From Function to Interaction: A New Paradigm for Accurately Predicting Protein Complexes Based on Protein-to-Protein Interaction Networks

Bin Xu and Jihong Guan

Abstract—Identification of protein complexes is critical to understand complex formation and protein functions. Recent advances in high-throughput experiments have provided large data sets of protein-protein interactions (PPIs). Many approaches, based on the assumption that complexes are dense subgraphs of PPI networks (PINs in short), have been proposed to predict complexes using graph clustering methods. In this paper, we introduce a novel from-function-to-interaction paradigm for protein complex detection. As proteins perform biological functions by forming complexes, we first cluster proteins using biology process (BP) annotations from gene ontology (GO). Then, we map the resulting protein clusters onto a PPI network (PIN in short), extract connected subgraphs consisting of clustered proteins from the PPI network and expand each connected subgraph with protein nodes that have rich links to the proteins in the subgraph. Such expanded subgraphs are taken as predicted complexes. We apply the proposed method (called CPredictor) to two PPI data sets of *S. cerevisiae* for predicting protein complexes. Experimental results show that CPredictor outperforms the existing methods. The outstanding precision of CPredictor proves that the from-function-to-interaction paradigm provides a new and effective way to computational detection of protein complexes.

Index Terms—Protein complex, protein-protein interaction networks, functional similarity, prediction

1 Introduction

Protein-protein interactions (PPIs) are fundamental to biological processes (BP). Most proteins perform their biological functions by forming complexes [1], [2]. For example, the hemoglobin molecule is an assembly of four globular protein subunits. Identification of protein complexes can help us to understand biological progresses as well as to predict protein functions.

The wet lab experiment, tandem affinity purification with mass spectrometry (TAP-MS) [3], is a technique to detect protein complexes. However, TAP-MS has a disadvantage that low transient affinity complexes are hard to extract due to the multiple times of washing and purification. Gavin et al. [1] indicated that only limited known yeast protein complex subunits can be detected by TAP-MS.

In recent years, high-throughput methods such as yeast two hybrid (Y2H) [4] and tandem affinity purification (TAP) have provided us large-scale protein-protein interaction data sets of different organisms [5], [6]. Many studies have tried to detect protein complexes computationally based on PPI data. These approaches treat PPI data as a graph or network where proteins are nodes and interactions are edges between nodes. Protein complexes are expected to be dense subgraphs in the PPI network (PIN in short) [7], as they are usually made up of proteins with common biological functions. Therefore, the problem of detecting protein complexes computationally can be addressed by locating dense subgraphs in PPI networks by using clustering techniques [8].

Up to date, researchers have proposed different methods for protein complex detection by extracting densely connected subgraphs from PPI networks, including MCODE [7], Clique [9], LCMA [10], CFinder [11] and CMC [12]. These methods focus mainly on the topological structures of PPI networks while ignoring their biological properties. However, the high false positive and false negative rates of PPI data from high-throughput experiments [13] make it difficult to predict protein complexes accurately.

Some studies [14], [15], [16], [17], [18], [19] therefore exploited additional information such as gene ontology (GO) and expression data to enhance the confidence of interactions between proteins and subsequently to construct more reliable PPI networks. Various subgraph extraction methods have been employed upon these networks and have achieved better prediction performance.

Some other approaches such as core-attachment model based methods [20], [21] and supervised methods [22] have been shown to further improve the prediction power.

In summary, existing methods have shown considerable promise for complex detection. Basically, they follow the strategy of extracting dense subgraphs from PPI networks, where edges may be additionally weighted with different information including topology, expression data as well as protein functions. Essentially, they are mainly based on protein interactions. However, the suspect quality of PPI data hinders the performance improvement on protein complex prediction.

In this paper, we propose a new complex detection paradigm, which first clusters proteins based on *functional similarity*, then maps the resulting protein clusters onto a *PPI network* to extract connected protein subgraphs, and finally

• *The authors are with the Department of Computer Science and Technology, Tongji University, Shanghai 201804, China. E-mail: xebecsean@gmail.com, jhguan@tongji.edu.cn.*

Manuscript received 30 Dec. 2013; accepted 24 Jan. 2014. Date of publication 18 Feb. 2014; date of current version 4 Aug. 2014.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TCBB.2014.2306825

expands the connected protein subgraphs to obtain protein complexes. So it employs a *from-function-to-interaction* paradigm, in the sense that it first exploits *function* information, then uses *interaction* data.

On the one hand, as proteins perform their biological functions by forming complexes, proteins in a real complex are more possible to have high *functional similarity*. Given the protein interaction data, we first calculate a similarity matrix of all proteins, where each element represents the functional similarity between two proteins. Functional similarity is evaluated using Biology Process (BP) annotations from gene ontology [23]. Then, the spectral clustering [24] method is applied to the matrix so as to derive clusters composed of proteins that are likely to cooperate to perform common functions. So far, proteins are clustered solely based on protein functional information.

On the other hand, it has been widely admitted that proteins in a real complex interact with each other at high probability, thus form a dense subgraph in PPI networks. Based on this point, we map each cluster of proteins generated above into a PPI network and extract connected subgraphs consisting of some or all proteins in the cluster as nodes. Unlike the existing methods, we do not require that the extracted subgraphs are densely connected. Moreover, considering that most proteins in complexes tend to have interactions with one or a few hub-proteins [25], we expand each connected subgraph with proteins that link to most proteins in the subgraph. Thus, a protein may be included in more than one expanded subgraph. These expanded subgraphs are finally taken as predicted protein complexes.

The above from-function-to-interaction paradigm can identify complexes of low density as we require only the extracted subgraphs to be connected. We call this method *CPredictor*. The experiments show that *CPredictor* can retrieve most real complexes from the benchmark set with the highest precision.

The rest of paper is organized as follows: Section 2 surveys related work. Section 3 introduces our method CPredictor for protein complex prediction. Section 4 presents performance evaluation on two yeast data sets. Finally, Section 5 concludes the paper.

2 Related Work

A PPI network can be presented as an undirected graph $G = (V, E)$ where V is the set of proteins and E is the set of edges that stand for the protein interactions. Two nodes in a PPI network are linked if there exists observed interaction between them. A subgraph of G can be represented by $G' = (V', E')$ where $V' \subseteq V$ and $E' \subseteq E$. The *density* of graph G that indicates the abundance of edges in the graph, is calculated by

$$density(G) = \frac{2 \times |E|}{|V| \times (|V| - 1)}. \quad (1)$$

If the PPI network is a weighted graph, where each edge is associated with a certain value w, then the density is evaluated as

$$density(G) = \frac{2 \times \sum_{e \in E} w(e)}{|V| \times (|V| - 1)}. \quad (2)$$

In recent years, a number of methods have been proposed to predict protein complexes by applying graph clustering approaches to finding dense subgraphs in PPI networks. These approaches can be roughly and incompletely classified into the following types.

2.1 Methods Based on Dense Subgraph Detection

MCODE [7] is one of the earliest computational methods to detect protein complexes. A k-core of a graph G is a maximal connected subgraph where all vertices have degree at least k. The k-core score of a node can be determined by locating the maximal connected subgraph containing the node. The local neighborhood of a node consists of the nodes directly connecting to it and all the edges between these nodes. There can be at most $n * (n-1)/2$ interactions in the neighborhood with n nodes. And the local neighborhood density of a node is the actual number of its edges divided by its maximal possible number of interactions. MCODE first weighs each vertex in the network by the product of its k-core score and the local neighborhood density. The node with the largest weight is selected as the seed node that represents the first cluster. Then, each of its neighbors is traversed and is included into the cluster if its weight exceeding a certain threshold. The same procedure is performed iteratively over the neighboring nodes of the cluster seed to determine whether any neighboring node should be added into the cluster. Certainly, a node will not be checked twice during the process. This process stops when no more nodes can be added and the resulting cluster represents the first predicted complex. Among the remaining, unprocessed nodes in the network, the node with the largest weight is selected as a new seed node to discover the second cluster in the same way as above. Such a process continues till all nodes are processed. All resulting clusters are taken as predicted complexes.

MCODE also provides "fluff" and "haircut" options. Given a resulting cluster, each neighbor of the cluster's members is further checked and can be assigned into the cluster if its weight exceeds the fluff parameter. So the resulting clusters may overlap with each other. The "haircut" option removes singly connected nodes in each resulting cluster.

In addition to the MCODE method, there are also some alternatives to extract dense subgraphs from PPI networks.

Pereira-Leal et al. [26] weighted edges in the PPI network by confidence value, which is measured by the number of experiments that support the interactions. Then, both ends of each interaction are condensed into a single node. Nodes are linked if they have a common protein node. And the edge weight of each linkage is the average of interaction confidence values in the original network. Finally, a Markov clustering method [27] is applied to finding clusters as protein complexes.

The hub duplication method [28] selects nodes with degree larger than 25 as hub proteins. It first identifies dense subgraphs in the neighborhoods of hub proteins. Then, a new hub-induced subgraph is constructed from the neighborhoods, where each hub protein is duplicated and the duplicates are connected to each other. Each identified subgraph is connected only to one duplicate. Dense regions inside each hub-induced subgraph are detected for complex prediction.

The SCAN method [29] defines structural similarity of two proteins as the number of common neighbors between them and a structure-reachable protein pair as two proteins with structural similarity larger than a threshold. SCAN identifies a complex by first selecting a core protein that has the largest number of structure-reachable neighbors and expanding the core protein as the initial member of a cluster by checking its neighbors iteratively.

Navlakha et al. [30] first compressed a PPI network into a summary graph. Then in the summary graph, each node represents a cluster of proteins that have similar neighbors in PPI network, which is taken as a complex.

2.2 Methods Based on Clique Detection

Several methods including Clique [9], LCMA [10], CFinder [11] and CMC [12] identify protein complexes from PPI networks by locating, merging and modifying cliques according to respective criteria. These methods attach great importance to strong connectivity of detected subgraphs when predicting protein complexes.

The Clique method proposed by Spirin and Mirny attempts to discover complexes by exhaustively searching cliques in a PPI network. They also applied super-paramagnetic clustering (SPC) and Monte Carlo simulation to identifying densely-connected subgraphs. The detected cliques are further checked with their statistical significances that indicate the probabilities of their occurrences in a comparable random graph. According to the statistical significance, these cliques are further cleaned, merged and selected.

The Local Clique Merging Algorithm (LCMA) first obtains the cliques in the local neighborhood of a node. The complexes are detected by merging these local cliques if they share a high overlapping score.

CFinder explores k-cliques from the PPI network and defines k-cliques sharing $(k-1)$ common nodes as neighbors. Adjacent cliques are merged to obtain complexes.

CMC calculates the weighted densities of cliques detected from a weighted PPI network [31]. Cliques are merged if the edge weights in the non-overlapping area exceed a certain threshold.

2.3 Methods Integrating PPI and Gene Expression Data

Experimental PPI data sets are susceptible to false positives and false negatives [13]. When exploiting PPI networks by mainly topology attributes, the above methods tend to neglect peripheral complex members with few links. Therefore, some studies integrated additional information into PPI networks to resolve the inaccuracy problem resulting from false connections and to better reveal the biological meaning of the detected complexes. As proteins interacting with each other are likely to have similar gene expression profiles, a number of methods, including GFA [14], DMSP [15] and MATISSE [16] proposed different strategies to weight edges between proteins in PPI networks with the help of gene expression data.

In GFA, a protein v is weighted by $e^{-expression(v)}$ where $expression(v)$ is the log fold change of v's gene expression profile. Therefore, the density of a subgraph $G' = (V', E')$ can be calculated by $|E|/\sum w(v)$. Then, GFA extracts subgraphs from the network to maximize the density of the subgraphs. The subgraphs occurring in different microarray data are discarded and the resulting clusters are taken as complexes.

The DMSP method first clusters proteins based on gene expression data. The edge weight of two nodes is evaluated by the distance of each node to its corresponding cluster's center and the distance between the two corresponding clusters' centers. Clusters are detected by expanding a kernel protein set that is composed of a seed protein and a certain number of its neighbors. The goal is to find subgraphs that have the largest local weighted density.

In MATISSE, clusters are detected in a weighted network, where edges are weighted by gene expression correlation between the corresponding proteins and a node's weight is its weighted degree.

2.4 Methods Exploiting Functional Similarity Information

Since proteins usually perform functions by forming complex, high functional correlation is expected from real complexes. Therefore, various methods, such as SWEMODE [18] and OIIP [19], evaluate the reliability of protein-protein interaction by functional similarity between proteins. Identification of protein complexes is carried out afterwards by finding dense clusters. Concretely, SWEMODE calculates the semantic similarity of GO terms of linked nodes and assigns weights to the edges. OIIP counts the number C of common GO annotations of any two proteins x and y. The set $S_{g_i}(x, y)$ represents the set of annotated proteins (including both x and y) on the GO term g_i. The edge between x and y is weighted according to C and the size of S. The weight of a node is the sum of the weights of its incident edges.

Here, functional information is used to weight the interaction between proteins, which is quite different from our method, where functional similarity is only used for protein clustering.

2.5 Core-Attachment Model Based Methods

A promising strategy of complex detection was proposed by Gavin et al. [2]. They discovered that many complexes follow a core-attachment structure. Core proteins are those with dense connection while attachment proteins have only a few links to the core proteins.

Leung et al. [20] proposed a new complex prediction method based on the core-attachment model mentioned above. It first calculates the probability that two nodes interact and the probability that they share a certain number of common neighbors, and then evaluates a joint probability that two nodes have interaction and share a certain number of common neighbors. A small joint probability suggests that the two nodes is likely in a complex. All pairs of proteins are evaluated in this manner to determine the sets of core proteins. Given a core protein set, they treat proteins that are common neighbors of more than half of the core set members as attachment proteins. The resulting core-attachment sets are regarded as complexes.

Wu et al. [21] identifies core proteins as nodes with degree larger than average degree of its neighborhood members. For the neighborhood of each core protein, nodes

with degree at least the average degree of the neighborhood are selected, these nodes constitute one or more connected subgraphs. Then, the core protein is added back to each subgraph, which forms a core set. The common neighbors of a core set's members are regarded as its attachment proteins.

2.6 Method Using Topological and Biological Features

Qi et al. [22] argued that real complexes observe diverse structures and proposed a supervised method to predict protein complexes. They trained a Bayesian network using a set of known complexes to verify whether a candidate subgraph is complex. The features used consist of topological patterns such as node size, graph density and biological patterns such as average protein length and average protein weight.

3 Method

In this section, we present the technical details of our method. First, we give an overview of the method, and then introduce protein similarity computation based on gene ontology, protein clustering based on a spectral method with the protein similarity matrix, complex generation by mapping protein clusters onto a PPI network, and performance evaluation metrics, respectively.

3.1 Overview

Fig. 1 shows the flowchart of our method CPredictor. CPredictor consists of three major stages: protein functional similarity computation, protein clustering and complex generation.

Protein functional similarity computation. As most proteins interact collaboratively to perform biological functions, proteins constituting a complex possibly have similar functions. So we first examine the functions of proteins in a PPI network. Concretely, we compute the functional similarity between proteins by using BP annotations from GO. Thus, a functional similarity matrix S is obtained, where each element represents the functional similarity of two proteins.

Protein clustering. With the computed similarity matrix S, we cluster the proteins with a spectral algorithm to group the proteins into a set C of K clusters, each of which is of assumedly similar function.

Complex generation. This stage can be further split to three steps:

1) *Cluster mapping*. The generated clusters are mapped onto a PPI network.
2) *Subgraph extraction*. Connected subgraphs are extracted from each cluster C_i mapped onto the PPI network. Those subgraphs with less than three proteins are thrown away directly.
3) *Subgraph expansion*. For the set of extracted subgraphs, each of which consists of connected proteins of similar function, and maybe constitute the core of a possible complex. We expand each subgraph with additional proteins that are closely connected with the subgraph. Concretely, for an extracted subgraph $G_i^j = (N_i^j, E_i^j)$, N_i^j and E_i^j are the sets of nodes and edges of G_i^j respectively, we expand it by the proteins that connect to a number n_p of proteins in the subgraph, and we require that n_p is no less than $\lfloor \alpha \times |N_i^j| \rfloor$. Here, α is a prespecified parameter.

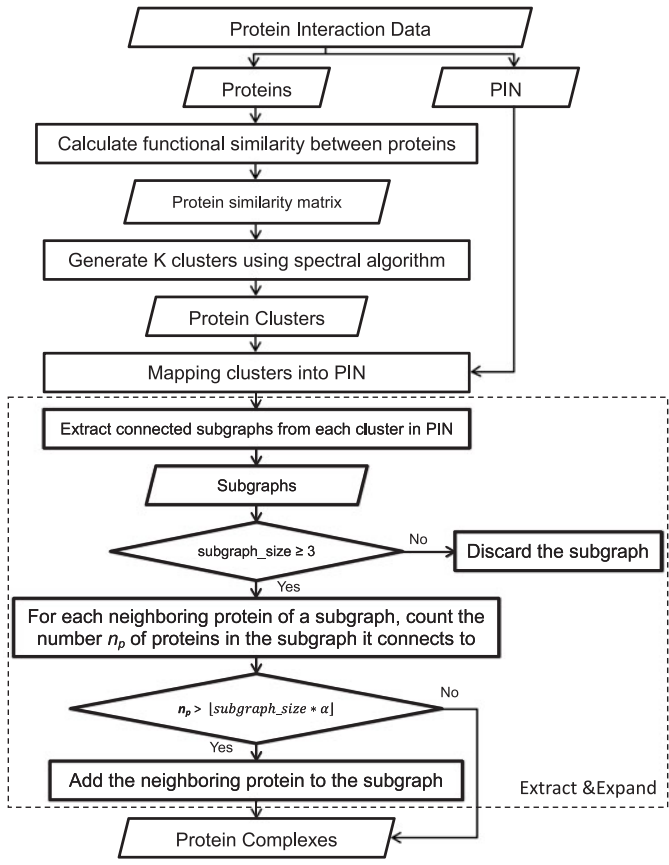


Fig. 1. The flowchart of CPredictor.

It is worth noting that in our method proteins from interaction data are first clustered solely based on the functional similarity as protein complexes are fully or partly made up of functional similar proteins. Then given the PPI network, protein clusters are mapped onto the network and connected subgraphs or components are extracted from the clusters. As some protein interactions in a real complex may be hard to detect experimentally, we do not discard the lowly-rated interactions in the PPI network. A subgraph is extracted from each functionally-cohesive cluster as long as it is connected in the PPI network. This procedure differs from most existing graph-based clustering methods. Those studies cluster proteins based on the assessment of interactions among proteins according to various metrics, such as socio-affinity score [2], the number of common neighbors [32] and so on. These criteria may be too general for all different protein interactions.

Also, it should be noted that unlike the methods based on core-attachment structure where *core* proteins should be densely connected [20], [21], the extracted subgraphs in our method are only required to be connected.

3.2 Protein Functional Similarity Computation

In our method, to find the preliminary clusters of proteins that are likely to perform common biological functions, we first calculate protein functional similarity by using the

method proposed by Wang et al. [33], which is based on the similarity of GO terms, and consider only biological process annotations in GO. In what follows, we present the similarity computation method.

GO provides a controlled vocabulary of terms for describing gene and gene product attributes across all species [23]. It consists of three major categories: biological process, molecular function (MF), and cellular component (CC). BP is a series of events accomplished by one or more ordered assemblies of molecular functions. MF describes activities that occur at the molecular level. CC consists of the location of the cell. GO exhibits a directed acyclic graph (DAG) structure and a child concept can be an instance or a component of its parent concept. One concept may have multiple parents with relations among "is_a," "part_of," "regulates," "positively regulates" and "negatively regulates" etc.

In Wang's method, a term A is represented by using $DAG_A(A, T_A, E_A)$ where T_A is the set of GO terms, including term A and all its ancestor terms; E_A is the set of edges connecting the nodes in DAG_A. The semantic value of term A is the accumulation of all the other terms in DAG_A. The S-value, the semantic contribution of a term t to A, is defined as

$$S_A(t) = \begin{cases} 1, & t = A \\ max\{w_e \times S_A(t') | t' \in children\ of(t)\}, & t \neq A, \end{cases} \tag{3}$$

where w_e is the weight of edge between t' and t, which is 0.8 for 'is_a' edge, and 0.6 for 'part_of' edge.

The semantic value of term A is calculated as

$$SV(A) = \sum_{t \in T_A} S_A(t). \tag{4}$$

The similarity between two terms A and B is calculated as

$$S_{GO}(A, B) = \frac{\sum_{t \in T_A \cap T_B} (S_A(t) + S_B(t))}{SV(A) + SV(B)}, \tag{5}$$

where t is a common ancestor of A and B.

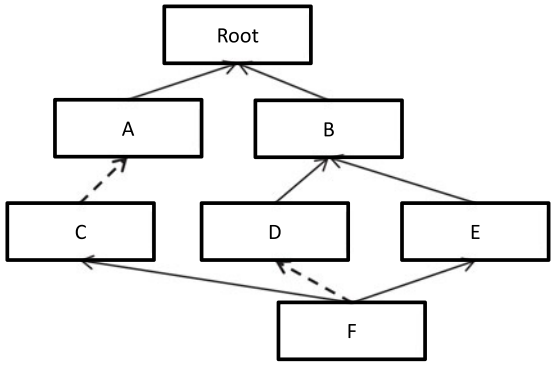
Usually, a protein is annotated by several GO terms, and then the functional similarity between proteins is calculated as

$$Sim(P_1, P_2) = \frac{\sum_{1 \leq i \leq m} Sim(go_{1i}, P_2) + \sum_{1 \leq j \leq n} Sim(go_{2j}, P_1)}{m + n}, \tag{6}$$

where P_1 and P_2 are annotated by m and n terms respectively. Sim(go, P) is the maximum semantic similarity between a GO term go and any of the k terms annotated to protein P:

$$Sim(go, P) = \max_{1 \leq i \leq k} (S_{GO}(go, go_i)). \tag{7}$$

Fig. 2 gives an example of calculating S-value.



Term t	F	E	D	C
S-value $S_F(t)$	1	0.8	0.6	0.8
Term t	B	A	Root	
S-value $S_F(t)$	0.64	0.48	0.512	

Fig. 2. Illustration of S-value computation. Here, the upper part of the figure is the DAG of term 'F', T_F contains seven terms: 'A', 'B', 'C', 'D', 'E', 'F' and 'Root'. The solid arrows indicate 'is_a' edges; and dashed arrows are 'part_of' edges. The S-value of each term in T_F to 'F' is given in the table.

3.3 Clustering Proteins by Spectral Method

Spectral clustering [24] uses the spectrum (eigenvalues) of the similarity matrix of the original data to achieve dimension reduction, and then performs clustering upon the derived data with fewer dimensions. Given a $n \times n$ similarity matrix $S = [s_{i,j}]$ of a set of n proteins $P = \{p_1, p_2, \ldots, p_n\}$ where $s_{i,j}$ is the functional similarity of protein i and protein j, we suppose that these proteins are divided into K disjoint groups $\{C_1, \ldots, C_K\}$.

The steps of spectral clustering are as follows:

1) Construct an unnormalized graph Laplacian matrix $L = D - S$, where D is a diagonal matrix with $D_{i,i} = \sum_{j=1}^{n} s_{i,j}$.
2) Find the eigenvectors $\{v_i | i = 1, 2, \ldots, K\}$ corresponding to the K smallest eigenvalues. The derived n-by-K matrix formed by those eigenvectors can be regarded as a low-dimension representation of the original matrix.
3) Perform K-means clustering [34].

The K-means algorithm works as follows: Given a set of observations $(x_1, x_2, \ldots, x_n)$ where each is a d-dimension vector, K-means clustering tries to partition the data into K disjoint subsets $S = \{S_1, S_2, \ldots, S_K\}$ so as to minimize

$$J = \sum_{i=1}^{K} \sum_{x_j \in S_i} \|x_j - \mu_i\|, \tag{8}$$

where μ is the mean value of points in S_i.

The algorithm starts at randomly choosing an initial set of K points which represents the centers of K clusters. Then each data is assigned to the cluster whose center is the closest one. Each center is updated by using the means of the data points assigned the corresponding cluster after all data points are assigned. The algorithm repeats the steps mentioned above till the centers no longer change or a certain number of iterations is reached.

3.4 Complex Generation

We first map the protein clusters obtained above onto a PPI network, and then extract subgraphs from the

clusters in the PPI network, and finally expand the subgraphs. The resulting subgraphs are taken as complexes. In what follows, we focus on subgraph extraction and expansion.

3.4.1 Subgraph Extraction

Given a protein interaction network $G=(N,E)$ where N represents the set of proteins and E represents the set of protein-protein interactions in the network. Each cluster C_i mapped onto the network can be seen as a subgraph of G, denoted by $G_i=(N_i,E_i)$ where N_i is the set of proteins in C_i, i.e., $N_i=C_i$, and E_i is the set of edges in G that connect any two proteins in N_i.

As G_i is not necessary a connected graph, it may consists of several disconnected subgraphs spreading over G, we are to extract all these subgraphs. We employ the breadth-first search (BFS) algorithm upon G_i to get subgraphs. When the first subgraph $G_i^1=(N_i^1,E_i^1)$ $(N_i^1 \subseteq N_i)$ is obtained, all its nodes and edges are removed from G_i, and the BFS method is performed $(G_i - G_i^1)$ to find the other subgraphs. Such a procedure repeats till all subgraphs are obtained. Suppose there are totally l connected subgraphs, we have $\bigcup_{j=1}^{l} G_i^j = G_i$ and $G_i^p \cap G_i^q = \emptyset$ for $1 \le p,q \le l$, $p \ne q$. We discard those extracted subgraphs that contain less than three proteins.

3.4.2 Subgraph Expansion

For each subgraph G_i^j extracted from G_i (corresponding to the functionally-cohesive proteins set C_i) of the PPI network, there are a number of proteins in N but not in N_i^j, which have interactions with proteins in N_i^j. These proteins constitute the set of neighboring proteins of G_i^j, denoted as NP_i^j, which can be formally represented as follows:

$$NP_i^j = \{p | E(p, G_i^j) \ne \emptyset \ and \ p \notin N_i^j\}, \tag{9}$$

where $E(p,G_i^j)$ is the set of edges between protein $p \notin N_i^j$ and all proteins in G_i^j. Among the proteins in NP_i^j, some are used to expand G_i^j, which make up a set of proteins ENP_i^j for expanding G_i^j as follows:

$$ENP_i^j = \{p \,|\, p \in (N - N_i^j) \ and \ |E(p, G_i^j)| \ge \lfloor \alpha \times |N_i^j| \rfloor\}, \tag{10}$$

where $|E(p,G_i^j)|$ is the number of interactions between protein p in N and proteins in G_i^j. $|N_i^j|$ is the size of N_i^j, and α is a prespecified threshold to control the required number of interactions for a neighboring protein to be added to the subgraph. Therefore, $N_i^j \cup ENP_i^j$ is a complex predicted by our method. As an added neighboring protein may also appear in different complexes, the predicted complexes may overlap with one another. However, if the overlap rate between the two complexes exceeds 0.8, then we merge them as one complex. For two complexes CC_i and CC_j, the overlap rate OR is computed as follows:

$$OR_{ij} = \frac{|CC_i \cap CC_j|}{|CC_i \cup CC_j|}, \tag{11}$$

where CC_i and CC_j also indicate the sets of proteins included in the two complexes.

The process of subgraph extraction and expansion is outlined in Algorithm 1. Here, Lines 1-13 describe the extraction of connected subgraphs; Lines 15-27 are for subgraph expansion.

Algorithm 1 The algorithm of subgraph extraction and expansion.

Input:
The set of protein clusters C;
PPI network $G=(N,E)$;
Output:
Predicted complexes set PC;
1: $SG = \emptyset$; /* the set of connected subgraphs */
2: **for** $i = 1$ *to* $|C|$ **do**
3: $G_i = (N_i, E_i)$; /* G_i is the graph corresponding to cluster C_i, N_i is the set of proteins in C_i and E_i is set of edges between proteins in N_i */
4: $j = 1$;
5: **while** $N_i \ne \emptyset$ **do**
6: Find the largest connected subgraph G_i^j in G_i by breadth first search;
7: **if** $|G_i^j| > 2$ **then**
8: $SG = SG \cup G_i^j$;
9: **end if**
10: $G_i = G_i - G_i^j$;
11: j++;
12: **end while**
13: **end for**
14: $PC = \emptyset$;
15: **for** each subgraph G_i^j in SG **do**
16: N_i^j = the set of proteins in G_i^j;
17: NP_i^j = the neighbors of G_i^j;
18: $ENP_i^j = \emptyset$;
19: **for** each protein p in NP_i^j **do**
20: $conns$ = the number of edges between $p \in NP_i^j$ and proteins in N_i^j;
21: **if** $conns > \lfloor \alpha \times |N_i^j| \rfloor$ **then**
22: $ENP_i^j = ENP_i^j \cup p$;
23: **end if**
24: **end for**
25: $N_i^j \cup ENP_i^j$ forms a new complex CP_i^j;
26: $PC = PC \cup CP_i^j$;
27: **end for**

Fig. 3 is an example to illustrate the major steps of complex prediction with our method CPredictor.

3.5 Performance Evaluation Metrics

Here we use two sets of performance evaluation metrics: 1) precision, recall and F1-measure; 2) Maximum matching ratio (MMR), maximum matching recall (MM recall) and maximum matching precision (MM precision).

3.5.1 Precision, Recall and F1-Measure

Let $B=\{b_1,b_2,\ldots,b_m\}$ and $PC=\{pc_1,pc_2,\ldots,pc_n\}$ be the benchmark set and the set of predicted complexes respectively. Given a real complex $b_i \in B$ and a predicted one $pc_j \in PC$, we check the overlap rate between them:

$$w = \frac{|V_{b_i} \cap V_{pc_j}|^2}{|V_{b_i}| \times |V_{pc_j}|}, \tag{12}$$

where $|V_{b_i}|$ is the number of proteins in the real complex b_i and $|V_{pc_j}|$ is the size of the predicted complex pc_j. Usually, if $w \ge 0.2$, it is considered that the predicted complex pc_j matches with the real one b_i. Let M_b represent

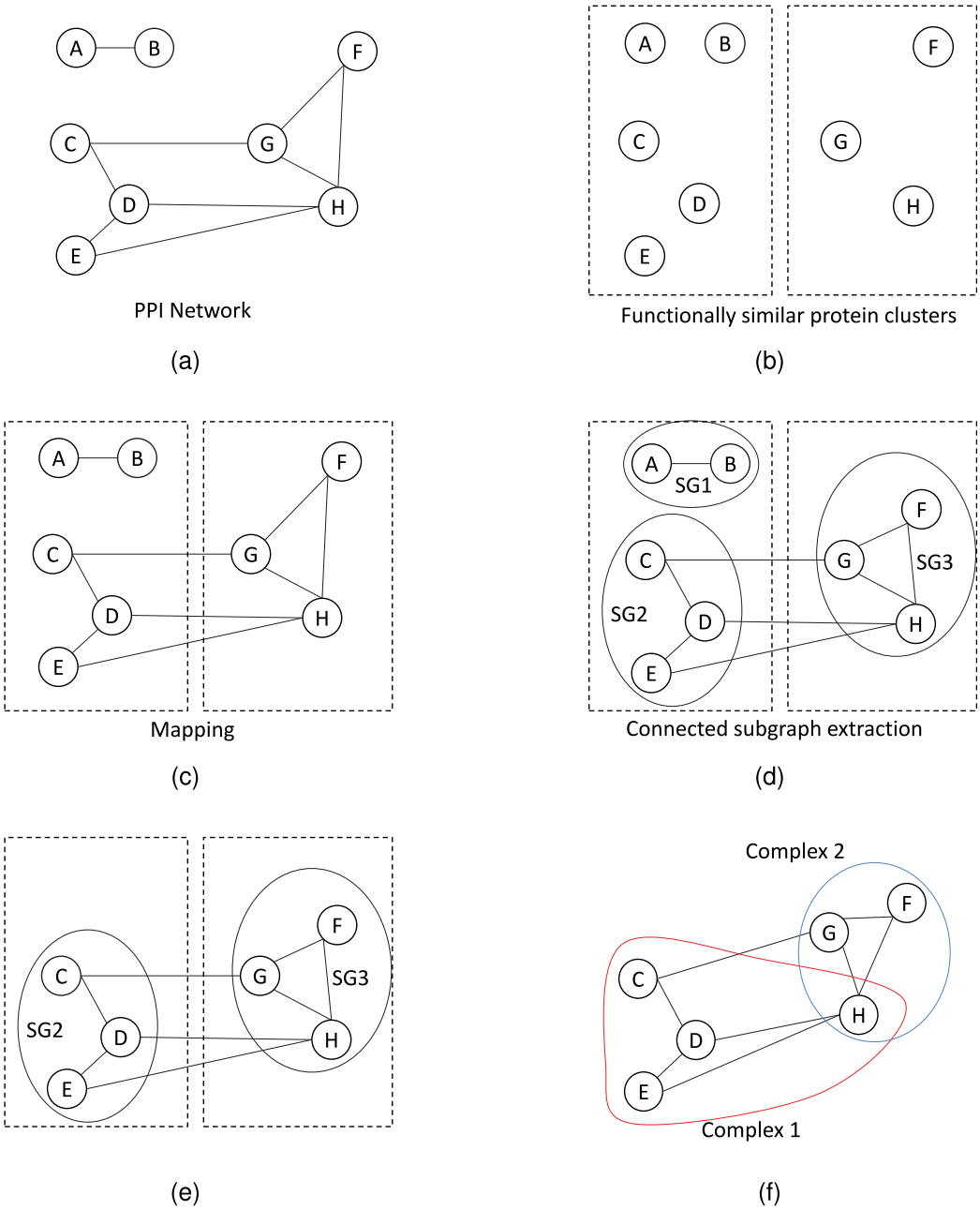


Fig. 3. An example of complex prediction by CPredictor. (a) A PPI network consists of eight nodes (proteins) and nine edges. (b) All eight proteins are grouped by using spectral method to two clusters (set $K = 2$). (c) Mapping the clusters onto the PPI network. (d) Three connected subgraphs (marked by oval) extracted from the PIN. (e) As the size of subgraph SG1 is less than 3, it is discarded. (f) Expanding the subgraphs. The parameter α is set to 0.5, then the threshold for adding neighbors to SG2 is $\lfloor 0.5 \times 3 \rfloor = 1$. Node 'H' is the neighboring node of node 'D' and node 'E', it has two edges with SG2, therefore it is regard as a member of complex 1, which consists of proteins 'C', 'D', 'E' and 'H'. The predicted complex 2 consists of proteins 'F', 'G' and H'.

the number of benchmark complexes that have at least one matched predicted complex and M_{pc} represent the number of predicted complexes that match at least one real complex. *Recall* and *precision* are defined as follows:

$$Recall = \frac{M_b}{|B|},$$
$$Precision = \frac{M_{pc}}{|PC|}, \tag{13}$$

where $|B|$ is the number of benchmark complexes and $|PC|$ is the number of predicted complexes. *F1-measure* (or simply $F1$) is the harmonic mean of *recall* and *precision* evaluated by

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision}. \tag{14}$$

3.5.2 Maximum Matching Analysis

In [35], Nepusz et al. proposed to use *maximum matching ratio* to evaluate the result of complex prediction.

The MMR is evaluated in a bipartite graph where the two sets of nodes are real complexes and predicted ones respectively, and a set of weighted edges between the two sets of

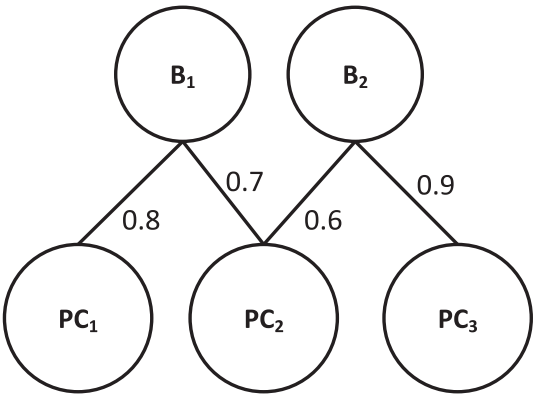


Fig. 4. An example of MMR evaluation. Here, B_1 and B_2 are members of the real or reference set; PC_1, PC_2 and PC_3 are members of the predicted complexes. There is an edge between a reference complex and a predicted one if the overlap score of the two exceeds a given threshold (say 0.2). PC_1 matches better with B_1 than PC_2, and PC_3 matches better with B_2 than PC_2. Therefore, MMR is $(0.8 + 0.9)/2 = 0.85$.

nodes represent the overlap score between real and predicted complexes. Note that the set of edges contains only node pairs with overlap score larger than a prespecified threshold.

Intuitionally, one real complex should not be assigned to more than one predicted complex, and vice versa. We start from choosing the edge with the largest weight, and then we remove the corresponding nodes and all edges linked to these nodes in the bipartite graph. This procedure is repeated till there is no edge in the bipartite graph. MMR is calculated as the mean of weights of the selected edges. It measures how accurately the predicted complexes represent the real complexes. Fig. 4 shows an example of MMR evaluation.

Following the maximum matching idea, we further given the definitions of *recall* and *precision* in the one-to-one matching situation. Let the maximum matching number (MMN) be the number of matching node-pairs, the *maximum matching recall* and the *maximum matching precision* are evaluated as follows:

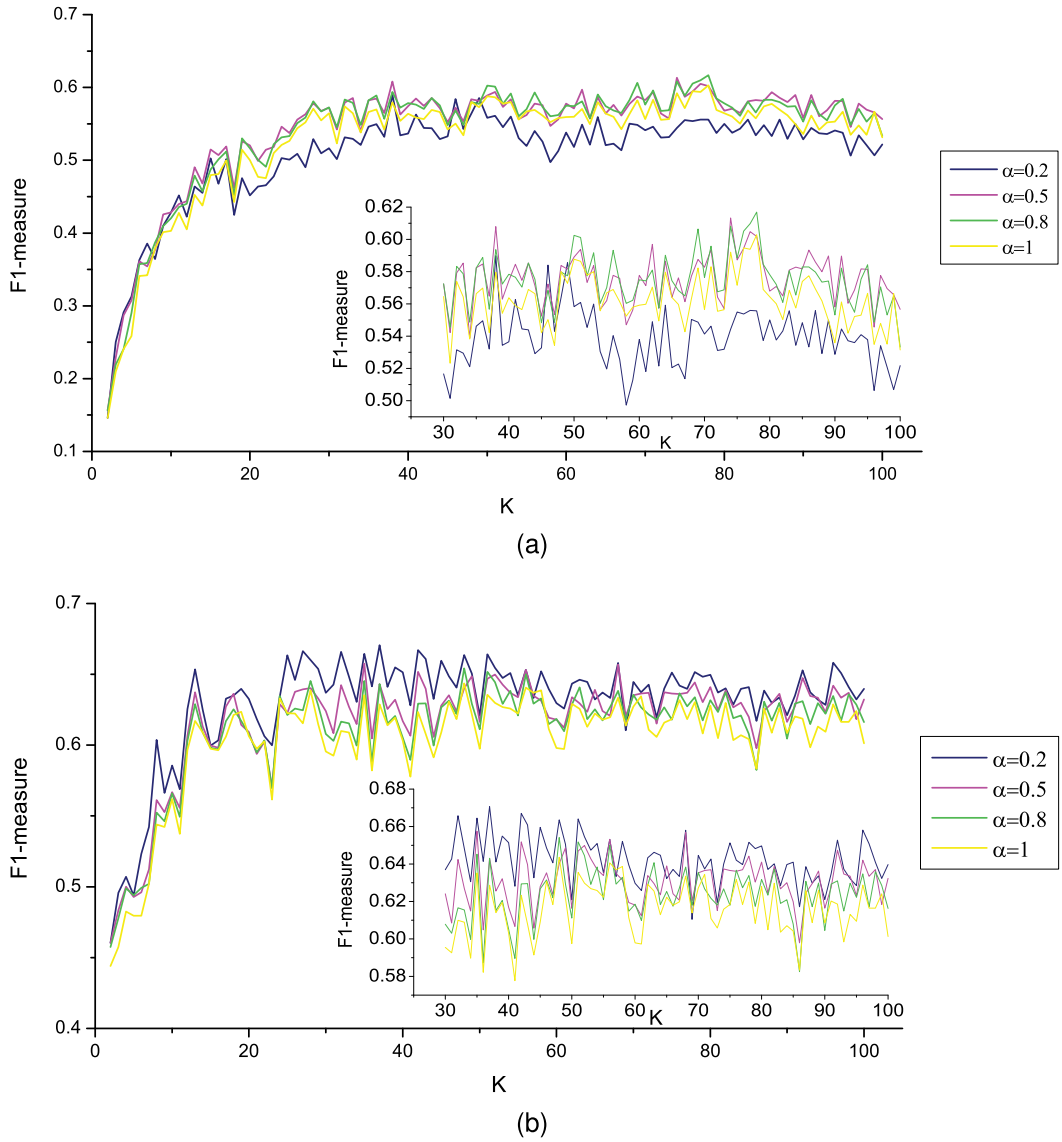


Fig. 5. Performance of CPredictor with different K and α values for (a) Gavin et al. data set, and (b) Collins et al. data set. Performance is evaluated by $F1\text{-}measure$. The cluster number K is set from 2 to 100.

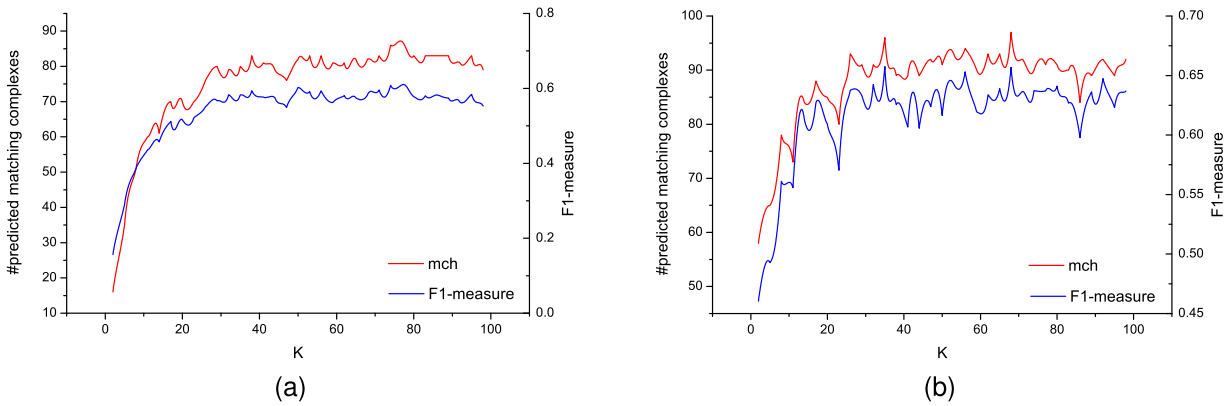


Fig. 6. Prediction performance for different K values on (a) Gavin et al. data set with $\alpha = 0.8$, and (b) Collins et al. data set with $\alpha = 0.6$. The number of predicted matching complexes is denoted by *mch*. Both *mch* and *F1-measure* reach relatively stable when K is larger than 30.

TABLE 1
The Effect of Parameter α (the Gavin et al. Data Set, $K = 76$)

α	mch	F1	Recall	Precision	MMR	MM Recall	MM Precision
0.1	70	0.50	0.47	0.53	0.50	0.38	0.50
0.2	79	0.55	0.53	0.58	0.57	0.42	0.55
0.3	81	0.58	0.55	0.62	0.61	0.45	0.59
0.4	84	0.59	0.57	0.62	0.62	0.45	0.60
0.5	85	0.59	0.57	0.62	0.63	0.45	0.60
0.6	84	0.59	0.57	0.62	0.63	0.45	0.60
0.7	**88**	**0.61**	**0.59**	**0.63**	0.63	**0.46**	**0.61**
0.8	**88**	**0.61**	**0.59**	0.62	0.65	**0.46**	**0.61**
0.9	87	0.60	**0.59**	0.62	**0.66**	**0.46**	0.60
1.0	87	0.59	**0.59**	0.60	0.65	0.45	0.58

TABLE 2
The Effect of Parameter α (the Collins et al. Data Set, $K = 68$)

α	mch	F1	Recall	Precision	MMR	MM Recall	MM Precision
0.1	94	0.65	0.64	0.67	0.68	**0.53**	**0.64**
0.2	95	0.66	0.64	**0.68**	0.71	0.52	**0.64**
0.3	94	0.65	0.64	0.67	**0.73**	0.51	0.63
0.4	95	0.65	0.64	0.67	**0.73**	0.51	0.63
0.5	**97**	**0.66**	**0.66**	0.66	0.72	0.52	**0.64**
0.6	96	0.65	0.65	0.66	**0.73**	0.52	**0.64**
0.7	93	0.64	0.63	0.65	**0.73**	0.51	0.63
0.8	94	0.64	0.64	0.64	0.72	0.51	0.63
0.9	94	0.64	0.64	0.64	0.71	0.51	0.63
1.0	94	0.63	0.64	0.63	0.69	0.51	0.61

$$MM\_Recall = \frac{MMN}{|B|}, \qquad MM\_Precision = \frac{MMN}{|PC|}. \tag{15}$$

4 Performance Evaluation

In this section, we present the results of performance evaluation.

4.1 Data Sets and Experimental Settings

The protein-protein interaction data of *Saccharomyces cerevisiae* are provided by Gavin et al. (1,855 proteins and 7,669 interactions) [2] and Collins et al. (1,622 proteins and 9,074 interactions) [36]. The similarity matrix is calculated using the GOSemsim package [37] implemented in R [38]. The protein complex data set CYC2008 [39] is used, which contains 408 manually curated protein complexes. A collection of complexes with at least three proteins, 148 in total, serves as the benchmark set.

We set the threshold of overlap rate w to 0.2. That is, if the overlap rate between a real complex and a predicted one exceeds 0.2, we consider them matched. We first examine the effect of parameters K and α on prediction performance, and then compare our method with the state-of-the-art complex prediction methods.

TABLE 3
The Number of Predicted Complexes on the Gavin et al. Data Set

CPredictor	MCODE	RNSC	DPClus	ClusterOne	CFinder	CORE	mcl
112	130	242	223	196	137	217	224

4.2 Effect of Parameters

Here we test the effect of parameter K and α on prediction performance. As the size of most protein complexes is below 30 and we only have a few thousands of proteins in the protein interaction data, we think that setting the largest value of K to 100 is proper to test our method. Also, to comprehensively evaluate the proposed method, we set K from 2, and increase it value to 100. The parameter α is to control subgraph expansion, which is set from 0.1 to 1.

Before studying the effect of each parameter separately, we show in Fig. 5 the results of combining the two parameters. We can see that CPredictor obtains the highest F1-measure when $K = 76$, $\alpha = 0.8$ for the Gavin et al. data set and $K = 68$, $\alpha = 0.5$ for the Collins et al. data set. Note that on the Collins et al. data set, although the F1-measure is higher when $\alpha = 0.2$, the corresponding MMR is the lowest. Therefore, we choose $\alpha = 0.5$.

We then fix α and change K to see how prediction performance varies with K value. Fig. 6 shows the number of predicted matching complexes and F1-measure of our method when K changes from 2 to 100 for the two data sets.

It is clear to see that the performance gets better as K increases. However, when K reaches 30, we witness relatively stable performance for both mch and F1-measure.

We further check the generated clusters for different K values. We find that there exist several large clusters (size > 100) when K is relatively small. As K grows, these large clusters begin to split into small clusters while the other clusters seldom change. This may explain the stable performance when K is relatively large. It also implies that K cannot be too small.

Finally, we examine the effect of parameter α by changing its value from 0.1 to 1.0 while fixing K to 76 for Gavin et al. data set and to 68 for Collins et al. The results for both data sets are shown in Tables 1 and 2.

For the Gavin et al. data set, as shown in Table 1, the performance is relatively poor when α is small (0.1 and 0.2), which indicates that too many proteins added into the subgraphs during the expansion step does not benefit prediction performance. As α increases, we generally witness a slow but not monotonic increasing trend for all metrics. When α reaches 0.7-0.9, we see a slight down trend. The number of real complexes predicted by our method increases to 88 when $\alpha = 0.7$ or 0.8. Overall, the best α for the Gavin et al. data set is 0.7, at which six out of the total seven metrics are best.

TABLE 4
The Number of Predicted Complexes on the Collins et al. Data Set

CPredictor	MCODE	RNSC	DPClus	ClusterOne	CORE	mcl
120	111	177	160	100	160	156

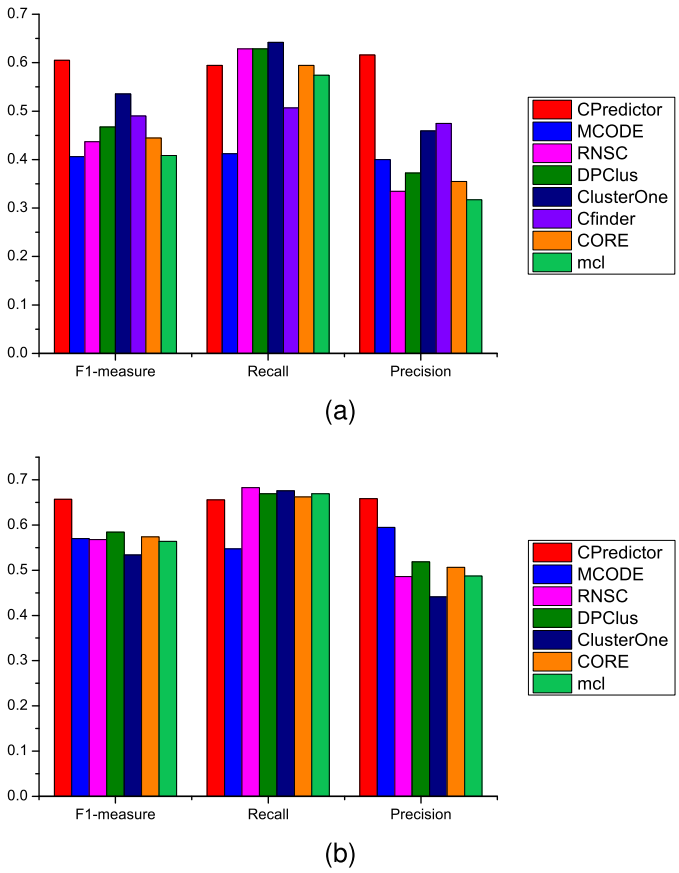


Fig. 7. Performance comparison on (a) the Gavin et al. data set, and (b) the Collins et al. data set by recall, precision and F1-measure. CPredictor achieves the highest F1-measure and precision among all methods.

The results on Collins et al. data set is given in Table 2, where we see that the performance varies slightly for different α values, and the best performance is achieved when $\alpha = 0.5$, as four of the seven metrics have the highest values.

In summary, the results above suggest that a general principle of setting K and α is to avoid too large and too small values.

4.3 Performance Comparison

Here we compare our method CPredictor with seven existing complex prediction methods, including MOCDE [7], RNSC [17], DPClus [32], ClusterOne [35], CFinder [11], CORE [20] and mcl [26].

The parameters of these methods, if have any, are set as suggested in their original papers or as default. Details are listed as follows:

1) MCODE: haircut = FALSE, fluff = TRUE, VWP = 0.05.
2) RNSC: Default.
3) DPClus: CP $= 0.5$, Density $= 0.7$, min size $= 3$.
4) CFinder: K $= 4$. CFinder cannot obtain prediction results on the Collins et al. data within a week.
5) CPredictor: $K = 76$ and $\alpha = 0.8$ for the Gavin et al. data set; $K = 68$ and $\alpha = 0.5$ for the Collins et al. data set.

Tables 3 and 4 show the number of predicted complexes of each method on the Gavin et al. data set and the Collins

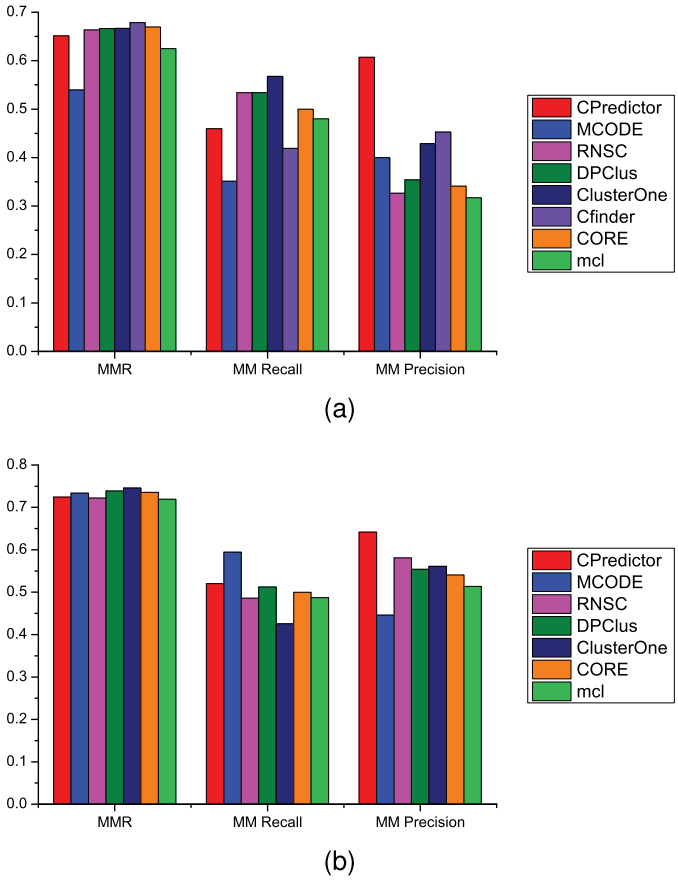


Fig. 8. Performance comparison on (a) the Gavin et al. data set, and (b) the Collins et al. data set by maximum matching analysis. CPredictor achieves the highest MM_Precision, competitive MMR and MM_Recall.

et al. data set, respectively. Fig. 7 illustrates the results of F1-measure, recall and precision of different methods, Fig. 8 shows the results of MMR, MM_Recall and MM_Precision of different methods.

As can be seen from Fig. 7, our method clearly outperforms the other methods in both precision and F1-measure. The highest precision achieved by our method CPredictor indicates that the subgraphs generated by our method have higher possibility to be real complexes. Moreover, the recall of CPredictor is still comparable to the other methods although we predict much less complexes than the other methods. In other words, many predicted complexes output by the other methods are possibly false positives.

Fig. 8 shows that although the MMR values of different methods are very close to each other, our method substantially dominates the other methods in MM_precision, which means that our method can offer the prediction with higher accordance and less redundancy.

5 Conclusion

In this study, we have proposed a novel method to detect protein complexes. Different from the existing methods, which employ a from-interaction-to-function idea, we adopt a from-function-to-interaction paradigm. Concretely, proteins are first clustered according to the functional similarity measured by their biological process annotations. Then, protein clusters are mapped onto a PPI network, from which connected subgraphs are extracted and expanded. The resulting subgraphs are taken as complexes.

We have applied our method to Sacchromyces cerevisiae. Experimental results have shown that our method outperforms the major existing methods.

Acknowledgments

Jihong Guan is the corresponding author. This work was partially supported by National Natural Science Foundation of China under grants No. 61173118 and No. 61272380.

References

[1] A.-C. Gavin et al., "Functional Organization of the Yeast Proteome by Systematic Analysis of Protein Complexes," *Nature*, vol. 415, no. 6868, pp. 141-147, 2002.

[2] A.-C. Gavin et al., "Proteome Survey Reveals Modularity of the Yeast Cell Machinery," *Nature*, vol. 440, no. 7084, pp. 631-636, 2006.

[3] G. Rigaut, A. Shevchenko, B. Rutz, M. Wilm, M. Mann, and B. Séraphin, "A Generic Protein Purification Method for Protein Complex Characterization and Proteome Exploration," *Nature Biotechnology*, vol. 17, no. 10, pp. 1030-1032, 1999.

[4] P.L. Bartel and S. Fields, *The Yeast Two-Hybrid System*. Oxford Univ. Press, 1997.

[5] P. Uetz et al., "A Comprehensive Analysis of Protein–Protein Interactions in Saccharomyces Cerevisiae," *Nature*, vol. 403, no. 6770, pp. 623-627, 2000.

[6] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki, "A Comprehensive Two-Hybrid Analysis to Explore the Yeast Protein Interactome," *Proc. Nat'l Academy of Sciences USA*, vol. 98, no. 8, pp. 4569-4574, 2001.

[7] G.D. Bader and C.W. Hogue, "An Automated Method for Finding Molecular Complexes in Large Protein Interaction Networks," *BMC Bioinformatics*, vol. 4, no. 1, article 2, 2003.

[8] C. Pizzuti, S.E. Rombo, and E. Marchiori, "Complex Detection in Protein-Protein Interaction Networks: A Compact Overview for Researchers and Practitioners," *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, pp. 211-223, Springer, 2012.

[9] V. Spirin and L.A. Mirny, "Protein Complexes and Functional Modules in Molecular Networks," *Proc. Nat'l Academy of Sciences USA*, vol. 100, no. 21, pp. 12123-12128, 2003.

[10] X.-L. Li et al., "Interaction Graph Mining for Protein Complexes Using Local Clique Merging," *Genome Informatics Series*, vol. 16, no. 2, p. 260, 2005.

[11] B. Adamcsek, G. Palla, I.J. Farkas, I. Derényi, and T. Vicsek, "CFinder: Locating Cliques and Overlapping Modules in Biological Networks," *Bioinformatics*, vol. 22, no. 8, pp. 1021-1023, 2006.

[12] G. Liu, L. Wong, and H.N. Chua, "Complex Discovery from Weighted PPI Networks," *Bioinformatics*, vol. 25, no. 15, pp. 1891-1897, 2009.

[13] C. von Mering, R. Krause, B. Snel, M. Cornell, S.G. Oliver, S. Fields, and P. Bork, "Comparative Assessment of Large-Scale Data Sets of Protein–Protein Interactions," *Nature*, vol. 417, no. 6887, pp. 399-403, 2002.

[14] J. Feng, R. Jiang, and T. Jiang, "A Max-Flow-Based Approach to the Identification of Protein Complexes Using Protein Interaction and Microarray Data," *IEEE/ACM Trans. Computational Biology and Bioinformatics*, vol. 8, no. 3, pp. 621-634, May/June 2011.

[15] I.A. Maraziotis, K. Dimitrakopoulou, and A. Bezerianos, "Growing Functional Modules from a Seed Protein via Integration of Protein Interaction and Gene Expression Data," *BMC Bioinformatics*, vol. 8, no. 1, article 408, 2007.

[16] I. Ulitsky and R. Shamir, "Identification of Functional Modules Using Network Topology and High-Throughput Data," *BMC Systems Biology*, vol. 1, no. 1, article 8, 2007.

[17] A. King, N. Pržulj, and I. Jurisica, "Protein Complex Prediction via Cost-Based Clustering," *Bioinformatics*, vol. 20, no. 17, pp. 3013-3020, 2004.

[18] Z. Lubovac, J. Gamalielsson, and B. Olsson, "Combining Functional and Topological Properties to Identify Core Modules in Protein Interaction Networks," *Proteins: Structure, Function, and Bioinformatics*, vol. 64, no. 4, pp. 948-959, 2006.

[19] B. Xu, H. Lin, and Z. Yang, "Ontology Integration to Identify Protein Complex in Protein Interaction Networks," *Proteome Science*, vol. 9, no. suppl. 1, p. S7, 2011.
[20] H.C. Leung, Q. Xiang, S. Yiu, and F.Y. Chin, "Predicting Protein Complexes from PPI Data: A Core-Attachment Approach," *J. Computational Biology*, vol. 16, no. 2, pp. 133-144, 2009.
[21] M. Wu, X. Li, C.-K. Kwoh, and S.-K. Ng, "A Core-Attachment Based Method to Detect Protein Complexes in PPI Networks," *BMC Bioinformatics*, vol. 10, no. 1, p. 169, 2009.
[22] Y. Qi, F. Balem, C. Faloutsos, J. Klein-Seetharaman, and Z. Bar-Joseph, "Protein Complex Identification by Supervised Graph Local Clustering," *Bioinformatics*, vol. 24, no. 13, pp. i250-i268, 2008.
[23] T.Z. Berardini, V.K. Khodiyar, R.C. Lovering, and P. Talmud, "The Gene Ontology in 2010: Extensions and Refinements," *Nucleic Acids Research*, vol. 38, no. database issue, pp. D331-D335, 2010.
[24] U.Von Luxburg, "A Tutorial on Spectral Clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395-416, 2007.
[25] B. Chen, J. Shi, and F.-X. Wu, "Not AU Protein Complexes Exhibit Dense Structures in S. Cerevisiae PPI Network," *Proc. IEEE Int'l Conf. Bioinformatics and Biomedicine (BIBM)*, pp. 1-4, 2012.
[26] J.B. Pereira-Leal, A.J. Enright, and C.A. Ouzounis, "Detection of Functional Modules from Protein Interaction Networks," *Proteins: Structure, Function, and Bioinformatics*, vol. 54, no. 1, pp. 49-57, 2004.
[27] A.J. Enright, S. Van Dongen, and C.A. Ouzounis, "An Efficient Algorithm For Large-Scale Detection of Protein Families," *Nucleic Acids Research*, vol. 30, no. 7, pp. 1575-1584, 2002.
[28] D. Ucar, S. Asur, U. Catalyurek, and S. Parthasarathy, "Improving Functional Modularity in Protein-Protein Interactions Graphs Using Hub-Induced Subgraphs," *Proc. 10th European Conf. Principle and Practice of Knowledge Discovery in Databases (PKDD '06)*, pp. 371-382, 2006.
[29] M. Mete, F. Tang, X. Xu, and N. Yuruk, "A Structural Approach for Finding Functional Modules from Large Biological Networks," *BMC Bioinformatics*, vol. 9, no. suppl. 9, article S19, 2008.
[30] S. Navlakha, M.C. Schatz, and C. Kingsford, "Revealing Biological Modules via Graph Summarization," *J. Computational Biology*, vol. 16, no. 2, pp. 253-264, 2009.
[31] G. Liu, J. Li, and L. Wong, "Assessing and Predicting Protein Interactions Using Both Local and Global Network Topological Metrics," *Genome Informatics*, vol. 22, pp. 138-149, 2008.
[32] M. Altaf-Ul-Amin, Y. Shinbo, K. Mihara, K. Kurokawa, and S. Kanaya, "Development and Implementation of an Algorithm for Detection of Protein Complexes in Large Interaction Networks," *BMC Bioinformatics*, vol. 7, no. 1, article 207, 2006.
[33] J.Z. Wang, Z. Du, R. Payattakool, S.Y. Philip, and C.-F. Chen, "A New Method to Measure the Semantic Similarity of Go Terms," *Bioinformatics*, vol. 23, no. 10, pp. 1274-1281, 2007.
[34] J.B. Macqueen, "Some Methods of Classification and Analysis of Multivariate Observations," *Proc. Fifth Berkeley Symp. Math. Statistics and Probability*, pp. 281-297, 1967.
[35] T. Nepusz, H. Yu, and A. Paccanaro, "Detecting Overlapping Protein Complexes in Protein-Protein Interaction Networks," *Nature Methods*, vol. 9, no. 5, pp. 471-472, 2012.
[36] S.R. Collins, P. Kemmeren, X.-C. Zhao, J.F. Greenblatt, F. Spencer, F.C. Holstege, J.S. Weissman, and N.J. Krogan, "Toward a Comprehensive Atlas of the Physical Interactome of Saccharomyces Cerevisiae," *Molecular & Cellular Proteomics*, vol. 6, no. 3, pp. 439-450, 2007.
[37] G. Yu, F. Li, Y. Qin, X. Bo, Y. Wu, and S. Wang, "Gosemsim: An R Package for Measuring Semantic Similarity among Go Terms and Gene Products," *Bioinformatics*, vol. 26, no. 7, pp. 976-978, 2010.
[38] R. Ihaka and R. Gentleman, "R: A Language for Data Analysis and Graphics," *J. Computational and Graphical Statistics*, vol. 5, no. 3, pp. 299-314, 1996.
[39] S. Pu, J. Wong, B. Turner, E. Cho, and S.J. Wodak, "Up-to-Date Catalogues of Yeast Protein Complexes," *Nucleic Acids Research*, vol. 37, no. 3, pp. 825-831, 2009.

Bin Xu is currently working toward the PhD degree in the Department of Computer Science and Technology, Tongji University, Shanghai 201804, China. His research interests include protein interface hotspot prediction and complex prediction based on data mining and machine learning methods.

Jihong Guan received the bachelor's degree from Huazhong Normal University, Wuhan, China, in 1991, the master's degree from the Wuhan Technical University of Surveying and Mapping (merged into Wuhan University since August 2000), Wuhan, in 1998, and the PhD degree from Wuhan University, Wuhan, in 2002, respectively. Before joining Tongji University, she was in the Department of Computer, Wuhan Technical University of Surveying and Mapping from 1991 to 1997 as an assistant professor where she became an associate professor in August 2000. She was an associate professor from August 2000 to October 2003 and became a professor in November 2003 in the School of Computer, Wuhan University. She is currently a professor in the Department of Computer Science and Technology, Tongji University, Shanghai, China. She has published more than 100 papers in domestic and international journals and conferences. Her research interests include databases, data mining, distributed computing, bioinformatics, and geographic information systems.