Tsinghua Science and Technology

Volume 22 | Issue 1 Article 4

2017

Framework to Identify Protein Complexes Based on Similarity Preclustering

Xiaoqing Peng
the School of Information Science and Engineering, Central South University, Changsha 410083, China.

Xiaodong Yan
the School of Information Science and Engineering, Central South University, Changsha 410083, China.

Jianxin Wang
the School of Information Science and Engineering, Central South University, Changsha 410083, China.

TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 04/11 pp42–51
Volume 22, Number 1, February 2017

Framework to Identify Protein Complexes Based on Similarity Preclustering

Xiaoqing Peng, Xiaodong Yan, and Jianxin Wang*

Abstract: Proteins interact with each other to form protein complexes, and cell functionality depends on both protein interactions and these complexes. Based on the assumption that protein complexes are highly connected and correspond to the dense regions in Protein-protein Interaction Networks (PINs), many methods have been proposed to identify the dense regions in PINs. Because protein complexes may be formed by proteins with similar properties, such as topological and functional properties, in this paper, we propose a protein complex identification framework (KCluster). In KCluster, a PIN is divided into K subnetworks using a K-means algorithm, and each subnetwork comprises proteins of similar degrees. We adopt a strategy based on the expected number of common neighbors to detect the protein complexes in each subnetwork. Moreover, we identify the protein complexes spanning two subnetworks by combining closely linked protein complexes from different subnetworks. Finally, we refine the predicted protein complexes using protein subcellular localization information. We apply KCluster and nine existing methods to identify protein complexes from a highly reliable yeast PIN. The results show that KCluster achieves higher Sn and Sp values and f-measures than other nine methods. Furthermore, the number of perfect matches predicted by KCluster is significantly higher than that of other nine methods.

Key words: protein complex; similarity preclustering; protein-protein interaction networks; K-means

1 Introduction

Protein complexes formed by the interaction of proteins are always part of the biological processes of a cell. For example, enzymatic complexes ensure substrate channeling that drastically increases fluxes through metabolic pathways. Large protein complexes, such as histones, RNA polymerase complexes, DNA polymerase complexes, ribosomes, and proteasomes, play essential roles in basal cellular mechanisms such as DNA packaging, transcription, replication, translation, and protein degradation. Studying protein complexes is a fundamental requirement for understanding cellular mechanisms and protein functions. Many experimental techniques and prediction methods have made possible the construction of the large-scale Protein-protein Interaction Networks (PINs) of many species. These PINs provide a comprehensive framework for the use of computational methods to predict protein complexes and protein functions[1].

Based on the observation that highly interconnected or dense regions in PINs may represent protein complexes[2], many methods have been proposed to identify these highly connected areas as such. The density (d) of a subgraph with n vertices and m edges is usually used to determine whether a subgraph is highly connected, which is defined as $d = 2m/(n\times(n-1))$[3]. Based on the density, clique and maximal cliques are used in several algorithms to identify protein complexes[3–7]. In some algorithms, such as MCODE[2], DPClus[8], IPCA[9], and ClusterOne[10], a "seed and extension" paradigm is used to detect

• Xiaoqing Peng, Xiaodong Yan, and Jianxin Wang are with the School of Information Science and Engineering, Central South University, Changsha 410083, China. E-mail: xqpeng@mail.csu.edu.cn; yxdcsu@foxmail.com; jxwang@mail.csu.edu.cn.

∗ To whom correspondence should be addressed.
Manuscript received: 2015-11-16; revised: 2016-01-28; accepted: 2016-02-03

dense sub-graphs in PINs. These methods use different strategies for seed selection, cluster expansion, and stop conditions in the detection of protein complexes. Random walk models can be used to discover dense regions, and several methods have been proposed to identify protein complexes based on random walk[11–14]. With respect to the uncertainty relationship between nodes, Zhao et al.[15] developed an algorithm to detect complexes based on the uncertain graph model, and employed expected density and relative degree to determine whether a subgraph represents a complex with high cohesion and low coupling. The visualization of clustering results can help to better understand the structures of biological networks. Wang et al.[16] developed ClusterViz, a Cytoscape 3 app, which provides three clustering algorithms, FAG-EC[17], EAGLE[18], and MCODE[2], for cluster analysis by which clustering results can be visualized.

Hierarchical clustering is one of the most common classification methods used in PINs to detect protein complexes[19–22], in which hierarchical clustering algorithms represent the network hierarchy as a tree. With respect to the different processes used in constructing the tree, hierarchical clustering algorithms can be classified into two groups: agglomerative and divisive. The G-N algorithm[19] is a classic divisive algorithm, which iteratively computes the edge betweenness of all edges, removes those with the highest score, and draws the corresponding part of the dendrogram if at least two of the resulting subgraphs fulfill the module definition. The HC-PIN algorithm[22] performs fast agglomerative clustering by combining two clusters if an edge between them has the highest edge clustering coefficient and both clusters satisfy the module definition.

In their study of the organization of protein complexes, Gavin et al.[23] found that a complex consists of two parts: a core and an attachment. Based on this core-attachment structure, they proposed the CoreMethod[24] and COACH[25] to identify complexes from the PINs by separately identifying their cores and attachments. Using the weighted PageRank-Nibble algorithm, the WPNCA method[14] first divides the PIN into multiple dense clusters, and then detects the core-attachment structures in these clusters as the predicted protein complexes.

Recently, some methods for integrating multiple sources of information have been proposed to identify protein complexes. Li et al.[26] integrated PIN and gene expression data to identify protein complexes and functional modules on a time-course PIN constructed by Tang et al.[27] Dynamic protein information has also been used to identify protein complexes. Wang et al.[28] used a 3-sigma principle on time-serial gene expression data to differentiate the inactive and active points of each protein. Based on the extracted dynamic protein information, authors proposed a protein complex formation model based on the just-in-time mechanism and applied this model to refine the prediction of protein complexes by clustering algorithms[29].

Other methods first cluster the proteins that appear at the same time by constructing time-series PINs, and then identify the protein complexes from each subnetwork. The results of these experiments demonstrate that these methods achieve better performance than most that perform clustering on the PIN directly[26–28]. In this paper, we assume that protein complexes may be formed by proteins with similar properties, such as topological and functional properties and temporal and spatial features, and we propose a protein complex identification framework (KCluster). In KCluster, we use a K-means algorithm[30] to construct K subnetworks, and each subnetwork consists of proteins with similar properties. Then, we idenitfy protein complexes in each subnetwork, as well as those spanning two subnetworks. Considering that the proteins in a complex are usually localized in the same subcellular compartment[31], the protein complex predictions are refined using protein subcellular localization information. To evaluate the efficiency of KCluster, we applied it and nine other methods to a high-confidence yeast PIN and compared their performances.

2 Methods

In the implementation of KCluster, the K-means algorithm clusters the proteins based on their degrees of proximity. K subnetworks are then constructed based on these clusters. Protein complexes are identified from each subnetwork, also are the protein complexes spanning two subnetworks. Finally, the predicted protein complexes are filtered using protein subcellular localization information. The algorithm of the KCluster method is shown in Algorithm 1, and we describe in detail each step of the KCluster method in the following subsections.

Algorithm 1 KCluster

Input: PPI network G=(V, E), Parameter K and T_{cn}, Protein sublocalization information Inf
Output: Predicted protein complexes
Process:
1. Calculate the DC of each node in V
2. Sort the nodes in a descending order according to their DC
3. Kvertex=∅
4. Kvertex=K-means(V, K) //Kvertex=$\{V_1,\ldots,V_i,\ldots,V_k\}$
5. KG=$\{(V_1, E_1),\ldots,(V_i, E_i),\ldots, (V_k, E_k)\}$ //K subnetworks
6. for i=1; $i\leqslant K$; i++**do**
   //Identify complexes from each subnetwork S_i=(V_i, E_i)
   s=0
   for each $p\in V_i$ **do**
   $C_{i,s}$=$\{p\}$
   Expect_Neighbor=(|$N(p)$|−1)×0.5
   for each $v\in N(p)$ **do**
   if $|N(p)\cap N(v)|\geqslant$Expect_Neighbor **then**
   $C_{i,s}$= $C_{i,s}\cup\{v\}$
   s++
   C_i=$\{C_{i,j}|j$=0, …, s−1$\}$
   //Generate protein complexes spanning two Partitions
7. s=0
8. **for** i=1; $i\leqslant K$; i++ **do**
   for j=i+1; $j\leqslant K$; j++ **do**
   for n=0; $n\leqslant|C_i|$; n++ **do**
   for m=0; $m\leqslant|C_j|$; m++ **do**
   if Closeness($C_{i,n}$, $C_{j,m}$)>T_{cn} **then**
   $C_{K+1,s++}$= $\{v|v\in C_{i,n}\cup C_{j,m}\}$
9. C_{K+1}=$\{C_{K+1,j}|j$=0,..., $|C_{K+1}|\}$
10. Complex=$\{$cp|cp$\in C_1\cup C_2\cup\cdots\cup C_{k+1}\}$
    //Filter complexes based on sublocalizations, Inf=$\{S_1,\ldots,S_i,\ldots,S_{11}\}$ where S_i=$\{v|v$ is sublocalized in compartment $i\}$
11. FComplex=∅
12. **for** each cp in Complex **do**
13. Count[1…11]={0}
14. **for** each p in cp **do**
15. **for** each S_j in Inf **do**
16. **if** $p\in S_j$ **then**
17. Count[j]++
18. **for** k=1; $k\leqslant$ 1; k++ **do**
19. **if** Count[k]/|cp|>0.9 **then**
20. FComplex=FComplex $\cup\{c_p\}$
21. **break**
22. **Return** FComplex

2.1 Construction of K subnetworks

Given a protein-protein interaction network $G=(V, E)$, we divide it into K subnetworks. Firstly, we classify the nodes into K clusters. Let DC(v), as defined in Eq. (1), denote the number of node v's neighbors in the network. Based on DC of each node, the nodes can be clustered into K clusters $\{\text{Cluster}_1, \ldots, \text{Cluster}_i, \ldots, \text{Cluster}_K\}$ by the K-means algorithm[30]. By mapping cluster Cluster$_i$ to G, we can construct a corresponding subnetwork S_i, denoted as $S_i = (V_i, E_i)$, where $V_i = \{v\|v \in V \cap \text{Cluster}_i\}$ and $E_i = \{(u, v)|(u, v) \in E, u \text{ and } v \in V_i\}$. Let $G^K = \{S_1, \ldots, S_i, \ldots, S_K\}$ denote the set of K subnetworks of G. The method used to construct G^K consists of 5 steps, as follows.

$$\text{DC}(v) = |N(v)|, \text{ where } N(v) = \{u|(u, v) \in E\} \quad (1)$$

Step 1: Initialize K centers $(\mu_1, \ldots, \mu_i, \ldots, \mu_K)$. We assume that the DCs of all nodes in V are calculated and the nodes are sorted by DC in descending order. The initial value μ_i of Cluster$_i$ is calculated using Eq. (2), where $x_{\max}$ and $x_{\min}$ denote the maximum and minimum DC values of all the nodes in V, respectively.

$$\mu_i = \frac{(x_{\max} - x_{\min}) \times (i-1)}{K} + x_{\min} \quad (2)$$

Step 2: Assign each node to the cluster whose center yields the least within-cluster sum of squares. Let $X^{(i)}$ denote the DC of node i and $c^{(i)}$ denote the ID of the cluster to which node i will be assigned, calculated as shown by Eq. (3). For node i, we calclate the squares of the difference between $X^{(i)}$ and the centers $(\mu_1, \ldots, \mu_i, \ldots, \mu_K)$. Then the $c^{(i)}$ of node i is assigned by the ID of the cluster having the smallest square of the difference between $X^{(i)}$ and its center, as calculated using Eq. (3).

$$c^{(i)} = \arg \min_j ||X^{(i)} - \mu_j||^2 \quad (3)$$

Step 3: Update K centers, based on Eq. (4). For Cluster$_j$ $(j = 1, \ldots, K)$, update the value μ_j with the average value of the DC of nodes that belong to Cluster$_j$.

$$\mu_j = \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\}X^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}} \quad (4)$$

Step 4: Repeat Steps 2 and 3 iteratively to update the centers of the K clusters, until the values of the K centers no longer change.

Step 5: Construct K subnetworks. Map the nodes in Cluster$_i$ $(i = 1, \ldots, K)$ to the original network $G = (V, E)$ to construct a subnetwork $S_i = (V_i, E_i)$, where $V_i = \{v|v \in V \cap \text{Cluster}_i\}$ and $E_i = \{(u, v)\|(u, v) \in E, u \text{ and } v \in V_i\}$.

2.2 Identify protein complexes from each subnetwork

For a subnetwork S_i, a randomly selected node v in V_i is initialized as a cluster C. Let $N(S_i, v)$ denote the neighbor of protein v in S_i. The expected number of

common neighbors of v, ENC(S_i, v), is defined as half the total number of neighbors, which can be calculated as shown in Eq. (5). For each protein u in $N(S_i, v)$, if the number of common elements between $N(S_i, u)$ and $N(S_i, v)$ is not less than ENC(S_i, v), u can be added to C. When all the proteins in $N(S_i, v)$ have been visited, C is considered as a predicted protein complex.

$$\mathrm{ENC}(v) = 0.5 \times (|N(v)| - 1) \tag{5}$$

2.3 Identify protein complexes spanning two subnetworks

Some protein complexes may span two subnetworks, where some proteins are in one subnetwork and other proteins are in another. We assume that the protein complexes spanning two subnetworks have the following characteristics: proteins in the same subnetwork are closely linked, and proteins from two subnetworks are also closely linked. Then, the protein complexes spanning two subnetworks can be identified by combining the closely linked protein complexes from different subnetworks. The closeness of two protein complexes is defined based on the interactions between the proteins from the two complexes, which can be calculated as shown in Eq. (6). For two protein complexes C_i and C_j, which are identified as being from different subnetworks, if their closeness is greater than a given threshold T_{cn}, they will be combined to generate a new protein complex.

$$\mathrm{Closeness}(C_i, C_j) = \frac{|E(C_i, C_j)|}{|C_i| \times |C_j|} \tag{6}$$

where $E(C_i, C_j) = \{(u, v) | u \in C_i, v \in C_j, (u, v) \in E\}$ denotes the interactions between nodes from C_i and C_j.

2.4 Refine protein complexes

Proteins in the same complex are usually localized in the same subcellular compartment[31]. Therefore, KCluster uses protein subcellular localization information to refine the predicted protein complexes. The protein subcellular localization information used in this paper was extracted from the COMPARTMENTS database[32]. The subcellular compartments in a cell are generally classified into the following 12 categories: (1) *Chloroplast*, (2) *Endoplasm*, (3) *Cytoskeleton*, (4) *Golgi*, (5) *Cytosol*, (6) *Lysosome* (or *Vacuole*), (7) *Mitochondrion*, (8) *Endosome*, (9) *Plasma*, (10) *Nucleus*, (11) *Peroxisome*, or (12) *Extracellular*, where Chloroplasts only exist in plant cells[33]. Since the protein subcellular localization information is incomplete, if the proportion of the number of proteins localized in a subcellular compartment out of the total number of proteins in the complex is greater than a threshold T_{s}, the predicted protein complex is kept. In KCluster, the T_{s} threshold is set as 0.9, which means that each predicted complex contains more than 90% co-localized proteins.

3 Results and Discussion

To validate the effectiveness of the KCluster method, we compared its prediction performance with that of nine other methods on a highly reliable yeast PIN. These nine methods include density-based algorithms (DPClus[8], IPCA[9], CMC[6], and ClusterOne [10]), the hierarchical clustering algorithm HC-PIN[22], random-walk-based algorithms (MCL[11, 12], RRW[13], and WPNCA[14]), and the algorithm based on core-attachment structure (CoreMethod[24]).

We constructed the highly reliable PIN used in this paper with Protein-Protein Interactions (PPIs), whose reliability scores were not less than 0.95 in the comprehensive PIN constructed by Yong et al.[34] The comprehensive yeast PIN[34] was integrated with PPI data from multiple data sources in which each PPI was associated with a reliability score. These multiple data sources can be classified into three categories. The first category contains physical PPIs extracted from BioGRID[35], IntAct[36], and MINT[37], whose PPIs were scored using a topological function known as the Iterative AdjustCD[6]. Protein pairs that do not directly interact but have shared neighbors were also scored. The second category is predicted functional association data obtained from the STRING database[38], wherein there is a functional association score for each predicted association between two proteins. The third category is the co-occurrence of proteins or genes in the PubMed literature and each protein pair (u, v) is scored by the Jaccard similarity of the sets of papers in which u and v appear.

We obtained 408 manually annotated yeast protein complexes from papers published in the journal *Nucleic Acids Research*[39], and these were used as benchmark data for comparison. Given a threshold T_{os}, we considered a predicted complex (Pc) and a known complex (Kc) to be a match if their overlapping score OS (Pc, Kc) is not less than T_{os}[2]. A perfect match, OS (Pc, Kc) = 1, indicates that the known complex is identical to the predicted complex. Sn is

defined as the fraction of the known complexes that are matched by the predicted complexes of all the known complexes[2]. Sp is the fraction of the predicted complexes that match the known complexes out of the total number of predicted complexes[2]. The f-measure is a comprehensive metric that combines Sn and Sp[2]. Usually, the number of matched known protein complexes (denoted as MKC), Sn, Sp, and the f-measure values of each method are compared when T_{os} is set to be 0.2.

3.1 Parameter analysis

The KCluster method has two parameters, K and T_{cn}. K is the pre-defined number of clusters for the K-means algorithm. T_{cn} is the threshold for the closeness of the protein complexes, and a new protein complex spanning two subnetworks is generated if the closeness of the two protein complexes from different subnetworks is greater than T_{cn}. In this subsection, we analyze the influence of parameters of different values on the KCluster performance.

In the K-means algorithm, the number of clusters must be pre-defined. For most situations, the most appropriate number K of clusters for a given data set is not known beforehand. To analyze the influence of different values of K on the prediction performance, for simplicity, we ran KCluster with K varying over the range [2, 12] and $T_{cn} = 1$.

For each K in the range [2, 12], Table 1 lists the number of predicted protein complexes (PC), the number of matched known protein complexes (MKC), the *Sensitivity* (Sn), the *Specificity* (Sp), the f-measure, and the number of perfect matches (Perfect Match). From Table 1, we find that when K becomes larger, the PC, MKC, Sn, Sp, and f-measure values decrease, which is obvious in the range [2, 5]. The reason for this is that when K becomes larger, there are more subnetworks and the size of each subnetwork may decrease. Consequentially, the number of neighbors of a protein in a subnetwork will decrease, which will make the predicted number of protein complexes smaller than the number of the known protein complexes. Therefore the prediction accuracy may be affected by an increase in K. When $K \geqslant 6$, the KCluster PCs slowly decrease, so do the values of Sn, Sp, and the f-measure. This demonstrates that the KCluster prediction performance is similar when $K \geqslant 6$, and the classification of many nodes tends to be stable.

If the closeness of two protein complexes from two subnetworks is greater than a given threshold T_{cn}, a new protein complex will be generated. To investigate the influence of different values of T_{cn} on the KCluster performance, we ran KCluster with T_{cn} varying over the range [0.1, 1] and K varying over the range [2, 5].

Figure 1 shows plots of the values of PC, MKC, f-measure, and Perfect Match of KCluster with T_{cn} varying from $[0.1, 1]$ when $K = 2, 3, 4$ and 5, respectively. As shown in Fig. 1a, it is easy to see that the KCluster PC is very large when T_{cn} is small, since many protein complexes that span two subnetworks are generated. When T_{cn} increases, the KCluster PC is reduced sharply, because a greater closeness is demanded to generate a protein complex spanning two subnetworks. When T_{cn} is in the range [0.8, 1],

Table 1 Comparison of KCluster prediction performance when K varies over the range [2, 12].

KCluster(K, $T_{cn} = 1$)	PC	MKC	Sn	Sp	f-measure	Perfect Match
KCluster(2, 1)	1768	315	0.9090	0.5255	0.6659	75
KCluster(3, 1)	1546	283	0.8575	0.4864	0.6207	84
KCluster(4, 1)	1493	291	0.8563	0.4668	0.6042	82
KCluster(5, 1)	1479	290	0.8518	0.4584	0.5960	82
KCluster(6, 1)	1327	285	0.8170	0.4137	0.5493	66
KCluster(7, 1)	1282	286	0.8132	0.4142	0.5488	64
KCluster(8, 1)	1231	282	0.8065	0.4265	0.5579	56
KCluster(9, 1)	1228	282	0.8065	0.4275	0.5588	56
KCluster(10, 1)	1226	288	0.8116	0.4217	0.5550	54
KCluster(11, 1)	1210	287	0.8045	0.4116	0.5446	54
KCluster(12, 1)	1231	287	0.7990	0.3907	0.5248	54

Notes: PC, MKC, Perfect Match, Sn, and Sp denote the number of predicted protein complexes, the number of matched known protein complexes, the number of perfect matches, the Sensitivity, and Specificity, respectively. The MKC, Sn, Sp, and f-measure values are calculated with $T_{os} = 0.2$.

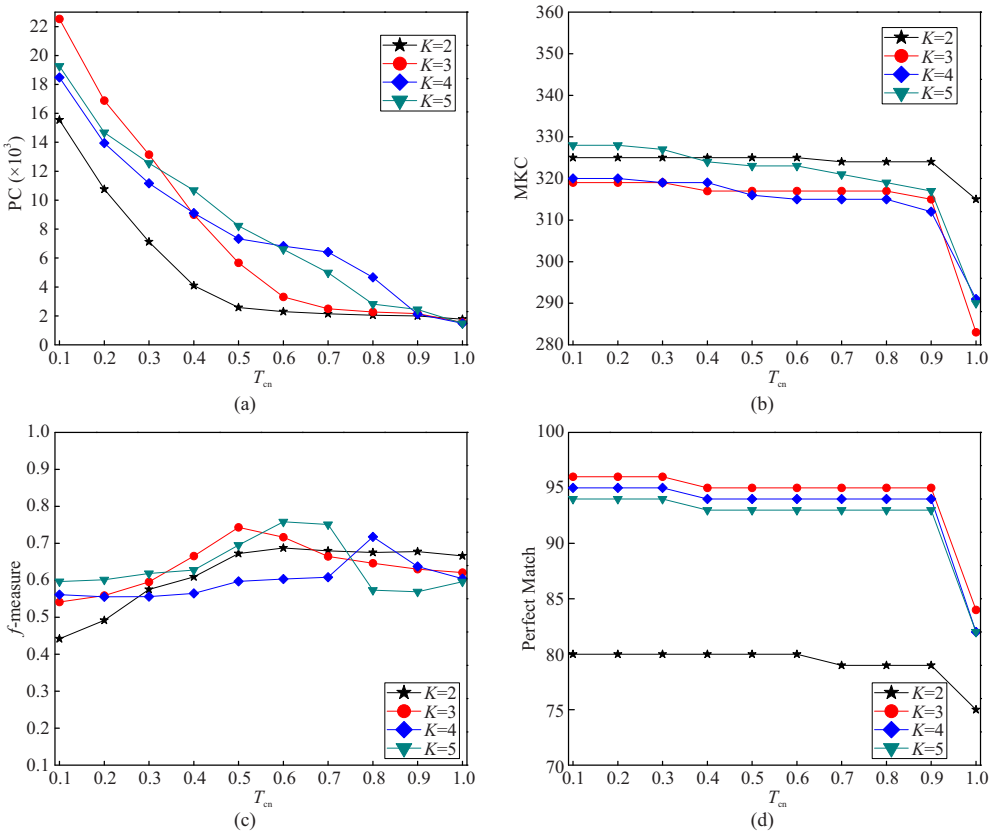

Fig. 1 KCluster prediction performance with different values of T_{cn}, when K= 2, 3, 4, and 5.

the KCluster PC is relatively stable, indicating that the closeness between the predicted complexes from different subnetworks is usually lower than 0.8. From Figs. 1b and 1d, we can see that the MKC and Perfect Match values of KCluster with K = 2, 3, 4, and 5 are relatively stable when T_{cn} is in the range [0.1, 0.9], while the MKC and Perfect Match values of KCluster decrease slightly when T_{cn} = 1. This indicates that there are a small number of protein complexes spanning two subnetworks. In Fig. 1c, for a certain K, the KCluster f-measure rises at first, reaches its maximum value, and then declines with the increase of T_{cn}. The reason for this is that when T_{cn} increases, the KCluster PC decreases sharply and the MKC is relatively stable.

3.2 Comparison with known complexes

To compare KCluster with the other algorithms, we set the KCluster parameters K and T_{cn} to be 2 and 0.9, respectively. We set the parameters of the other nine methods to the values recommended by their corresponding papers, as shown in Table 2.

Table 3 lists the PCs, the average size of the protein complexes (denoted as Av. Size), the MKCs, and the Perfect Matches. 324 known protein complexes were identified by KCluster, of which 79 known protein complexes were perfectly matched, which is much higher than the number realized by the other algorithms. Table 3 compares the Sn, Sp, and f-measure values of each algorithm. It is evident that the KCluster Sn is just slightly lower than that of the IPCA algorithm, while the KCluster Sp and f-measure values are higher than those of the other nine methods.

The MKC indicates the ability of an algorithm to predict correct protein complexes. Figure 2 shows

Table 2 Parameter settings for each method.

Algorithms	Parameters
KCluster	K=2, T_{cn}=0.9
ClusterOne	Minimum_size=2
CoreMethod	Default
HC-PIN	threshold_lambda=1.0, threshold_size=2
IPCA	S=2, P=2, T=0.6
CMC	min_deg_ratio=1, min_size=2, overlap_thres=0.5, merge_thres=0.5
DPClus	cp=0.5, dn=0.9
MCL	inflation=2.0
WPNCA	threshold_lambda=0.3, threshold_size=2
RRW	r=0.7, max=200, min=2, overlap=0.2, lambda=0.6

Table 3 Comparison of prediction performance of different methods.

Method	PC	Av. Size	MKC	Perfect Match	Sn	Sp	f-measure
KCluster(2,0.9)	1995	12.1	324	79	0.9268	0.5338	0.6774
ClusterOne	877	6.43	278	31	0.6649	0.2941	0.4079
CoreMethod	1014	4.94	284	39	0.6702	0.2485	0.3625
HC-PIN	453	7.22	191	36	0.3955	0.3134	0.3497
IPCA	2377	15.36	362	47	0.9505	0.3723	0.5350
CMC	936	6.54	324	28	0.7660	0.2938	0.4247
DPClus	968	4.69	329	44	0.7780	0.2861	0.4184
MCL	706	6.18	225	29	0.4902	0.2492	0.3305
WPNCA	1674	32.22	278	14	0.8600	0.4772	0.6139
RRW	1073	2.65	264	45	0.6139	0.2134	0.3167

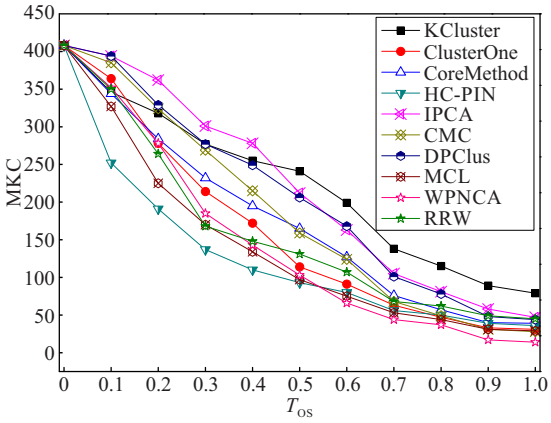

Fig. 2 MKC of different algorithms with respect to different T_{os} threshold values.

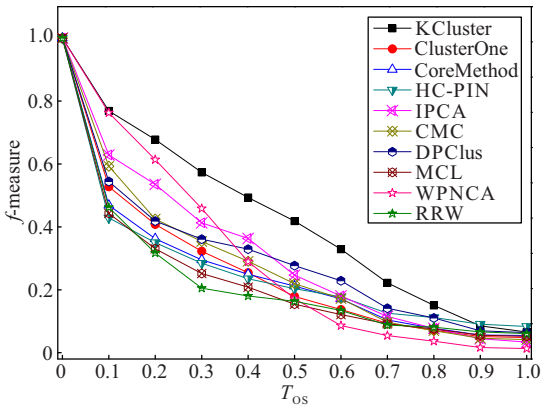

Fig. 3 f-measure of different algorithms with respect to different values of T_{os} in the range [0, 1].

the MKC of each algorithm with the value of T_{os} varying over the range [0, 1]. With the increase of T_{os}, which demands a higher degree of matching between predicted and known protein complexes, the MKC values of all the methods decrease. When T_{os} is in the range [0.5, 1], the KCluster MKC is obviously higher than that of the other methods. For example, when $T_{os} = 0.5$, 240 known protein complexes are identified by KCluster, while only 114, 165, 93, 212, 159, 206, 97, 103, and 131 known protein complexes are identified by the ClusterOne, CoreMethod, HC-PIN, IPCA, CMC, DPClus, MCL, WPNCA, and RRW algorithms, respectively. This demonstrates that KCluster can accurately identify a greater number of protein complexes.

Figure 3 compares the f-measure values of different algorithms, when T_{os} is in the range [0, 1]. The f-measure of the WPNCA algorithm is slightly higher than that of KCluster when $T_{os} = 0.1$. With the increase of T_{os}, although the f-measure of all the methods decreases, the KCluster f-measure is obviously higher than that of other methods in most cases.

3.3 Distribution of perfect matches

If OS(Pc, Kc) = 1, this means that the predicted complex Pc perfectly matches a known complex Kc. By analyzing the size distribution of the perfect matches of each algorithm, we can determine the advantages of the algorithm and its prediction accuracy. 79 known protein complexes are perfectly matched by the KCluster predicted protein complexes, while only 31, 39, 36, 47, 28, 44, 29, 14, and 45 of the known protein complexes are perfectly matched by the predicted protein complexes of ClusterOne, CoreMethod, HC-PIN, IPCA, CMC, DPClus, MCL, WPNCA, and RRW, respectively.

The size distribution of the perfect matches of each algorithm is shown in Table 4. KCluster contains perfect matches for some sizes that cannot be predicted by other nine methods. The KCluster perfect matches have 11 sizes, of which the minimum size is 2 and the maximum is 12. For other methods, there are 8 different sizes at most. Furthermore, the number of perfect matches identified by KCluster is larger than those identified by

Table 4 The size distribution of perfect matches of each algorithm.

Method	Number of perfect matches of each size											Total number of perfect matches
	2	3	4	5	6	7	8	9	10	12	15	
KCluster	31	13	14	6	4	2	5	1	1	1	1	79
CoreMethod	15	11	6	2	2	0	2	0	1	0	0	39
DPClus	14	13	8	2	3	1	2	0	0	0	1	44
HC-PIN	16	8	7	1	2	1	1	0	0	0	0	36
MCL	13	6	5	3	1	0	1	0	0	0	0	29
WPNCA	0	5	5	2	1	0	0	0	1	0	0	14
RRW	20	7	10	3	2	1	2	0	0	0	0	45
IPCA	19	8	8	3	4	0	3	0	1	1	0	47
CMC	7	11	4	2	2	0	1	0	1	0	0	28
ClusterOne	10	7	5	3	2	1	2	0	1	0	0	31

Notes: A perfect match means the a known complex is totally identical with a predicted complex, with OS(Pc, Kc)=1. The size of a perfect match is the number of proteins in the known complex or predicted complex. The size of perfect matches from all the methods is in the range of [2, 15].

the other nine methods.

3.4 Discussion

KCluster provides a framework to identify protein complexes based on K subnetworks, which are constructed using the K-means algorithm. In this section, we apply another protein feature to K-means clustering and discuss the resulting KCluster prediction performance.

Wang et al.[40] proposed a centrality measure based on an edge clustering coefficient, referred to as NC. NC considers both the centrality of a node and the relationship between it and its neighbors, and is determined by the sum of the edge clustering coefficients of the interactions connecting it to its neighbors, defined as in Eq. (7). This measure can be considered as another protein feature. We compared the KCluster prediction performance for NC and DC, respectively.

$$\mathrm{NC}(v) = \sum_{u}^{N(v)} \frac{|N(v) \cap N(u)|}{\min(|N(v)|, |N(u)|)} \tag{7}$$

As shown in Table 5, two properties of the KCluster prediction performance are compared, when K ranges from [2, 5] and T_{cn}=0.9. We can see that KCluster for NC performs well, also outperforming the other nine algorithms, with respect to the f-measure and Perfect Match. However, there are some differences in the KCluster performance with respect to two different properties. When K becomes large, the KCluster f-measure for NC is higher than that for DC. The KCluster PC for NC is always less than that for DC. The KCluster Perfect Match for DC is more than that for NC, for the same values of K and T_{cn}. Therefore, other protein properties can also be used in KCluster to identify protein complexes.

4 Conclusion

In this paper, we proposed a framework, known as KCluster, to identify protein complexes based on K subnetworks, which are constructed by employing the K-means algorithm. Protein complexes are identified from each subnetwork, so are the protein complexes

Table 5 Comparison of the prediction performance of KCluster with two different properties used in K-means algorithm.

	Property	PC	MKC	Sn	Sp	f-measure	Perfect Match
KCluster(2, 0.9)	DC	1995	324	0.9268	0.5338	0.6774	79
KCluster(3, 0.9)		2156	315	0.9174	0.4795	0.6299	95
KCluster(4, 0.9)		2118	312	0.9151	0.4891	0.6375	94
KCluster(5, 0.9)		2443	317	0.9171	0.4126	0.5691	93
KCluster(2, 0.9)	NC	1798	325	0.9195	0.5278	0.6706	77
KCluster(3, 0.9)		1912	321	0.9192	0.5183	0.6628	86
KCluster(4, 0.9)		1941	302	0.9051	0.5208	0.6612	79
KCluster(5, 0.9)		2089	309	0.9119	0.4906	0.6380	84

Notes: PC, MKC, Perfect Match, Sn, and Sp denote the number of predicted protein complexes, the number of matched known protein complexes, the number of perfect matches, the Sensitivity, and the Specificity, respectively. MKC, Sn, Sp, and f-measure are calculated with T_{os}=0.2.

spanning two subnetworks. The results show that KCluster achieves higher values for Sn, Sp, and the f-measure than the other nine methods. Furthermore, the number of perfect matches predicted by KCluster is significantly higher than that of the other nine methods. In addition, other protein properties can also be used in KCluster for K-means clustering, such as NC, to identify protein complexes.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 61232001, 61379108, and 61472133).

References

[1] W. Peng, M. Li, L. Chen, and L. Wang, Predicting protein functions by using unbalanced random walk algorithm on three biological networks, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, doi:10.1109/TCBB.2015.2394314.

[2] 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, pp. 2–28, 2003.

[3] V. Spirin and L. A. Mirny, Protein complexes and functional modules in molecular networks, *Proceedings of the National Academy of Sciences*, vol. 100, no. 21, pp. 12123–12128, 2003.

[4] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[5] 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.

[6] G. Liu, L. Wong, and H. N. Chua, Complex discovery from weighted ppi networks, *Bioinformatics*, vol. 25, no. 15, pp. 1891–1897, 2009.

[7] J. Wang, B. Liu, M. Li, and Y. Pan, Identifying protein complexes from interaction networks based on clique percolation and distance restriction, *BMC Genomics*, vol. 11, no. Suppl 2, p. S10, 2010.

[8] 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, pp. 207–219, 2006.

[9] M. Li, J. Chen, J. Wang, B. Hu, and G. Chen, Modifying the DPClus algorithm for identifying protein complexes based on new topological structures, *BMC Bioinformatics*, vol. 9, no. 1, pp. 398–413, 2008.

[10] 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.

[11] S. Van Dongen, Graph clustering by flow simulation, Ph.D. dissertation, University of Utrecht, The Netherlands, 2000.

[12] 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.

[13] K. Macropol, T. Can, and A. K. Singh, RRW: Repeated random walks on genome-scale protein networks for local cluster discovery, *BMC Bioinformatics*, vol. 10, no. 1, pp. 283–292, 2009.

[14] W. Peng, J. Wang, B. Zhao, and L. Wang, Identification of protein complexes using weighted PageRank-Nibble algorithm and core-attachment structure, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 12, no. 1, pp. 179–192, 2015.

[15] B. Zhao, J. Wang, M. Li, and F. X. Wu, Detecting protein complexes based on uncertain graph model, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 3, pp. 486–497, 2014.

[16] J. Wang, J. Zhong, G. Chen, M. Li, F.-X. Wu, and Y. Pan, Clusterviz: A cytoscape app for cluster analysis of biological network, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 12, no. 4, pp. 815–822, 2015.

[17] M. Li, J. Wang, and J. E. Chen, A fast agglomerate algorithm for mining functional modules in protein interaction networks, in *2008 International Conference on BioMedical Engineering and Informatics*, vol. 1, pp. 3–7, 2008.

[18] H. Shen, X. Cheng, K. Cai, and M.-B. Hu, Detect overlapping and hierarchical community structure in networks, *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 8, pp. 1706–1712, 2009.

[19] M. Girvan and M. E. Newman, Community structure in social and biological networks, *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[20] F. Luo, Y. Yang, C.-F. Chen, R. Chang, J. Zhou, and R. H. Scheuermann, Modular organization of protein interaction networks, *Bioinformatics*, vol. 23, no. 2, pp. 207–214, 2007.

[21] N. Pržulj, D. A. Wigle, and I. Jurisica, Functional topology in a network of protein interactions, *Bioinformatics*, vol. 20, no. 3, pp. 340–348, 2004.

[22] J. Wang, M. Li, J. Chen, and Y. Pan, A fast hierarchical clustering algorithm for functional modules discovery in protein interaction networks, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 3, pp. 607–620, 2011.

[23] A.-C. Gavin, P. Aloy, P. Grandi, R. Krause, M. Boesche, M. Marzioch, C. Rau, L. J. Jensen, S. Bastuck, B. Dümpelfeld, et al., Proteome survey reveals modularity of the yeast cell machinery, *Nature*, vol. 440, no. 7084, pp. 631–636, 2006.

[24] H. C. Leung, Q. Xiang, S. M. Yiu, and F. Y. Chin, Predicting protein complexes from ppi data: A core-attachment approach, *Journal of Computational Biology*, vol. 16, no. 2, pp. 133–144, 2009.

[25] 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.

[26] M. Li, X. Wu, J. Wang, and Y. Pan, Towards the identification of protein complexes and functional modules by integrating PPI network and gene expression data, *BMC Bioinformatics*, vol. 13, no. 1, pp. 109–113, 2012.

[27] X. Tang, J. Wang, B. Liu, M. Li, G. Chen, and Y. Pan, A comparison of the functional modules identified from time course and static PPI network data, *BMC Bioinformatics*, vol. 12, no. 1, pp. 339–353, 2011.

[28] J. Wang, X. Peng, M. Li, and Y. Pan, Construction and application of dynamic protein interaction network based on time course gene expression data, *Proteomics*, vol. 13, no. 2, pp. 301–312, 2013.

[29] J. Wang, X. Peng, Q. Xiao, M. Li, and Y. Pan, An effective method for refining predicted protein complexes based on protein activity and the mechanism of protein complex formation, *BMC Systems Biology*, vol. 7, no. 1, pp. 28–39, 2013.

[30] J. MacQueen, Some methods for classification and analysis of multivariate observations, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[31] J. R. Hutchins, Y. Toyoda, B. Hegemann, I. Poser, J.-K. Hériché, M. M. Sykora, M. Augsburg, O. Hudecz, B. A. Buschhorn, J. Bulkescher, et al., Systematic analysis of human protein complexes identifies chromosome segregation proteins, *Science*, vol. 328, no. 5978, pp. 593–599, 2010.

[32] J. X. Binder, S. Pletscher-Frankild, K. Tsafou, C. Stolte, S. I. O'Donoghue, R. Schneider, and L. J. Jensen, COMPARTMENTS: Unification and visualization of protein subcellular localization evidence, *Database*, 2014, doi: 10.1093/database/bau012.

[33] K. C. Chou and Y. D. Cai, Using functional domain composition and support vector machines for prediction of protein subcellular location, *Journal of Biological Chemistry*, vol. 277, no. 48, pp. 45765–45769, 2002.

[34] C. H. Yong, G. Liu, H. N. Chua, and L. Wong, Supervised maximum-likelihood weighting of composite protein networks for complex prediction, *BMC Systems Biology*, vol. 6, no. Suppl 2, p. S13, 2012.

[35] C. Stark, B. J. Breitkreutz, A. Chatr-Aryamontri, L. Boucher, R. Oughtred, M. S. Livstone, J. Nixon, K. Van Auken, X. Wang, X. Shi, et al., The BioGRID interaction database: 2011 update, *Nucleic Acids Research*, vol. 39, no. Suppl 1, pp. D698–D704, 2011.

[36] S. Kerrien, B. Aranda, L. Breuza, A. Bridge, F. Broackes-Carter, C. Chen, M. Duesbury, M. Dumousseau, M. Feuermann, U. Hinz, et al., The IntAct molecular interaction database in 2012, *Nucleic Acids Research*, vol. 40, no. D1, pp. D841–D846, 2011.

[37] L. Licata, L. Briganti, D. Peluso, L. Perfetto, M. Iannuccelli, E. Galeota, F. Sacco, A. Palma, A. P. Nardozza, E. Santonico, et al., MINT, the molecular interaction database: 2012 update, *Nucleic Acids Research*, vol. 40, no. D1, pp. D857–D861, 2012.

[38] D. Szklarczyk, A. Franceschini, M. Kuhn, M. Simonovic, A. Roth, P. Minguez, T. Doerks, M. Stark, J. Muller, P. Bork, et al., The STRING database in 2011: Functional interaction networks of proteins, globally integrated and scored, *Nucleic Acids Research*, vol. 39, no. Suppl 1, pp. D561–D568, 2011.

[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.

[40] J. Wang, M. Li, H. Wang, and Y. Pan, Identification of essential proteins based on edge clustering coefficient, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 4, pp. 1070–1080, 2012.

Xiaoqing Peng received the BS and MS degrees from Central South University, China, in 2009 and 2012, respectively. She is currently a PhD candidate at Central South University, China. Her research interests include genomic data analysis, dynamic protein network construction, and systems biology. She has published more than 10 papers in international journals and conferences.

Xiaodong Yan received the BS degree from Central South University, China, in 2013. He is currently a master student at Central South University, China. His research interests include genomic data analysis and sequence assembly.

Jianxin Wang received the BS and MS degrees in computer engineering from Central South University, China, in 1992 and 1996, respectively, and the PhD degree in computer science from Central South University, China, in 2001. He is the vice dean and a professor in School of Information Science and Engineering, Central South University, China. His current research interests include algorithm analysis and optimization, parameterized algorithm, bioinformatics, and computer network. He has published more than 150 papers in various international journals and refereed conferences. He is a senior member of the IEEE.