modules on all four PPI networks except the *Sce*BioGrid PPI network. For the *Sce*BioGrid PPI network, RMCL only covers one more protein than SLCP^2.

For overlapping module identification algorithms, based on Tables 2 and 3, we find that GLCP^2 outperforms LinkComm and ClusterOne. Although both GLCP^2 and LinkComm identify competitive numbers of protein complexes in different golden standards, GLCP^2 consistently achieves higher *Acc* scores for all four PPI networks. Finally, GLCP^2 also has the best coverage on all four PPI networks except the *Sce*BioGrid PPI network, on which LinkComm has a higher coverage than GLCP^2. If we consider that LinkComm identifies larger numbers of smaller overlapping modules as shown in both tables, we expect that GLCP^2 may provide more biologically meaningful results.

Furthermore, we test the statistical significance of our algorithms in terms of predicting the SGD golden standard complexes on the *Sce*DIP PPI network. We first generate 100 random networks from the original *Sce*DIP PPI network by randomly shuffling the original edges based on the Maslov–Sneppen procedure (Maslov and Sneppen, 2002). Then, we obtain the empirical distributions of *Acc* scores with respect to the prediction of the SGD golden standard on these 100 randomized networks for the competing algorithms. Based on the results provided in Table 2, we compare SLCP^2 with RMCL for non-overlapping algorithms and GLCP^2 with LinkComm for overlapping algorithms because they are the two best-performing algorithms in predicting the SGD complexes among non-overlapping algorithms and overlapping algorithms, respectively. For non-overlapping algorithms, the average and the standard deviation of *Acc* scores obtained by SLCP^2 are 0.5180 and 0.0064. Although for RMCL, the average and the standard deviation of the *Acc* scores are 0.5137 and 0.0044. For overlapping algorithms, the average and the standard deviation of *Acc* scores of GLCP^2 are 0.5018 and 0.0054. For LinkComm, the average and the standard deviation of the *Acc* scores are 0.4983 and 0.0047. By two-sample *t*-test, SLCP^2 is significantly better than RMCL with the *P*-value 1.429e-08 and GLCP^2 is significantly better than LinkComm with the *P*-value 1.095e-07.

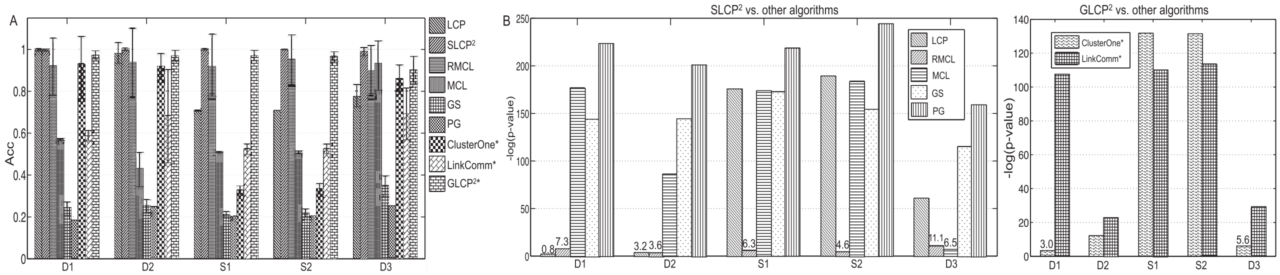


Fig. 3. Statistical performance in detecting each module: (**A**) *Acc* scores comparison among all algorithms; (**B**) *t*-test results based on distributions of *Acc*. For low bars in (B), we put the $-log(P - value)$ values on top of the bars

Table 2. Performance comparison for complex prediction on *Sce* PPI networks

Network	Golden standard	Method	Coverage	Number of clusters(size ≥3)	Average size	Number of matched	*Sn*	*PPV*	*Acc*
SceDIP	MIPS	LCP($k = 1000$)	2572	525	4.90	62	0.2346	0.3825	0.2995
		RMCL	3725	814	4.57	79	0.2834	0.3977	0.3357
		MCL ($I_F = 2.2$)	3846	675	5.70	68	0.2821	0.3787	0.3269
		GS	2391	550	4.35	65	0.185	**0.4067**	0.2743
		PG	2717	364	7.46	14	0.1153	0.2978	0.1853
		SLCP^2($k = 1000$)	4564	783	5.83	84	0.3050	0.3732	0.3378
		ClusterOne[a]	1461	358	4.89	81	0.2641	0.3605	0.3085
		LinkComm[a]	2344	1725	4.12	102	**0.3093**	0.3575	0.3326
		GLCP^2[a]	3447	1517	4.46	**104**	0.3066	0.3928	**0.3470**
	SGD	LCP($k = 1000$)	2572	525	4.90	75	0.3484	0.6058	0.4594
		RMCL	3725	814	4.57	125	0.4572	0.6039	0.5254
		MCL (I_F=2.3)	3630	659	5.70	115	0.4468	0.5735	0.5102
		GS	2391	550	4.35	88	0.2915	**0.6689**	0.4416
		PG	2717	364	7.46	11	0.1714	0.4102	0.2615
		SLCP^2($k = 1000$)	4564	783	5.83	125	**0.4917**	0.5621	0.5257
		ClusterOne[a]	1461	358	4.89	113	0.4037	0.5775	0.4828
		LinkComm[a]	2344	1725	4.12	136	0.4567	0.4895	0.4727
		GLCP^2[a]	3447	1517	4.46	**155**	0.4894	0.5850	**0.5350**
SceBioGRID	MIPS	LCP($k = 1000$)	3503	557	6.30	77	0.2978	0.4252	0.3558
		RMCL	5210	772	4.57	81	0.4908	0.3921	0.4346
		MCL ($I_F = 3.3$)	3544	338	10.49	45	.3495	0.3270	0.3380
		GS	3315	609	5.44	83	0.2420	**0.4296**	0.3224
		PG	2601	356	7.31	2	0.0740	0.3128	0.1521
		SLCP^2($k = 1000$)	5209	782	6.76	84	0.3723	0.3906	0.3810
		ClusterOne[a]	2580	473	7.57	101	0.4797	0.3938	0.4346
		LinkComm[a]	4633	4108	5.48	**143**	**0.5891**	0.3526	0.4557
		GLCP^2[a]	4440	2183	7.74	136	0.5006	0.4204	**0.4587**
	SGD	LCP($k = 1000$)	3503	556	6.30	98	0.4672	0.6236	0.5398
		RMCL	5210	772	4.57	137	0.6628	0.5915	0.6262
		MCL ($I_F = 3.2$)	3652	335	10.49	80	0.4291	0.4752	0.4516
		GS	3315	609	5.44	130	0.3774	**0.6544**	0.4969
		PG	2601	356	7.31	3	0.135	0.4517	0.2469
		SLCP^2($k = 1000$)	5209	782	6.76	151	0.5847	0.5926	0.5886
		ClusterOne[a]	2580	473	7.57	158	0.6703	0.5621	0.6138
		LinkComm[a]	4633	4108	5.48	**207**	**0.7955**	0.4637	0.6037
		GLCP^2[a]	4440	2183	7.74	204	0.7341	0.5887	**0.6574**

Note: Bold values denote the best scores corresponding to specific criteria.
[a]Overlapping module identification algorithms.

In summary, both SLCP^2 and GLCP^2 based on our new optimization formulation LCP^2 using the concept of random walk on graphs are among the best performing algorithms for protein complex prediction. However, protein complexes have typical dense modular structure within which proteins are highly connected. As our SLCP^2 and GLCP^2 aim to detect both dense and sparse modules, these protein complex prediction results only exhibit one aspect of our algorithms' performance. In the

Table 3. Performance comparison for complex prediction on *Hsa* PPI networks

Network	Golden standard	Method	Coverage	Number of clusters (size ≥3)	Average size	Number of matched	*Sn*	*PPV*	*Acc*
HsaHPRD	PCDq	LCP($k = 1000$)	8561	979	8.74	205	0.3986	0.4206	0.4095
		RMCL	6879	1508	4.56	290	0.3538	0.5990	0.4604
		MCL ($I_F = 3.3$)	6534	1279	5.11	237	0.3255	0.5633	0.4282
		GS	4719	1167	4.04	167	0.2169	**0.6785**	0.3836
		PG	5172	805	6.25	22	0.2016	0.3453	0.2639
		SLCP^2($k = 1000$)	8657	1494	5.79	303	0.3916	0.4774	0.4324
		ClusterOne[a]	2915	771	4.53	199	0.2379	0.6478	0.3925
		LinkComm[a]	7183	4107	4.58	418	**0.4314**	0.3029	0.3652
		GLCP^2[a]	8181	4257	4.50	**450**	0.4145	0.5377	**0.4721**
	CORUM	LCP($k = 1000$)	8561	979	8.74	172	0.3729	0.2049	0.2764
		RMCL	6879	1508	4.56	247	0.3291	0.2777	0.3023
		MCL ($I_F = 3.3$)	6534	1279	5.11	215	0.3192	0.2567	0.2862
		GS	4719	1167	4.04	195	0.2123	**0.3084**	0.2559
		PG	5172	805	6.25	2	0.1609	0.2084	0.1831
		SLCP^2($k = 1000$)	8657	1494	5.79	257	0.3748	0.2227	0.2889
		ClusterOne[a]	2915	771	4.53	233	0.2623	0.2624	0.2623
		LinkComm[a]	7183	4107	4.58	**614**	**0.4676**	0.1349	0.2510
		GLCP^2[a]	8181	4257	4.50	418	0.3859	0.2413	**0.3051**
HsaBioGrid	PCDq	LCP($k = 1000$)	7042	958	7.35	111	0.2798	0.4945	0.3720
		RMCL	10 698	1536	6.96	223	0.3777	0.5054	0.4369
		MCL ($I_F = 3.3$)	5345	917	5.82	59	0.1668	**0.5563**	0.3046
		GS							
		PG							
		SLCP^2($k = 1800$)	12889	1622	7.95	205	0.3523	0.4281	0.3884
		ClusterOne[a]	10543	1753	10.31	162	0.4098	0.3869	0.3982
		LinkComm[a]	10322	6954	5.70	**372**	**0.4467**	0.2784	0.3526
		GLCP^2[a]	10948	5607	5.31	360	0.4190	0.4943	**0.4545**
	CORUM	LCP($k = 1000$)	958	7042	7.35	166	0.3558	0.2611	0.3047
		RMCL	10698	1536	6.96	190	0.4286	**0.2689**	0.3395
		MCL ($I_F = 3.3$)	5345	917	5.82	82	0.2094	0.2535	0.2304
		GS							
		PG							
		SLCP^2($k = 1800$)	12889	1622	7.95	221	0.4235	0.2331	0.3142
		ClusterOne[a]	10543	1753	10.31	197	0.5797	0.2548	0.3445
		LinkComm[a]	10322	6954	5.70	**724**	**0.6856**	0.1193	0.286
		GLCP^2[a]	10948	5607	5.31	615	0.5047	0.2313	**0.3476**

Note: For *Hsa*BioGRID PPI network, GS and PG do not have results due to the memory limitation. Bold values denote the best scores corresponding to specific criteria.
[a] Overlapping module identification algorithms.

following sections, we further compare the performance of different algorithms on functional module identification, especially for sparse module identification.

3.4 GO term prediction

In this section, we follow the same strategy in Shih and Parthasarathy (2012) to compare the biological significance of identified modules by all nine algorithms with respect to GO term prediction. Instead of using all GO terms, we only consider high-level GO terms with IC larger than two so that we can better understand the functional specificity of identified modules. The comparison for GO term prediction is illustrated in Figure 4A illustrates the F-measure comparison among all the algorithms. Figure 4B shows the percentage of GO terms that are considered to be correctly matched to at least one of the identified modules by different algorithms. Among non-overlapping algorithms, Figure 4 clearly illustrates that SLCP^2 not only detects the largest number of matched high-level GO terms for each PPI network, but also obtains the best F-measure score. Therefore, for non-overlapping module identification, SLCP^2 outperforms other state-of-the-art non-overlapping algorithms on high-level GO term prediction. For overlapping algorithms, based on Figure 4, GLCP^2 identifies more matched GO terms and achieves higher F-measure scores than ClusterOne and LinkComm on two *Sce* PPI networks, which indicates that GLCP^2 outperforms ClusterOne and LinkComm for two yeast networks. For both *Hsa* PPI networks, GLCP^2 and LinkComm uncover competitive numbers of matched GO terms; however, LinkComm obtains better F-measure scores because it gets higher recall scores due to the fact that LinkComm detects a larger number of small overlapping modules, as it does not have a post-processing procedure to deal with highly overlapping modules. These small overlapping modules can be matched to the same GO terms and hence the recall scores can get higher. Among all nine algorithms, for GO term prediction, GLCP^2 and LinkComm perform competitively with each other and outperform the other compared algorithms.

3.5 Sparse module identification

To further illustrate the advantage of our LCP^2 formulation in detecting functional modules with similar interaction patterns, we compare the performances of different algorithms with respect to identifying functional sparse modules in this section. However, in general, as we do not have sparse module golden standards, it is hard to provide quantitative measures for detecting sparse modules. In this section, we provide the examples of well-understood biologically meaningful sparse modules to evaluate the capability of different algorithms in identifying functional sparse modules. Through the comparison of identified corresponding modules, we demonstrate that our SLCP^2 and GLCP^2 are superior in detecting functional sparse modules.

3.5.1 Pro-survival proteins and cytochrome c release The pro-survival proteins (BCL2, MCL1 and BCL2A1), which constitute the Bcl-2 subfamily, directly or indirectly prevent the release of cytochrome c from mitochondria (Yang *et al*., 1997). Therefore, the pro-survival proteins module should interact with the module that has the release of cytochrome c from mitochondria functionality. In Figure 5, we provide the comparison of the module identification results for detecting these two modules in the *Hsa*BioGRID PPI network. For the pro-survival proteins module, we mark the three members in circle shapes. For the functional module with the release of cytochrome c from mitochondria functionality (HRK, BCL2L11, BID, BNIP3, BIK, PMAIP1, BAK1, BMF and BBC3), we mark the members in rectangle shapes. Shaded areas represent the modules detected by the corresponding algorithms. Based on the interactions in the *Hsa*BioGRID PPI network, we find these two modules are sparse with constituent proteins having similar interaction patterns. As shown in Figure 5, LCP detects part of the cytochrome c release module but fails to identify the pro-survival module. RMCL splits pro-survival proteins into two modules. MCL fails to detect both the cytochrome c release module and the pro-survival module. ClusterOne groups those two modules into one. LinkComm fails to detect the pro-survival modules. Only our algorithms SLCP^2 and GLCP^2, which take the interaction patterns into account, achieve the most promising results. For two algorithms PG and GS, which also consider the interaction patterns, we do not have their module detection results because both algorithms run out of memory on this relatively large network.

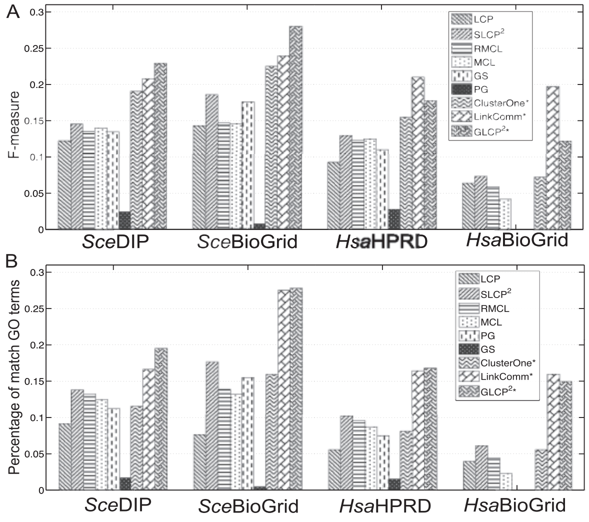
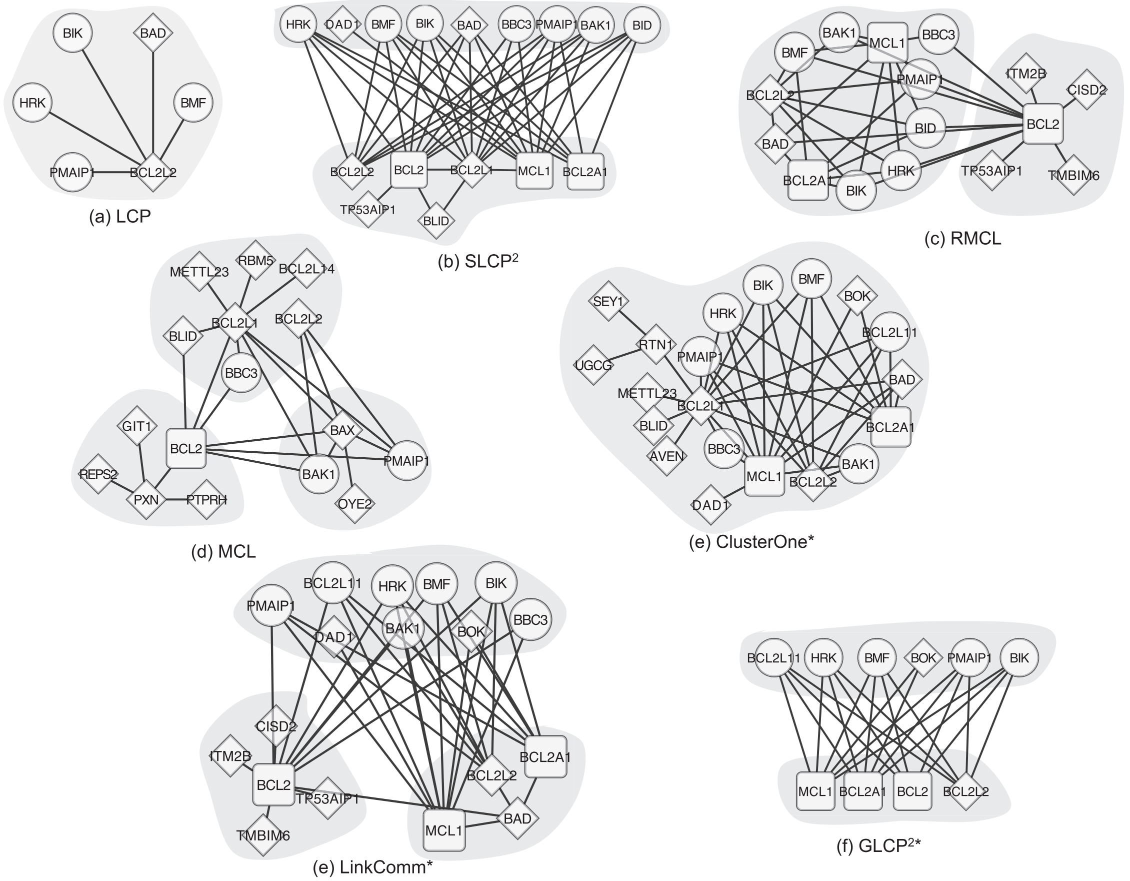


Fig. 4. The top bar figure shows the comparison results based on the *F* measure on four PPI networks. The bottom figure displays the comparison of the percentages of matched GO terms in the complete set of selected high-level GO terms. For the *Hsa*BioGRID PPI network, GS and PG fail to execute due to the memory limitation

3.5.2 FGF/FGFR signaling FGF/FGFR signaling has been associated with a diverse and broad range of biological functions, including cell growth, cell differentiation and the promotion of angiogenesis (Powers *et al*., 2002). FGFR stands for the fibroblast growth factor receptors, which bind to the members of the family of FGF (fibroblast growth factor) proteins. Based on their functionality, FGFR proteins should interact with FGF proteins. Figure 6 illustrates the module identification results for FGFR and FGF modules in the *Hsa*HPRD PPI network. Based on the network structure, FGFR and FGF modules are two sparse modules. We mark the FGFR proteins in rectangle shapes and FGF proteins in circle shapes. Shaded areas represent the modules detected by the corresponding algorithms. As shown in Figure 6, LCP, RMCL, MCL, ClusterOne and LinkComm again can not identify these two modules correctly. PG, GS and our algorithms have the ability to correctly detect them. However, PG and GS over-segment the FGFR module, whereas our algorithms can provide better module identification results.

4 DISCUSSION AND CONCLUSION

The compared module identification algorithms in this article use different module definitions and methods. LCP, ClusterOne, SLCP^2 and GLCP^2 are all based on finding LC sets defined by the Markov chain of random walk on networks. LCP and ClusterOne are the non-overlapping and overlapping algorithms of searching for LC sets defined by the transition matrix P (LCP) of the underlying Markov chain. Therefore, they tend to find densely connected modules. However, SLCP^2 and GLCP^2 are respective algorithms for searching for non-overlapping and overlapping modules by finding LC sets based on the two-hop transition matrix P^2 (LCP^2) of the random walk Markov chain. By taking the advantage of finding two-hop LC sets, our new algorithms detect modules based on the interaction patterns, which reflect functional similarity between proteins. In Satuluri and Parthasarathy (2011), the authors present a similar formulation to search for modules based on the interaction similarity. However, our formulation depending on the Gram matrix W derived by LCP^2 can be viewed as the normalized version of the symmetrization matrix proposed in Satuluri and Parthasarathy (2011). Generally, as in normalized cuts, the normalized version often gives balanced modules that may lead to more



(a) LCP

(b) SLCP^2

(c) RMCL

(d) MCL

(e) ClusterOne*

(e) LinkComm*

(f) GLCP^{2*}

Fig. 5. The pro-survival and cytochrome c release modules in *Hsa*BioGRID PPI network detected by all the algorithms (GS and PG fail to execute because of running out of memory). The pro-survial proteins are in rectangle shapes and the cytochrome c release proteins are in circle shapes. Diamond shapes denote the proteins that belong to neither the pro-survial proteins nor the cytochrome c release proteins. Shaded areas represent the modules detected by the corresponding algorithms

promising functional module identification results. Both MCL and RMCL are network clustering algorithms based on (stochastic) flow simulation that extends the similar random walk Markov chain idea by two operations for better performance: ‘Inflation’ and ‘Expand’. However, both operators are heuristic strategies. Theoretically, why they give good results is still a mystery. PG and SG are two non-overlapping algorithms that identify functional modules in terms of interaction patterns. Because they apply greedy algorithms to solve the module identification problem, the quality of the results is not guaranteed. Last but not least, LinkComm is a novel algorithm based on an edge graph representation that tends to detect a large number of overlapping modules whose biological meaning may not be immediately clear due to the fine-grained modular structure.

In our experiments, we have applied our algorithms to analyze four unweighted PPI networks, which can be viewed as binary ($\{0, 1\}$) edge-weighted networks. However, both SLCP^2 and GLCP^2 can be extended in a straightforward manner for the analysis of general edge-weighted networks by modifying corresponding terms in Algorithms 1 and 2 proposed in this article. We will evaluate the performances of algorithms in module identification by introducing reliable edge weights when they are available in our future work. Another limitation for SLCP^2 is how to decide the desirable number of modules k in advance. One possible way is to search k values within a certain range and choose k with the best average weight density computed by (8). In our future research, we will also explore the ideas adopted in Ahn *et al*. (2010) and Brunet *et al*. (2004) to determine k based on the partition density and/or module entropy score, respectively. Finally, GLCP^2 is our preliminary solution strategy for identifying overlapping modules based on the LCP^2 formulation. We plan to further investigate the properties of the Gram matrix W, and we expect that we may achieve better performance with a better understanding of the problem structure.

In conclusion, we propose a novel formulation to achieve functional module identification based on protein interaction patterns in PPI networks. An efficient spectral algorithm, which can obtain a close-to-optimal solution based on *Ky Fan theorem*, is designed to solve the new optimization problem for non-overlapping module identification. We also develop a greedy algorithm to solve the same problem but obtain overlapping results. Our algorithms not only can overcome the limitation of traditional module identification algorithms, which only focus on identifying dense modules, but they also have a better scalability for large-scale PPI networks to efficiently solve module identification problem. Experimental results show that

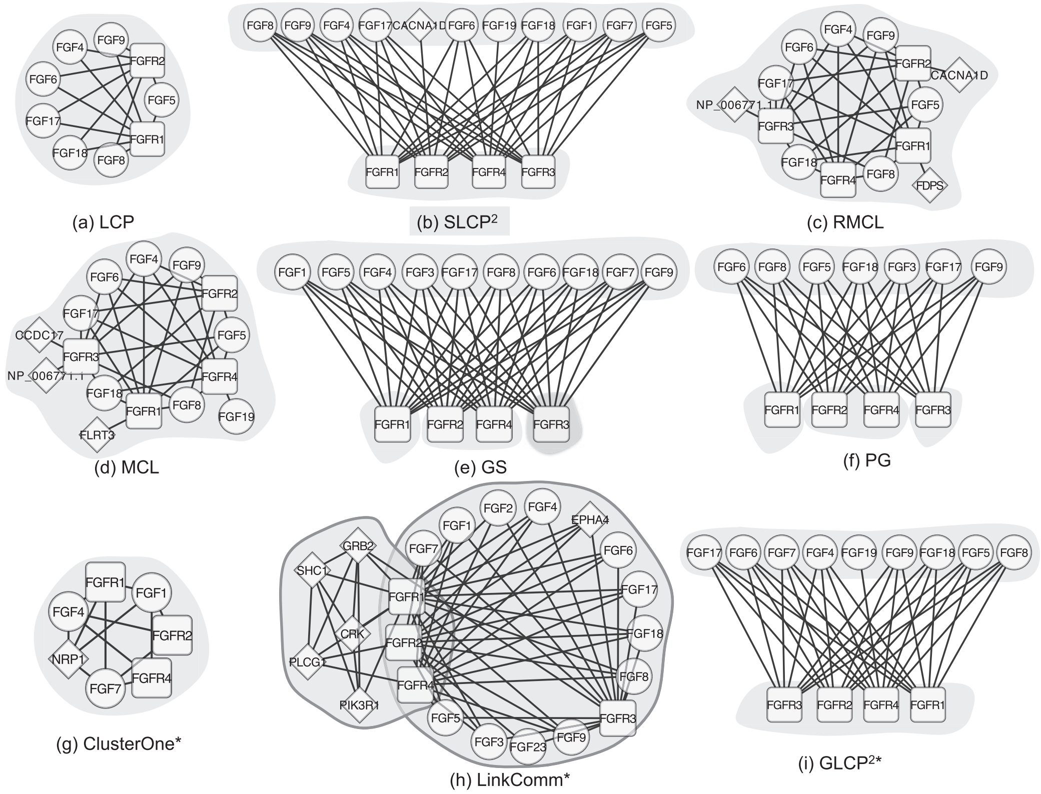

(a) LCP
FGF4
FGF9
FGFR2
FGF6
FGF5
FGF17
FGFR1
FGF18
FGF8
(b) SLCP2
FGF8 FGF9 FGF4 FGF17 CACNA1D FGF6 FGF19 FGF18 FGF1 FGF7 FGF5
FGFR1 FGFR2 FGFR4 FGFR3
(c) RMCL
FGF4 FGF9 FGF6 FGFR2 CACNA1D FGF17 NP_006771.1 FGF5 FGFR3 FGFR1 FGF18 FGFR4 FGF8 FDPS
(d) MCL
FGF4 FGF9 FGF6 FGFR2 FGF17 CCDC17 FGF5 FGFR3 NP_006771.1 FGFR4 FGF18 FGFR1 FGF8 FGF19 FLRT3
(e) GS
FGF1 FGF5 FGF4 FGF3 FGF17 FGF8 FGF6 FGF18 FGF7 FGF9
FGFR1 FGFR2 FGFR4 FGFR3
(f) PG
FGF6 FGF8 FGF5 FGF18 FGF3 FGF17 FGF9
FGFR1 FGFR2 FGFR4 FGFR3
(g) ClusterOne*
FGFR1 FGF1 FGF4 FGFR2 NRP1 FGF7 FGFR4
(h) LinkComm*
FGF2 FGF4 FGF1 EPHA4 GRB2 FGF7 FGF6 SHC1 FGFR1 FGF17 CRK FGF18 FGFR2 PLCG1 FGF8 FGFR4 PIK3R1 FGF5 FGFR3 FGF3 FGF23 FGF9
(i) GLCP2*
FGF17 FGF6 FGF7 FGF4 FGF19 FGF9 FGF18 FGF5 FGF8
FGFR3 FGFR2 FGFR4 FGFR1

Fig. 6. The FGF/FGFR signaling modules in *Hsa*HPRD PPI network detected by all algorithms. FGF proteins are in the circle shapes and FGFR proteins are in the rectangle shapes. Diamond shapes indicate proteins of neither FGF proteins nor FGFR proteins. Shaded areas represent the modules detected by the algorithms

our SLCP^2 and GLCP^2 have achieved promising results on both protein complex and GO term predictions on four large-scale PPI networks. Most importantly, our new algorithms can detect functional sparse modules, which are often ignored by many other existing algorithms.

ACKNOWLEDGEMENTS

The authors thank Deborah Stabler and the anonymous reviewers for helpful suggestions to improve the presentation of this work.

Funding: X Qian was support in part by Award R21DK092845 from the National Institute Of Diabetes And Digestive And Kidney Diseases, National Institutes of Health.

Conflict of Interest: none declared.

REFERENCES

Ahn,Y.Y. *et al.* (2010) Link communities reveal multiscale complexity in networks. *Nature*, **466**, 761–764.

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

Bisgin,H. and Dalfes,H. (2008) Parallel clustering algorithms with application to climatology. In: *Technical report*. Informatics Institute, Istanbul Technical University, Turkey.

Breitkreutz,B. *et al.* (2008) The BioGRID Interaction Database: 2008 update. *Nucleic Acids Res.*, **36**, D637–D640.

Brunet,J. *et al.* (2004) Metagenes and molecular *pattern* discovery using matrix factorization. *Proc. Natl Acad. Sci. USA*, **101**, 4164–4169.

Hofman,J. and Wiggins,C. (2008) A bayesian approach to network modularity. *Phys. Rev. Lett.*, **100**, 258701.

Hong,E. *et al.* (2008) Gene ontology annotations at SGD: new data sources and annotation methods. *Nucleic Acids Res.*, **36**, D577–D581.

Kikugawa,S. *et al.* (2012) PCDq: human protein complex database with quality index which summarizes different levels of evidences of protein complexes predicted from H-invitational protein-protein interactions integrative dataset. *BMC Syst. Biol.*, **6** (**Suppl. 2**), S7.

King,J. (2003) Conductance and rapidly mixing markov chains. In: *Technical report*. University of Waterloo.

Lancichinetti,A. *et al.* (2009) Detecting the overlapping and hierarchical community structure in complex networks. *New J. Phys.*, **11**, 033015.

Li,X. *et al.* (2010) Computational approaches for detecting protein complexes from protein interaction networks: a survey. *BMC Genomics*, **11** (**Suppl. 1**), S3.

Maslov,S. and Sneppen,K. (2002) Specificity and stability in topology of protein networks. *Science*, **296**, 910–913.

Mewes,H.W. *et al.* (2004) MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acids Res.*, **32**, D41–D44.

Morrison,J.L. *et al.* (2006) A lock-and-key model for protein-protein interactions. *Bioinformatics*, **22**, 2012–2019.

Navlakha,S. *et al.* (2008) Graph summarization with bounded error. In: *Processing of the 33rd International Conference on Management of Data (ACM SIGMOD Conference)*. pp. 419–432.

Navlakha,S. *et al.* (2009) Revealing biological modules via graph summarization. *J. Comp. Biol.*, **16**, 253–264.

Nepusz,T. *et al.* (2012) Detecting overlapping protein complexes in protein-protein interaction networks. *Nat. Methods*, **9**, 471–472.

Newman,M. (2006) Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, **74**, 036104.

Newman,M. and Girvan,M. (2004) Finding and evaluating community structure in networks. *Phys. Rev. E*, **69**, 026113.

Phizicky,E. and Fields,S. (1995) Protein-protein interactions: methods for detection and analysis. *Microbiol. Rev*., **59**, 94–123.

Pinkert,S. *et al.* (2010) Protein interaction networks: more than mere modules. *PLoS Comput. Biol*., **6**, e1000659.

Powers,C.J. *et al.* (2002) Fibroblast growth factors, their receptors and signaling. *Endocr. Relat.Cancer*, **7**, 165–197.

Prasad,T. *et al.* (2009) Human Protein Reference Database—2009 update. *Nucleic Acids Res*., **37**, D767–D772.

Raman,K. (2010) Construction and analysis of protein–protein interaction networks. *Autom. Exp*., **2**, 2.

Reichardt,J. (2009) *Structure in Complex Networks*. Lect. Notes Phys. 766, Springer, Berlin, Heidelberg.

Rivas,J. and Fontanillo,C. (2010) Protein–protein interactions essentials: Key concepts to building and analyzing interactome networks. *PLoS Comput. Biol*., **6**, e1000807.

Royer,L. *et al.* (2008) Unraveling protein networks with power graph analysis. *PLoS Comput. Biol*., **4**, e1000108.

Ruepp,A. *et al.* (2008) Corum: the comprehensive resource of mammalian protein complexes. *Nucleic Acids Res*., **36**, D646–D650.

Salwinski,L. *et al.* (2004) The Database of Interacting Proteins: 2004 update. *Nucleic Acids Res*., **32**, D449–D451.

Satuluri,V. and Parthasarathy,S. (2009) Scalable graph clustering using stochastic flows: Applications to community discovery. In: *15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*. Paris, France.

Satuluri,V. and Parthasarathy,S. (2011) Symmetrizations for clustering directed graphs. In: *14th International Conference on Extending Database Technology (EDBT11)*. Uppsala, Sweden.

Satuluri,V. *et al.* (2010) Markov clustering of protein interaction networks. In: *ACM Conference on Bioinformatics, Computational Biology and Biomedicine 2010*. New York, USA.

Shih,Y. and Parthasarathy,S. (2012) Identifying functional modules in interaction networks through overlapping markov clustering. *Bioinformatics*, **28**, i473–i479.

Stark,C. *et al.* (2006) BioGRID: a general repository for interaction datasets. *Nucleic Acids Res*., **34**, D535–D539.

van Dongen,S. (2000) A cluster algorithm for graphs. In: *Technical Report INS-R0010*. National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam.

Voevodski,K. *et al.* (2009) Finding local communities in protein networks. *BMC Bioinformatics*, **10**, 297.

Wang,Y. and Qian,X. (2012) Functional module identification by block modeling using simulated annealing with path relinking. In: *ACM Conference on Bioinformatics, Computational Biology and Biomedicine 2012*. Orlando, USA

Wang,Y. and Qian,X. (2013) A novel subgradient-based optimization algorithm for block model functional module identification. *BMC Bioinformatics*, **14** (**Suppl 2**), S23.

Xing,E. and Jordan,M. (2003) On semidefinite relaxation for normalized k-cut and connections to spectral clustering. In: *Technical report UCB/CSD-03-1265*. EECS Department, University of California, Berkeley.

Yang,J. *et al.* (1997) Prevention of apoptosis by Bcl-2: release of cytochrome c from mitochondria blocked. *Science*, **275**, 1129–1132.

Zha,H. *et al.* (2001) Spectral relaxation for k-means clustering. In: *Advances in Neural Information Processing Systems*. Vol. 14, pp. 1057–1064.