Density Based Merging Search of Functional Modules in Protein-Protein Interaction (PPI) Networks

Wei Wang and Jinwen Ma*

Department of Information Science, School of Mathematical Sciences
And LMAM, Peking University, Beijing, 100871, China
jwma@math.pku.edu.cn

Abstract. Discovering functional modules in a protein-protein interaction (PPI) network is very important for understanding the organization and function of the related biological system. The main strategy for this discovery is to translate the PPI network into a mathematical graph which can be analyzed with graph theory. In this paper, we propose a Density Based Merging Search (DBMS) algorithm to discover the complexes of a PPI graph corresponding to the functional modules in the PPI network. The DBMS algorithm starts from a single vertex with the highest density of connecting, then adds it's neighbor vertexes with the sufficiently high density of connecting one by one, and finally obtain one complex when there is no vertex to be added. The same DBMS procedure can be conducted on the rest of the vertexes in the PPI graph till all the complexes are found out. It is demonstrated by the experiments on six PPI datasets that the DBMS algorithm is efficient for discovering the complexes of a PPI network.

Keywords: PPI network; complex; vertex; connecting rate; filter.

1 Introduction

With the development of proteomics technology, a great number of large data sets of protein-protein interactions (PPI) have been accumulated from the proteomic experiments. Actually, a PPI dataset describes a PPI network within a living cell. Recent studies [1] have shown that a large PPI network consists of some complexes which correspond certain functional modules with clear biological meanings. So, it is very important and significant to reveal the structure of a large PI network and find out the existing complexes in it.

As a PPI network can be considered as an interaction graph, a complex can be mathematically represented as a densely connected subgraph in it. In this way, the discovery of the complexes is equivalent to the search of these densely connected subgraphs in the whole PPI interaction graph. To this end, several efficient algorithms have been proposed from different aspects, but they can be generally divided into the splitting and merging search categories.

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2010, LNCS 6215, pp. 634–641, 2010.

In the splitting search, the algorithm starts from the whole PPI interaction graph. It splits the graph into certain subgraphs of densely connecting, and then checks whether these subgraphs fit the conditions of a complex. If some subgraphs satisfy the conditions, we accept them as the complexes. If some subgraphs do not satisfy the conditions, we can further split them into smaller subgraphs of densely connecting, and continue this splitting until all the complexes are finally founded. A typical example of the splitting search is the *GN* algorithm [2] based on *the Hierarchical Clustering*. The *HCS* (Highly Connected Sub-graph) algorithm [3] is another example of this stream, which classifies the vertexes into subgraphs based on *the similarity* between each pair of vertexes. The *RSNC* (Restricted Neighborhood Search Clustering) algorithm [4] is also a such kind of method by implementing a cost based clustering mechanism to identify the complexes.

As for the merging search, the algorithm starts from the most densely connected vertex, extends it into a complex by adding the densely connected vertexes until no such vertex can be founded. Then, the merging search carries on the rest of vertexes and continues the same procedure till all the complexes are found. A representative of the merging search is Newman's Fast Algorithm [5], which conducts the merging process via an evaluation function Q such that a group of vertexes can be merged into a complex as long as Q has the largest increment. Inspired by Newman's work, Clauset further proposed a greedy algorithm [6] which can be applied to very large networks. Recently, Bader and Hogue proposed the MCODE (Molecular Complex Detection) algorithm [8] which utilizes the concept of vertex weighting and can identify the complexes efficiently.

In this paper, in light of the merging search we propose a Density Based Merging Search (DBMS) algorithm for discovering the complexes of a PPI graph with the help of the concept of vertex density of connecting. That is, by defining the connecting density for each vertex, we can find the densest vertex, search its neighbor vertexes with the high enough densities and finally find a complex in the graph. In the same way for the rest of the vertexes and step by step, we can find out all the complexes of the graph. It is demonstrated by the experiments on six PPI datasets that the DBMS algorithm is efficient for discovering the complexes of a PPI network. In comparison with the MCODE algorithm, the DBMS algorithm obtains a better result for discovering the complexes, but it is computationally expensive.

2 The DBMS Algorithm

2.1 The Characteristics of a Complex

A complex in a PPI graph should represent some functional module which involves a number of proteins densely connected together. That is, mathematically, a complex is certainly a densely-connecting subgraph of a graph. However, there has not been a unified definition of a complex yet. In fact, it was defined in different ways related to the complex search methods. For example, Brun [9] and Samanta [10] defined the degree of similarity between two interaction vertexes via their common neighbor vertexes, utilize these similarity degrees for clustering analysis and regard the clusters as the complexes. On the other hand, Watt et al. [11] defined the density of a graph

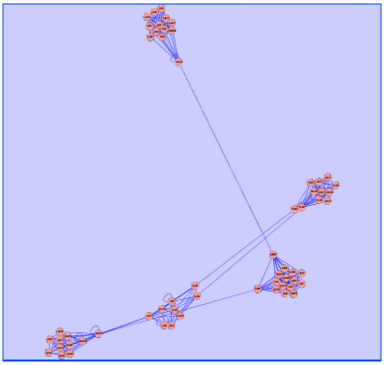


Fig. 1. The PPI graph with five clear complexes

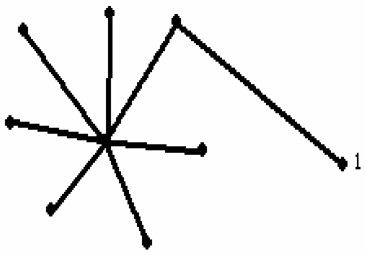


Fig. 2. The sketch of a complex

and consider a dense subgraph as a complex. For graph G with n vertexes, the edges of G will be not more than emax=n(n-1)/2, when G is an undirected graph, the edges of which have no directions. So, d=e/emax can be defined as the density of G, where e is the number of edges in G. We can define that G is dense when d is greater than a certain positive threshold.

In some specific PPI graphs, we can easily find out these dense subgraphs and accept them as the complexes. Figure 1 gives such an example. However, in general, the dense subgraphs are not separated so clearly, especially when there are thousands of vertexes in the graph.

Based on the proteomics experiments and biological experiences, we have the following characteristics of a complex: (1). No vertex belongs to two or more complexes. (2). A vertex belongs to a certain complex if it has a high degree. (3). A vertex with a low degree can be in a complex as a terminal vertex. In fact, as shown Figure 2, the most of the vertexes have one degree, but they can be in the complex. Moreover, if a vertex P belongs to a complex S, the most of the vertexes connecting to P are also in S.

2.2 The Description of the DBMS Algorithm

We begin to define the density of vertex p by

$$\rho_p = (\sum_{v \in U_p} d_v + d_p^2) / d_p , \tag{1}$$

where d_v is the degree of the vertex v, U_p is a set of vertexes that are the neighbors of p. Obviously, we can filter the vertexes with a small density since they are generally isolated and do not belong to any complex. Here, a vertex is filtered out before the complex search if its density is less than 3. After then, we try to divide the remaining vertexes into some subgraphs in order to search the complexes. Actually, we further need the concept of correlation between a vertex and a dense subgraph or a complex S which is assumed to consist of n vertexes, i.e., $|S|=n$. Given a vertex P, $d_p = m$, we hope to identify whether $P \in S$. Obviously, P belongs to S if most of P's neighbors are in S. Suppose there are k vertexes of S connecting with P, so $k \le n$ and $k \le d_p$, we define $c = k / n$ as the correlation degree of P to S.

P belongs to S if c is large enough. Assume that $P \in S$ if $c \ge \frac{b}{n}$, so that $P \in S$ if $k \ge b$. Clearly, b should increase with n, but the rate of increase should become relatively small. As we let b be proportional to $\log n$, we then get $b = \min\{\log n, f\}$, where f is a real number related with d_p to balance $\log n$ since it may be too large. In fact, if d_p is smaller than $\log n$ and $b = \log n$, P may not belong to S even if most of the edges connecting to P are in S. In practice, we can set $f = \lambda d_p$, where $\lambda \in (0,1)$.

Given an undirected connected graph G (v, e), S is a subgraph of G, and c is a constant. We consider S as a complex if for any vertex v in S, $k_v / |S| \ge c$, where k_v is the number of the neighbors of v in S, and $|S|$ is the number of vertexes in S. In this way, we propose the Density Based Merging Search (DBMS) algorithm to discover the complexes one by one as follows.

Step 1. Let $S = \emptyset$.

Step 2. Select a vertex P from G such that $\rho_p = \max_{v \in G}\{\rho_v\}$, and let $S = S \cup \{P\}$, N={The neighbors of the vertexes of S in G}, $n = |S|$.

Step 3. Select $Q \in N$ such that $\rho_Q = \max_{v \in N}\{\rho_v\}$, and have

(a). If $c_Q \ge c(n)$, then $S = S \cup \{Q\}$, update N;

(b). If $c_Q < c(n)$, then $N = N/\{Q\}$.

Step 4. If $N \ne \emptyset$, go to 3. Otherwise, if $N = \emptyset$, output S as a complex , and let $G = G - S$, filter out G. If $G = \emptyset$, stop. Else go to 1.

Since |G| is decreased as each complex S is found in a search loop, the DBMS algorithm will certainly converge in a finite iterations. As for the complexity of the DBMS Algorithm, we can consider the worst case to check whether each vertex belongs to S in N. It can be easily found that the time complexity of this step is $O(n^2)$. On the other hand, the time complexity for the filtering in the worst cases is $O(hn^2)$. Therefore, the time complexity of the DBMS Algorithm is $O(hn^3)$. Moreover, we need a memory space with the size of $O(n^2)$ to store the set G, N and S.

3 Experiment Results

3.1 The PPI Datasets

A PPI dataset consists of a group of edges connecting two proteins. In some cases, the PPI datum even represents the strength of the edge, but here, we only consider whether two proteins are connected. So, it is easy to translate the PPI data into a PPI graph where each vertex serves as a protein. Moreover, for each vertex, we can get its neighbor set, i.e., the set of all the neighbors of it.

For the experiments, we use two kinds of PPI datasets. The first kind of dataset is a simulated one as shown by Figure 1, in which there are 59 vertexes and 317 edges. The second kind of datasets are eight real-world PPI datasets downloaded from The Database of Interacting Proteins (DIP: http://dip.doe-mbi.ucla.edu). Specifically, they are Celeg20090126, Dmela20090126, Ecoli20090126, Hpylo20090126, Hsapi20090126, Mmusc20090126, Rnorv20090126, and Scere20090126.

In the experiments, we will compare the DBMS algorithm with the we will take CODE algorithm whose code is downloaded from: http://www.baderlab.org/Software/MCODE/.

3.2 Simulation Results

We begin to implement the DBMS algorithm on the simulated dataset, i.e., the PPI graph given by Figure 1. We set the base of the logarithm is the natural number e, and $\lambda = 0.5$, i.e., $f = 0.4 d_p$. The results of the complex discovery by both the DBMS algorithm and the MCODE algorithm on this simulated dataset are given in Table 1.

It can be found from Table 1 that both the DBMS and MCODE algorithms have found five complexes from the PPI graph, but those complexes of the two algorithms are quite different in the way of content as well as the number of vertexes. By checking these results with Figure 1, we have found that the complexes found by the DMBS algorithm are consistent with the five true complexes. However, the complexes found by the MCODE algorithm are not so consistent with those true complexes. Therefore, we can consider the DMBS algorithm is more efficient than the MCODE algorithm on the complex discovery in this experiment.

3.3 Experimental Results on the Real-World PPI Datasets

We further implement the DBMS algorithm on the eight real-world PPI datasets and summarize the experimental results of complex discovery in Table 2, where #vexes denotes the number of vertexes in the PPI graph corresponding to the dataset, #Rvexes denotes the number of remaining vertexes in G at the end of the algorithm, #edges is the number of edges in the PPI graph, #Redges is the number of remaining edges connecting two vertexes in G at the end of the algorithm, #complexes denotes the number of complexes found by the algorithm, and Size-LC denotes the size, i.e., the number of vertexes in the largest complex.

Table 1. The complex discovery results of the DBMS and MCODE algorithms on the simulated PPI dataset, where each column represents a complex found. The blank fill with yellow is the vertexes not be handled by MCODE, and the blank fill with blue is the vertexes which are misclustered by MCODE.

The DBMS algorithm					The MCODE Algorithm				
S_1	S_2	S_3	S_4	S_5	S_1	S_2	S_3	S_4	S_5
taf60	pap1	cdc23	pat1	rpt1	hfi1	cft1	apc11	lsm7	rpn11
ngg1	cft2	cdc16	dcp1	rpn5	taf90	ysh1	apc1	lsm2	rpn5
spt8	yk059	apc1	kem1	rpn10	ngg1	cft2	apc2	lsm6	rpn8
spt3	cft1	apc11	lsm7	rpt3	taf25	pfs2	cdc23	dcp1	rpt1
						yk05			
taf25	pta1	apc5	lsm6	rpn11	spt8	9	cdc16	lsm8	rpn10
spt20	ref2	cdc27	lsm4	rpn6	cdc23	pti1	apc5	rpn6	rpn6
tra1	fip1	apc9	lsm3	rpn3	spt20	pap1	doc1	lsm5	rpt3
						rna1			
gcn5	pfs2	cdc26	lsm2	rpn8	tra1	4	rpt1	lsm4	rpn3
taf17	yth1	apc2	lsm5	rpt6	gcn5	glc7	apc4	lsm3	rpt6
hfi1	ysh1	apc4	lsm1		spt3	yth1	cdc27	kem1	
ada2	pti1	doc1	lsm8		rna14		apc9		
taf90	glc7				spt15		cdc26		
taf61	rna14				taf61				
spt7					taf17				
spt15					ada2				
					spt7				
					taf60				
15	13	11	11	9	17	10	12	10	9

Table 2. The experimental result of complex discovery by the DBMS algorithm on the eight real-world PPI datasets

Dataset	#vexes	#Rvexes	#edges	#Redges	#complexes	Size-LC
Celeg	2643	2098	4043	2278	159	637
Dmela	7494	6403	22872	6651	513	375
Ecoli	1559	1399	7002	5769	21	1282
Hpylo	704	590	1424	601	48	60
Hsapi	1755	1176	2171	1468	141	99
Mmusc	709	372	633	364	62	17
Rnorv	237	115	199	106	22	11
Scere	4965	4309	17612	8102	230	438

Specifically, we give some typical experimental results of complex discovery in Figures 3-6, respectively. The PPI graph of the dataset Rnorv20090126 is given in Figure 3, while 22 complexes found by the DBMS algorithm are shown in Figure 4. It can be observed clearly that all the possible complexes are correctly found.

According to the experimental results on those eight real-world PPI datasets, we find that the DBMS algorithm is applicable for complex discovery. Moreover, the DBMS algorithm generally outperforms the MCODE algorithm on complex discovery, which can be demonstrated by the experimental results shown in Figures 5 for the datasets Rnorv20090126, respectively. Actually, in Figure 5, we can find that the complexes found by the DBMS algorithm are more reasonable than those found by the MCODE algorithm.

In a summary, the DBMS algorithm can be efficiently implemented to discover the complexes in a PPI graph or dataset. Moreover, it is even better that the MCODE algorithm in certain cases.

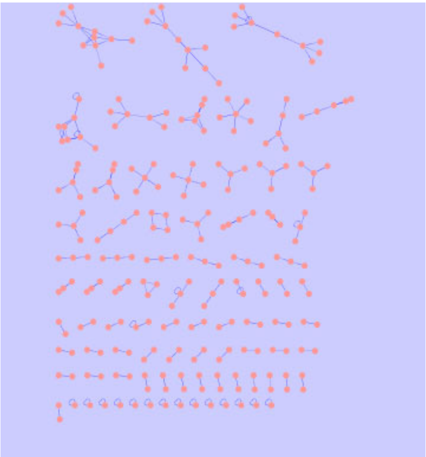


Fig. 3. The PPI graph of forRnorv20090126

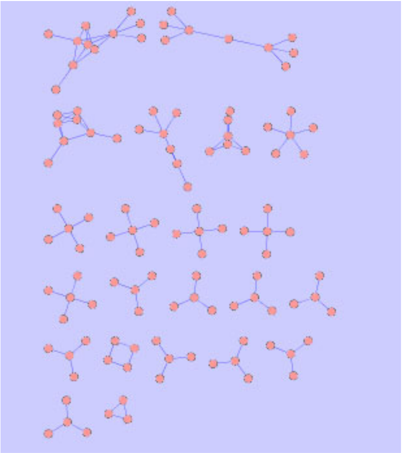
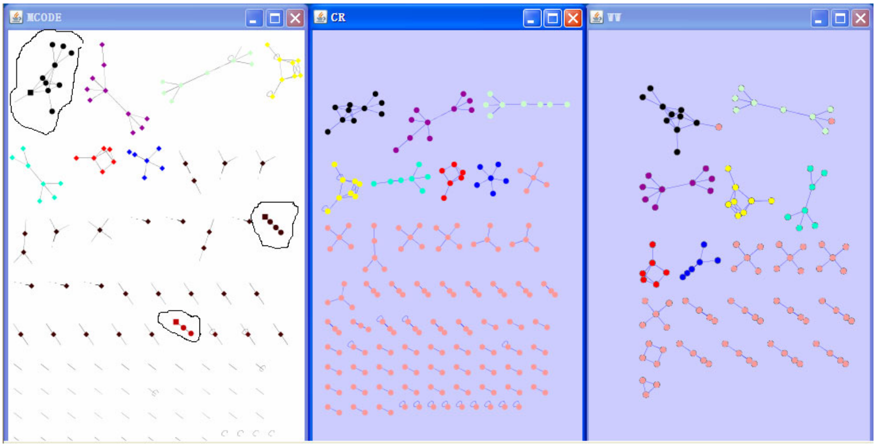


Fig. 4. The complexes found Rnorv20090126



(a) (b) (c)

Fig. 5. The comparison between the DBMS and MCODE algorithms on Rnorv20090126. (a). 3 complexes (circled clusters) found by the MCODE algorithm. (b). The true complexes given in the dataset for check. (c). The complexes found by the DBMS algorithm.

5 Conclusions

We have investigated the discovery of the functional modules or complexes in a protein-protein interaction (PPI) network or graph and have proposed the density based merging search algorithm to discover the complexes for a PPI dataset. The DBMS algorithm is a kind of merging search procedure which combines the highly connected vertexes from any vertex with the highest density step by step to form a complex. It is demonstrated by the experiments on both simulated and real-world PPI datasets that the DBMS algorithm is applicable and efficient for complex discovery and even outperforms the MCODE algorithm.

Acknowledgements

This work was supported by the Natural Science Foundation of China for Project 60771061.

References

1. Hartwell, L.H., et al.: From Molecule to Modular Cell Biology. Nature 402, C47–C52 (1999)
2. Girvan, M., Newman, M.E.J.: Community Structure in Social and Biological Networks. Proc. Natl. Acad. Sci. 99(12), 7821–7826 (2002)
3. Hartuv, E., Shamir, R.: A Clustering Algorithm Based on Graph Connectivity. Information Processing Letters 76(4-6), 175–181 (2000)
4. King, A.D., Przulj, N., Jurisica, I.: Protein Complex Prediction via Cost-based Clustering. Bioinformatics 20(17), 3013–3020 (2004)
5. Newman, M.E.J.: Fast Algorithm for Detecting Community Structure in Networks. Phys. Rev. E 69, 66133 (2004m)
6. Clauset, A., Newman, M.E.J.: Moore c: Finding Community Structure in Very Large Networks. Phys. Rev. E 70(6), 66111 (2004)
7. Palla, G., Dere'nyi, I., Farkas, l., et al.: Uncobering the Overlapping Community Structure of Complex Networks in Nature and Society. Nature 435(7043), 814–818 (2005)
8. Bader, G.D., Hogue, C.: An Automated Method for Finding Molecular Complexes in Large Protein Interaction Networks. BMC Bioinformatics 4, 2 (2003)
9. Brun, C., Chevenet, F., Martin, D., et al.: Functional Classification of Proteins for the Prediction of Cellular Function from a Protein-protein Interaction Network. Genome Biol 5(1), R6 (2003)
10. Samanta, M.P., Liang, S.: Predicting Protein Functions from Redundancies in Large-scale Protein Interaction Networks. Proc Natl. Acad. Sci. USA 100(22), 1257–11258 (2003)
11. Watts, D.J., Strogatz, S.H.: Collective Dynamics of Small-world Networks. Natur. 393, 440–442 (1998)