A Core-Attach Based Method for Identifying Protein Complexes in Dynamic PPI Networks

Jiawei Luo(✉), Chengchen Liu, and Hoang Tu Nguyen

School of Information Science and Engineering, Hunan University,
Changsha 410082, China
luojiawei@hnu.edu.cn

Abstract. Indentifying protein complexes is essential to understanding the principles of cellular systems. Many computational methods have been developed to identify protein complexes in static protein-protein interaction (PPI) network. However, PPI network changes over time, the important dynamics within PPI network is overlooked by these methods. Therefore, discovering complexes in dynamic PPI networks (DPN) is important. DPN contains a series of time-sequenced subnetworks which represent PPI at different time points of cell cycle. In this paper, we propose a dynamic core-attachment algorithm (DCA) to discover protein complexes in DPN. Based on core-attachment assumption, we first detect cores which are small, dense subgraphs and frequently active in the DPN, and then we form complexes by adding short-lived attachments to cores. We apply our DCA to the data of S.cerevisiae and the experimental result shows that DCA outperforms six other complex discovery algorithms, moreover, it reveals that our DCA not only provides dynamic information but also discovers more accurate protein complexes.

Keywords: Clustering · Protein complexes · Dynamic PPI networks · Core-attachment

1 Introduction

Detecting protein complexes in available PPI networks is an important and challenging task in the post-genomic era. Protein complexes are molecular aggregations of proteins assembled by multiple PPIs. They are key molecular entities to perform cellular functions. For example, complex "RNA polymerase II" transcribes genetic information into messages for ribosomes to produce proteins [1].

Up to now, many computational methods have been proposed to detect complexes in static PPI networks. Bader et al. [2] presented an algorithm MCODE, its a local-searched method which relies on the topological structure of the PPI network. Altaf-UI-Amin et al. [3] proposed a complex discovery method called DPClus which based on the combination of density and peripheral proteins to mine densely connected subgraphs. Moreover, the core-attachment concept has been proposed to identify complexes. Gavin et al. [4] illustrated the protein

T. Cao et al. (Eds.): PAKDD 2015, Part II, LNAI 9078, pp. 228–239, 2015.
DOI: 10.1007/978-3-319-18032-8_18

complex generally contain a core and attachments. The core is a small group of proteins with high degree of functional similarity, its the heart of a complex. While attachments are several peripheral neighbors of a core that assist their core to perform subordinate functions, which are often short-lived. Wu et al. [1] proposed the COACH algorithm which defined the core vertices among the neighborhood graphs, this method added attachments into the cores to form protein complexes.

However, all these methods discussed above only consider PPI networks as a static graph and overlook the dynamics inherent within them. In fact, PPI networks are varying with time and space. Therefore, understanding the dynamics of PPI networks is important to further understand molecular systems. Tang et al. [5] used gene expression data construct DPN by splitting the static PPI network into a series of time-sequenced subnetworks. In the framework of DPN, Li et al. [6] proposed a new framework to identify protein complexes and functional modules in DPN. Li et al. [7] discovered a novel method to identify dynamic complexes that integrate PPI network and gene expression data. All these efforts have made significant progress in protein complex discovery. However, only a few of these algorithms can both achieve high accuracy and capture the dynamic topology structure of DPN.

The protein complex consists of two parts in this paper: frequently active core and almost short-lived attachments. So our DCA operates in two phases: it first detects protein-complex cores and then identifies protein complexes by including attachment into cores. We compare our DCA with six competing complex discovery algorithm: DFM-CIN [6], COACH [1], ClusterOne [8], MCL [9], MCODE [2] and SPICI [10], which including the clustering method on the same DPN (DFM_CIN) and core-attachment method (COACH). Experiment results based on core analysis, F-measure, Coverage rate and functional enrichment show that our DCA performs better than these algorithms and can efficiently acquire the dynamic features of complexes.

2 Method

The static PPI network is generally considered as an undirected graph G(V, E), where a vertex in vertex set V represents a protein and an edge in edge set E represents an interaction between two proteins.

The dynamic PPI network (DPN) is constructed from static PPI network, which containing n time-sequenced subnetworks denoted as $\{D_1, D_2, \ldots, D_n\}$, as reported by Tang et al [5]. In each subnetwork of DPN, all the proteins and interactions activate at the same time, eg., subnetwork D_i is modeled as(V_i, E_i) where V_i represents the protein set and E_i represents the interaction set in the i^{th} subnetwork.

Our DCA is differing from the previous core-attachment method. We redefine the proteins in one core are not only highly connected with each other but also simultaneously occur at multiple subnetworks. Figure1 shows an example to illustrate the cores in DPN. While, attachments have much interact with cores and often short-lived. We first introduce some related definitions below.

For a protein v, its active subnetwork set can be abstracted into $Protein\_actives(v) = \{i, j, \ldots, k, \ldots\}$, where $i, j, \ldots, k$ denote the corresponding subnetworks that v appeared in. For a core S, $Core\_actives(S) = \{i, j, \ldots, h, \ldots\}$ represents the subnetwork set that S active in. Where $i, j, \ldots, h$ denote the corresponding subnetworks that the whole vertices of S are completely appeared in, it can be acquired by computing the intersection of all the proteins' $Protein\_actives()$, that is,

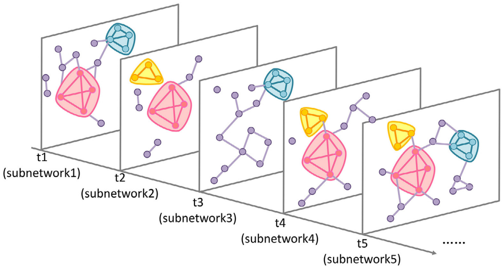

Fig. 1. Protein-complex cores in dynamic PPI networks: The nodes in pink, yellow, and blue color represent different cores in DPN. The pink core appears 4 times in DPN; the yellow and blue cores active 3 times in DPN; the purple nodes denote the remainder proteins in DPN.

$$Core\_actives(S) = \cap_{v \in S} Protein\_actives(v) \tag{1}$$

2.1 Complex Cores Mining

Based on the definition of DPN and complex cores, we assume n is the total number of subnetworks and $m(1 < m \leq n)$ is the least number of subnetworks where a core must be appeared in. In this paper, we define complex core in DPN should satisfy following four constrains with considering its topological structure and dynamic properties: (1) it's a dense subgraph of the PPI; (2) the core should active in no less than m subnetworks; (3) a core should include at least k proteins; (4) every two cores have no common proteins.

Algorithm 1 illustrates the detailed procedure on detecting cores in DPN. Before clustering, we should initialize the proteins in line 1-3. For each protein $v \in V$, we first calculate $Protein\_actives(v)$ in line 1-2, which is the set of subnetworks that v appeared in, eg., if $v \in V_i$, we then put i into the set $Protein\_actives(v)$, we iteratively traverse all the subnetworks in DPN to obtain $Protein\_actives()$ of each vertices. Next, we calculate the local clustering coefficient (CC) of v in graph G in line 3, CC(v) quantifies how close the v's neighbors are to being a clique or complete graph. The CC(v) is defined as the number of edges between v's neighbors, divided by the maximum number of edges that might potentially include in v's neighbors [11]:

$$CC(v) = \frac{\sum_{u,w \in N_v} e(u,w) | e(u,w) \in E}{d_v \times (d_v - 1)/2} \tag{2}$$

Where N_v is the set of all v's neighbors. It is obvious that protein with high CC value more tends to be involved in the core, and has a higher priority to be considered as a seed.

After initialization, all the vertices are queued into Q in descending order in terms of their CC in line 4. The first unused vertex v in Q is selected as a seed to expand a new probable complex core S in line 5-15. When we expand the seed v, we should first collect the New_N_v of v in line 7, New_N_v are the core's candidate proteins and consist of v's direct neighborhoods that are still in Q. Then we will calculate the closeness(cl) between S and each vertex $u \in New\_N_v$ in line 9, the closeness function [3] is given as follows:

$$cl(u, S) = \frac{E_{uS}}{d_S \times |V_S|} \tag{3}$$

Where E_{uS} is the number of edges that connect vertex u to core S; $|V_S|$ is the number of vertices in core S; d_S is the density of core S which is formed in the equation (4):

$$d_S = \frac{2 \times |E_S|}{|V_S| \times (|V_S| - 1)} \tag{4}$$

Algorithm 1. Cores Mining

Input:

$G = (V, E)$: static PPI network
$DPN = \{D_1, D_2, \ldots, D_n\}$: n time-sequenced dynamic PPI subnetwork
α : closeness threshold for expanding cores
m: the threshold of subnetworks number that a core active in
k : the threshold of vertices number in each core

Output:

CS: the set of cores in DPN
$Protein\_actives(v_1), Protein\_actives(v_2), \ldots, Protein\_actives(v_p)$: subnetwork set of each protein active in

```
 1. for $i = 1 : n$ do
 2.       for each vertex v in $V_i$, $Protein\_actives(v) = Protein\_actives(v) \cup \{i\}$
 3. for each protein $v \in V$, compute $CC(v)$
 4. sort proteins into queue Q in descending order by CC
 5. for $v \leftarrow Q$   //the first vertex $v$ in $Q$ is selected as a seed to expand core $S$
 6.       $S = \{v\}$   // initialize $v$ as a singleton core $S$
 7.       $New\_N_v = \{u \mid u \in N_v \wedge u \in Q\}$
 8.       while $New\_N_v \neq \phi$
 9.             for all the $u_i \in New\_N_v$, compute $cl(u_i, S)$
10.             if $max_{u_i \in New\_N_v} cl(u_i, S) \geq \alpha$
11.                   if $comNetwork(u_i, S) \geq m$, then add $u_i$ into $S$
12.                   remove $u_i$ from $New\_N_v$
13.             else break    //stop expanding core $S$
14.       if number of proteins in $S \geq k$
15.             remove all proteins of $S$ from $Q$, and put $S$ into CS
```

A higher cl value of a neighbor indicates that it is part of the core while a lower cl indicates that it's part of the periphery. Next we choose the neighbor u_i with maximum cl, if $cl(u_i, S)$ is smaller than a prefixed threshold α, we will stop expanding S. Otherwise, we should determine whether u_i has no less than m common subnetworks with proteins in S, as $comNetwork(u_i, S) \geq m$, and

$$comNetwork(u_i, S) = \cap_{w \in S} Protein\_actives(w) \cap Protein\_actives(u_i) \quad (5)$$

If u_i shares m common subnetworks with core S, add u_i into core S. Next, u_i is removed from New_N_v to prevent recalculated in line 10-13. We will repeatedly add neighbor to S until all the vertices in New_N_v is removed in line 8-13.

Once the preliminary core S is formed, we need to judge whether S includes at least k proteins, if it is, put S into final core set CS, and remove the whole vertices of S from Q to avoid being included into any other cores in line 14-15. Another round of expanding is performed until Q is empty in line 5-15, and output CS.

2.2 Protein Complexes Formation

Considering that cores in DPN have been generated, we should select attachments to cores to construct complexes. As attachments are often short-lived, we will detect attachments on each subnetwork of DPN respectively. The description of forming protein-complexes is shown in Algorithm 2.

Algorithm 2. Complexes Formation

Input:

$DPN = \{D_1, D_2, \ldots, D_n\}$: n time-sequenced dynamic PPI subnetwork

CS: the set of cores in DPN

$Protein\_actives(v_1), Protein\_actives(v_2), \ldots, Protein\_actives(v_p)$: subnetwork set of each protein active in

Output:

DC: the set of complexes in DPN

```
1. for each $S \in CS$
2.       computer $Core\_active(S)$
3.       for each $i \in Core\_active(S)$   //form a complex $C_i$ of S in $D_i$
4.             $C_i \leftarrow S$   //initialize core S as a complex $C_i$ in the $i^{th}$ subnetwork
5.             compute neighborhood proteins $N_{S\_i}$ of S in $D_i$
6.             for each $u \in N_{S\_i}$
7.                   if $E_{uS} \geq 0.5 \times |V_S|$ then
8.                         put $u$ into $C_i$   //select $u$ as an attachment of core S
9.       $C = \cup_{i \in Core\_actives(S)} C_i$, and put C into DC
10. filtering DC
```

For each core $S \in CS$. First, we need to calculate $Core\_actives(S)$ in line 2, which is the subnetwork set that S appeared in. And then, for every subnetwork D_i with $i \in Core\_Active(S)$, we add attachments to S to construct temp_complex C_i in line 3-8. The final complex C is made up of all the temp_complexes in corresponding subnetworks in line 9. When we choose attachments of core S, we based on the idea of majority rule that neighbor vertices interacting with no less than half of the proteins in the core S will be selected as attachments [1] in line 6-8. Although our cores are non-overlapped, the complexes detected by DCA may be overlapped as they could have common attachments. So we need to filter completely overlapped complexes, and output the filtered complexes of DPN in line 10. Its obvious that attachments may be active in one or several subnetworks and usually short-lived.

The time complexity is $O(cV^3)$ of algorithm 1 and $O(ncV^2)$ of algorithm 2 in the worst case. Where V is the number of whole vertices in PPI, c is the number of vertices in a core and n is the number of subnetworks. As $n, c \ll V$, the time complexity of our DCA is approximating $O(V^3)$.

3 Experiments and Results

3.1 Datasets and DPN Construction

We performed our method on two different yeast PPI networks, including DIP [12] and Krogan [13] data. The DIP data consist of 4930 proteins and 17201 interactions, while Krogan contain 3581 proteins and 14076 interactions. The gene-expressing profiles of S.cerevisiae were retrieved from Ref. [14] with the accession number GSE3431, there are 4851 genes involved in DIP and 3509 genes in Krogan. GO data was downloaded from Ref.[15]. For evaluating our identified complexes, the benchmark set consists of 428 complexes[16], from three source: (I)MIPS, (II)Aloy et al. and (III)SGD database based on GeneOntology(GO) annotations.

Previous studies [5,6] have shown that integrating gene expression profiles with the PPI networks can acquire efficient DPN. So we construct DPN as Tang et al have done in Ref. [5]. Considering that GSE3431 covers three successive cell cycles and each cycle includes 12 time points, the average expression value of gene at the same time point for three cycles is used as its expression value at the given time point. We normalize the expression values of each gene to make the values of genes range from 0 to 1, and use a proper threshold value 0.3 to screen gene products at each time point. Finally, we create the DPN based on these filtered expression values and obtain 12 time-sequenced subnetworks.

For the sake of evaluating our algorithm DCA, we compared it with six competing clustering algorithms: DFM-CIN[6], COACH[1], ClusterOne[8], MCL[9], MCODE[2] and SPICI[10]. DFM-CIN is a functional module detecting algorithm which performs on the same DPN. The others are all well-known complex discovery algorithms.Their values of the parameters are selected from those recommended by the authors.

3.2 Core Analysis

As core is the key functional unit of protein complex, we will analyze the biological similarity of our cores on DIP data. The GO annotations are used to evaluate the GO functional similarities of cores, complexes, DPN and PPI network. Two interacting proteins can have a similarity score based on their GO terms. Here, functional similarity between two proteins is calculated by the method in [17]. We sum up the similarity of all interactions in each component using three sub-ontologies (BP, CC, MF) of GO, and then average the overall similarity. Table 1 shows the average similarity of each component detected by our DCA and COACH algorithms respectively, as COACH is developed based on core-attachments and achieves an excellent performance among current algorithms.

Table 1. Average similarity of interactions involved in Cores, Complexes and PPI data on DIP

(a)Extracted by DCA

Interactions	BP	CC	MF
In DCA cores	0.335	0.451	0.233
In DCA complexes	0.243	0.447	0.193
In DPN	0.117	0.244	0.112
In PPI network	0.115	0.242	0.111

(b)Extracted by COACH

Interactions	BP	CC	MF
In COACH cores	0.241	0.436	0.185
In COACH complexes	0.190	0.440	0.155
In PPI network	0.115	0.242	0.111

Table 1(a) shows the interactions within cores in DCA achieve the highest similarity on DIP dataset, no matter which GO domain (BP,CC,MF) they are. The GO similarity scores are declined orderly in these four components, which suggests the cores have higher degree of functional similarities, and can be seen the biological hearts of protein complexes. From Table 1(b) we can see that, although cores identified by COACH is of highly biological similarities, their GO similarity score is much less than that of our DCA, which also indicates our DCA is better than COACH for producing high biological significance cores.

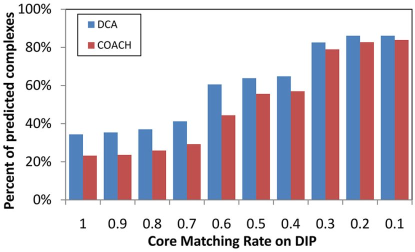

Fig. 2. The percentage of cores discovered by DCA and COACH with the CMR threshold from 1.0 to 0.1 on DIP

To further evaluate the quality of cores detected by our DCA, we quote the Core Matching Rate(CMR) [7] to measure the quality of complexes cores, which is defined as:

$$CMR(C) = max\Big(\frac{|C \cap K_i|}{|C|}\Big), \quad K_i \in K \tag{6}$$

Where K is the known benchmark complexes set [16]. $|C \cap K_i|$ denotes the number of proteins of core C included in one known complex; When a core C is completely included in a known complex K_i, $CMR(C) = 1$. Figure 2 shows the comparison results with respect to different CMR ranging from 1.0 to 0.1 on DIP. From Figure 2 we can see that as the threshold of CMR changing from 1.0 to 0.1, the percentage of DCA cores remains higher than that of COACH cores, which indicates that our DCA can produce more accurate cores than COACH method.

3.3 Functional Enrichment Analysis

To evaluate the biological enrichment and functional relevance of identified complexes, the functional homogeneity P-value [3] is applied. Accordingly, a predicted complex with a low P-value indicates it achieves a high statistical significance. The complex with corrected P-value of less than 0.01 [1] is considered significant. The proportion of significant complexes over the predictions can be used as an evaluation for assess the overall performance of various algorithms. Table 2 shows the comparison results obtained from six algorithms on DIP and Krogan datasets respectively.

Table 2. Functional enrichments of the identified complexes detected by DCA and other algorithms on DIP and Krogan datasets

Dataset	Algorithms	Identified complexes	% of significant complexes	Average P-values	Significant complexes (P)			
					< E-15	E-15 to E-10	E-10 to E-5	E-5 to E-0.01
DIP	DCA	381	94.23%	1.63E-04	30.45%	21.26%	25.46%	17.06%
	DFM_CIN	395	74.94%	3.56E-04	20.3%	9.1%	25.3%	20.3%
	COACH	746	87.67%	3.02E-04	19.6%	14.7%	27.6%	25.7%
	ClusterOne	343	67.64%	2.25E-04	10.8%	11.1%	26.5%	19.2%
	MCL	1246	30.74%	8.21E-04	2.6%	2.6%	8.5%	17%
	MCODE	59	89.83%	1.36E-04	18.7%	18.7%	37.3%	15.3%
	SPICI	583	53.52%	9.11E-04	5.8%	5.5%	15.6%	26.6%
Krogan	DCA	240	94.17%	1.55E-04	43.33%	16.67%	17.5%	16.67%
	DFM_CIN	358	75.42%	2.58E-04	21.23%	10.61%	26.26%	17.32%
	COACH	570	87.89%	3.85E-04	22.46%	10.53%	28.07%	26.84%
	ClusterOne	225	78.67%	4.42E-04	14.67%	12.89%	28.44%	22.67%
	MCL	834	37.89%	9.17E-04	3.12%	19.06%	12.47%	19.06%
	MCODE	50	94%	1.04E-04	24%	20%	36%	14%
	SPICI	383	59.53%	8.27E-04	7.83%	6.01%	18.02%	27.68%

From Table 2, it is easy to see that the percentage of significant complexes predicted by our DCA achieves the highest in the fourth column on both DIP and Krogan datasets. Moreover, the average P-values of DCA is much smaller than other algorithms except MCODE, the percentage of complexes produced by DCA with P-values less than E-15 is much higher than that of other algorithms, especially higher than DFM_CIN which is perform on DPN as well. This indicates that complexes predicted by DCA are quite accurately and have good functional enrichments.

3.4 F-measure and Coverage Rate

In order to estimate the performance of protein complexes discovered by DCA, two comprehensive evaluation methods called F-measure and Coverage rate (CR)[1] are used. F-measure is the harmonic mean of Precision and Recall. Precision measures how many correct predictions that matched real complex, Recall measures how many real complexes that matched predicted complex. CR evaluates the amount of proteins in the real complexes that can be covered by the predicted complexes. Generally, high F-measure and CR values indicate that the prediction has good efficiency.

Table 3. The precision and recall results of various algorithms on DIP and Krogan datasets

	DIP		Krogan	
Algorithms	Precision	Recall	Precision	Recall
DCA	0.546	0.437	0.704	0.348
DFM_CIN	0.387	0.4	0.492	0.418
COACH	0.382	0.582	0.421	0.449
ClusterOne	0.347	0.367	0.453	0.322
MCL	0.17	0.598	0.176	0.46
MCODE	0.525	0.143	0.56	0.105
SPICI	0.226	0.488	0.272	0.416

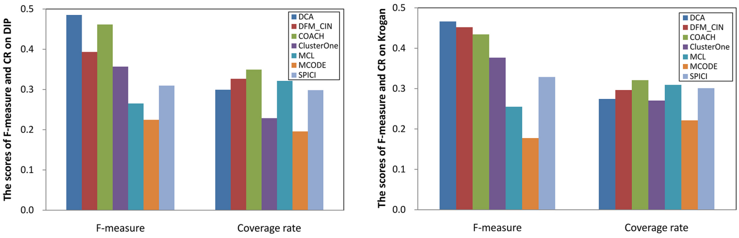

Fig. 3. The performance comparison for DCA and other algorithms on F-measure and Coverage rate on DIP(left) and Krogan(right) datasets

The basic informations for prediction by various algorithms on DIP and Krogan datasests are presented in Table 3, the precision of our DCA achieves the highest on two datasets showing DCA can identify precise complexes. In Figure 3 we can see that DCA achieves the best performance on F-measure. In detail, on DIP dataset, the F-measure of SCAIA is 48.5%, which is 2.4% higher than the second one COACH. For the number of proteins in DPN is less than that in static PPI network [6], the CR of complexes detected by DCA couldn't achieve a rather high value, but its better than ClusterOne and MCODE. All above results demonstrate that our algorithm can obtain good performance, and complexes detected by DCA match quite well with benchmark complexes.

3.5 An Example of Protein-Complex

To further reveal the results obtained by our algorithm, we display one of our protein complex that generated by DCA on DIP dataset. Figure 4 shows an example of *complex*#238. The biological process of *complex*#238 is "nuclear pore organization and biogenesis"(annotated in GO:0006999) with the lowest P-value=8.23E-26 which is carried out at the cellular level that results in the assembly, arrangement of constituent parts, or disassembly of the nuclear pore [18]. It contains 10 proteins and all of them are participating in the mechanism of nuclear pore organization.

As shown in Figure 4, the core of *complex*#238 contains 5 proteins that simultaneously active in the $2^{th}, 3^{th}, 7^{th}, 8^{th}, 9^{th}, 10^{th}, 11^{th}$ and 12^{th} subnetworks. It is a complete graph in which every pair of distinct proteins is connected by a unique edge. The core is perfectly recalled by the benchmark *complex*#50. However, three groups of attachments active in different subnetworks, eg., attachment YKL068W actives in the $8^{th}, 9^{th}, 11^{th}$ and 12^{th} subnetworks, while YJR042W is expressed in the $2^{th}, 3^{th}, 8^{th}, 9^{th}, 10^{th}, 11^{th}$ and 12^{th} subnetworks. It illustrates the dynamic properties of our complexes detected in DPN.

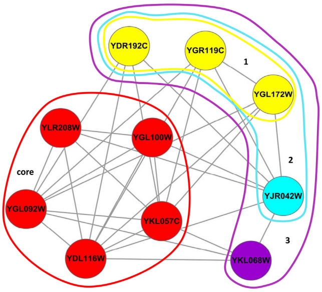

Fig. 4. An Example of protein complexes generated by DCA on DIP: 1,2,3 represent three sets of attachments in different timestamps; the red nodes represent the Core's proteins; the yellow nodes in yellow cycle denotes attachment set1; the nodes in blue cycle belong to attachment set2; the nodes in purple cycle belong to attachment set3

3.6 Effects of Parameters

We now discuss the effects of parameters of DCA in F-measure and CR, on DIP data. The parameters are the closeness threshold α for expanding cores, the number of subnetworks m that a core active in, the least vertices number k of each core.

As we can see from Figure 5(a), when α is small enough, the nodes with a very low value of *cl* are allowed to be included in the core, the core's size is much bigger than the real one, so the F-measure increases while α increases and achieves highest when α =0.8. As CR is not very sensitive to α , the optimum

α value is 0.8. Figure 5(b) reveals that as subnetwork number m increases, the number of cores decreases, thus leading to decreases in CR. Here, we set m=5, which obtains the best F-measure and comparable CR. Figure 5(c) shows that our algorithm can achieve a good balance between F-measure and CR while k=3 in our experiments.

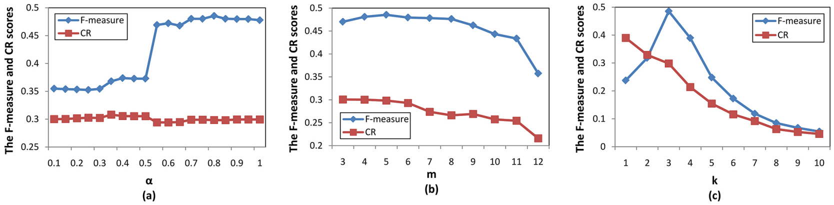

Fig. 5. The effects of α , m and k: (a)the variation of α affects the F-measure and CR; (b)the plot of the F-measure and CR for different values m; (c)the relation between k and F-measure and CR

4 Conclusion

Exploring biologically significant protein complexes are important challenges in post-genomic era. However, current complexes discovering algorithms have mainly focused on the static PPI network and failed to consider the inherent dynamics within them. Hence, we proposed a DCA algorithm to identify protein complexes in DPN. The DPN is constructed according to Tang et al' study[5]. We first detect cores which active frequently in DPN, and then extract the short-lived attachment to form complexes.

We tested our DCA on two yeast PPI data. First, we use GO similarity and CMR to analyze our core by comparing with another core-attachment method COACH. It demonstrates that proteins in our cores have high functional similarity and are highly recalled by known complexes. Second, we employ the P-value to evaluate the functional enrichment of predict complexes, the proportion of significant complexes over the predicted ones by DCA is much greater than other algorithms. The last, a lot of comparison based on F-measure and CR reveals our DCA outperforms other six algorithms, as we achieve the highest F-measure and get a comparable CR values. In conclusion, all the experiments show our DCA can not only provide a new way of detecting complexes in DPN but also identify more accurate protein complexes.

Acknowledgments. This work is supported by National Natural Science Foundation of China (Grant no.61240046) and Hunan Provincial Natural Science Foundation of China (Grant no.13JJ2017).

References

1. Wu, M., Li, X., Kwoh, C.K., Ng, S.K.: A core-attachment based method to detect protein complexes in PPI networks. J. BMC bioinformatics. **10**(1), 169 (2009)
2. Bader, G.D., Hogue, C.W.: An automated method for finding molecular complexes in large protein interaction networks. J. BMC bioinformatics. **4**(1), 2 (2003)
3. Altaf-Ul-Amin, M., Shinbo, Y., Mihara, K., Kurokawa, K.: Development and implementation of an algorithm for detection of protein complexes in large interaction networks. J. BMC bioinformatics. **7**(1), 207 (2006)
4. Gavin, A.C., Aloy, P., Grandi, P., Krause, R., Boesche, M., Marzioch, M., Superti-Furga, G.: Proteome survey reveals modularity of the yeast cell machinery. J. Nature. **440**(7084), 631–636 (2006)
5. Tang, X., Wang, J., Liu, B., Li, M., Chen, G., Pan, Y.: A comparison of the functional modules identified from time course and static PPI network data. J. BMC bioinformatics. **12**(1), 339 (2011)
6. Li, M., Wu, X., Wang, J., Pan, Y.: Towards the identification of protein complexes and functional modules by integrating PPI network and gene expression data. J. BMC bioinformatics. **13**(1), 109 (2012)
7. Li, M., Chen, W., Wang, J.: Identifying Dynamic Protein Complexes Based on Gene Expression Profiles and PPI Networks. J. BioMed Research International **2014** (2014)
8. Nepusz, T., Yu, H., Paccanaro, A.: Detecting overlapping protein complexes in protein-protein interaction networks. J. Nature methods. **9**(5), 471–472 (2012)
9. Enright, A.J., Van, D.S., Ouzounis, C.A.: An efficient algorithm for large-scale detection of protein families. J. Nucleic acids research. **30**(7), 1575–1584 (2002)
10. Jiang, P., Singh, M.: SPICi: a fast clustering algorithm for large biological networks. J. Bioinformatics. **26**(8), 1105–1111 (2010)
11. Watts, D.J., Strogatz, S.H.: Collective dynamics of ‘small-world’ networks. J. nature. **393**(6684), 440–442 (1998)
12. Xenarios, I., Salwinski, L., Duan, X.J., Higney, P., Kim, S.M.: Eisenberg D. DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. J Nucleic acids research. **30**(1), 303–305 (2002)
13. Krogan, N.J., Cagney, G., Yu, H., Zhong, G., Guo, X., Ignatchenko, A., et al.: Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. Nature **440**, 637–643 (2006)
14. Gene Expression Omnibus database. http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE3431
15. Gene Ontology Database. http://www.geneontology.org/GO.database.shtml
16. Friedel, C.C., Krumsiek, J., Zimmer, R.: Bootstrapping the interactome: unsupervised identification of protein complexes in yeast. In: Vingron, M., Wong, L. (eds.) RECOMB 2008. LNCS (LNBI), vol. 4955, pp. 3–16. Springer, Heidelberg (2008)
17. Wu, H., Su, Z., Mao, F., Olman, V., Xu, Y.: Prediction of functional modules based on comparative genome analysis and Gene Ontology application. J. Nucleic acids research. **33**(9), 2822–2837 (2005)
18. Huntley, R.P., Sawford, T., Mutowo-Meullenet P.: The GOA database: gene ontology annotation updates for 2015. J. Nucleic acids research. gku1113 (2014)