Dai QG, Guo MZ, Liu XY *et al.* CPL: Detecting protein complexes by propagating labels on protein-protein interaction network. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 29(6): 1083–1093 Nov. 2014. DOI 10.1007/s11390-014-1492-z

CPL: Detecting Protein Complexes by Propagating Labels on Protein-Protein Interaction Network

Qi-Guo Dai[1] (代启国), Mao-Zu Guo[1,*] (郭茂祖), Xiao-Yan Liu[1] (刘晓燕), Zhi-Xia Teng[1,2] (滕志霞) and Chun-Yu Wang[1] (王春宇)

[1] *School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China*

[2] *School of Information and Computer Engineering, Northeast Forestry University, Harbin 150040, China*

E-mail: {qiguo_dai, maozuguo, liuxiaoyan, teng_zhixia, chunyu}@hit.edu.cn

Received November 28, 2013; revised September 18, 2014.

Abstract Proteins usually bind together to form complexes, which play an important role in cellular activities. Many graph clustering methods have been proposed to identify protein complexes by finding dense regions in protein-protein interaction networks. We present a novel framework (CPL) that detects protein complexes by propagating labels through interactions in a network, in which labels denote complex identifiers. With proper propagation in CPL, proteins in the same complex will be assigned with the same labels. CPL does not make any strong assumptions about the topological structures of the complexes, as in previous methods. The CPL algorithm is tested on several publicly available yeast protein-protein interaction networks and compared with several state-of-the-art methods. The results suggest that CPL performs better than the existing methods. An analysis of the functional homogeneity based on a gene ontology analysis shows that the detected complexes of CPL are highly biologically relevant.

Keywords protein complex detection, label propagation, protein-protein interaction, graph clustering, bioinformatics

1 Introduction

A protein complex consists of a group of interacting proteins[1] that play an important role in a biological process or cellular component. A complex consisting of multiple proteins provides more information about the activities of the proteins than a pairwise interaction between two proteins. Although many biological technologies have been used to identify protein complexes, such as tandem affinity purification with mass spectrometry, they have several limitations, for example, missing transient protein complexes[2]. It is important to identify complexes from biological data with computational methods[3-4]. With the development of high-throughput experiment technologies, many protein-protein interaction (PPI) networks have been published in public databases. A PPI network can be modeled as a graph, in which the nodes are proteins and the edges are interactions, to illustrate the physical binding between the proteins at the system level. The PPI network is a significant data resource for the computational detection of complexes because complexes often correspond to dense sub-graphs in the network[2]. It should be noted that protein complexes often overlap, as a protein can participate in multiple complexes under different conditions. Detecting complexes from a PPI network can be formalized as an overlapping graph clustering problem. Many algorithms[5-8] have been proposed to detect complexes from PPI networks, which can be generally grouped into clique-based[9-11] and seed expansion[7,12-14] algorithms.

A clique is a complete sub-graph, in which all of the nodes connect with each other. CFinder is one of the most popular clique-based methods[15-16]. It assumes that a complex consists of a set of adjacent cliques and detects complexes by searching for adjacent cliques. However, as they rely heavily on a specific topological structure, most of the clique-based methods, including CFinder, are influenced dramatically by the incompleteness of and the noise in PPI networks.

Seed expansion algorithms expand an initial set of seed complexes to optimize a predefined quality function. MCODE[13] selects proteins with high weights as seeds and then expands the set by including any neigh-

Regular Paper

The work was supported by the National Natural Science Foundation of China under Grant Nos. 61271346, 61172098, and 91335112, the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20112302110040, and the Fundamental Research Funds for the Central Universities of China under Grant No. HIT.KISTP.201418.

*Corresponding Author

boring proteins with weights higher than a threshold. ClusterONE is a popular seed expansion method that iteratively adds proteins into or removes proteins from a set of initial complexes to maximize the cohesiveness function[7]. Wang and Wu proposed a new distance measurement method for expanding seed complexes[17]. Expansion and seed selection in most of the seed expansion methods generally depends on a predefined quality function or on expansion techniques.

Other technologies have also been proposed to detect complexes[18-23]. Markov clustering (MCL)[18] simulates random walking on a network. Anirban *et al.* proposed a multi-objective evolutionary approach based on semantic similarity[20]. ProRank+ uses a ranking algorithm to detect protein complexes by ordering the proteins based on their importance[21-22]. SLCP2 is a spectral approximate algorithm that enables the simultaneous identification of both dense and sparse regions in a network[23].

Although much progress has been made in identifying protein complexes from PPI networks, accurately identifying protein complexes still remains a challenge. The methods mentioned tend to find less-known protein complexes, since they may suffer from limitations such as depending on the distribution of a specific topological structure or on expanding techniques.

In this paper, we propose a novel framework that identifies complexes by propagating labels through a PPI network (CPL). In CPL, a label is used to denote the identifier of a complex. With proper label propagation, proteins in the same complex will be assigned with the same labels. It should be noted that CPL does not require any strong definitions of the interaction patterns in complexes, such as a predefined quality function or prior information of the topological structure. The experimental results show that the complexes identified by CPL are of a higher quality than those identified by several state-of-the-art methods. The complexes detected by CPL cover more known complexes and have a high functional homogeneity.

The remainder of this paper is organized as follows. Section 2 illustrates the basic idea of label propagation in CPL. Section 3 describes the proposed CPL algorithm. The experimental results are shown and discussed in Section 4. Finally, Section 5 presents some concluding remarks and future work.

2 Label Propagation for Complex Detection

The label propagation algorithm (LPA)[24] was first proposed to identify disjointed communities in social networks. In LPA, each node has only one label, denoting the community to which it may belong. Although LPA has many advantages for finding communities within a social network, it faces challenges when detecting complexes from PPI networks. For example, protein complexes generally overlap, as a protein can belong to multiple complexes. In addition, although PPI networks are available in many databases, they suffer from noise and incompleteness. Consequently, the performance of LPA is dramatically reduced when it is directly used to detect complexes.

We therefore present CPL, a novel framework of label propagation, to detect complexes in a PPI network. By allowing each protein to carry multiple labels and by using the propagating intensity, the propagation in CPL is able to handle the detection of complexes. CPL uses the interactions in a network alone to guide the progress of propagation, independent of any measure of complex quality, such as a clique or quality function, as in previous methods.

Fig.1(a) shows an example network. The propagation of CPL is illustrated by Figs. 1(b)∼1(f), in which different shapes denote different labels. The main ideas behind the proposed CPL are as follows.

1) Proteins are allowed to carry multiple labels during propagation, not just one. LPA cannot handle the overlapping between complexes, as it uses one label for each protein. By assigning each protein multiple labels, the proteins can belong to multiple complexes, such as protein 3 in Fig.1(f).

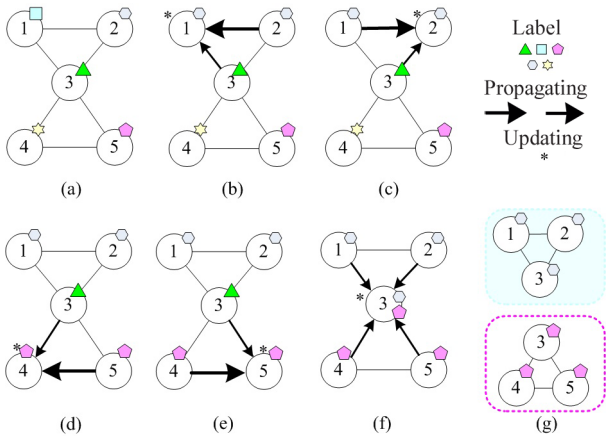


Fig.1. Illustration of propagation in CPL. Nodes denote proteins and edges denote their interactions. Different shapes on nodes represent different labels that the proteins carry. (a) An example network with five proteins, each of which is initialized with a unique label. The label propagation process proceeds from (b) to (f). (g) Two complexes sharing the overlapping protein 3.

2) When a label is propagated from one protein to another, it is assigned a certain propagating intensity. In Fig.1, the width of the arrow line denotes the intensity of the propagation. The intensity is determined on the basis of common neighbors of the interacting proteins. If two proteins have a more similar neighborhood,

then the intensity of the label propagated through their interaction is greater. A label with greater propagating intensity has a higher priority for being assigned to the target protein. This makes the propagation more robust.

3) The propagation in LPA depends on the order in which the nodes are updated. Using a different update order may yield a poorer result. Assigning protein labels according to a specific order may feasibly address this problem. As shown in the following section, it is better to update the labels of proteins with greater weights later.

As shown in Fig.1, the proteins update their label information one by one according to their order. The protein marked by * is being updated. Before protein 1 in Fig.1(b) updates, all of its neighbors first propagate a label to protein 1. Protein 1 then updates its own labels using the received labels with high intensities. After propagation, two complexes sharing the overlapping protein 3 are identified in Fig.1(g).

3 CPL Algorithm

Based on the above framework, the CPL algorithm is developed to detect complexes from PPI networks. A high-level description of the algorithm is given in Algorithm 1.

Algorithm 1. CPL

Input: PPI network N

Output: complex set C

1) Initialize the label memory for each protein in network N.
2) According to the assigning order, select a protein u whose label memory has not been updated.
3) All of the neighbors of protein u propagate a label to it with a certain intensity.
4) The label memory of protein u is assigned based on the received labels.
5) Repeat steps 2)∼4) until no protein remains.
6) Post-process and output complex set C.

In step 1), we initialize the label memory of each protein. In the propagation process of steps 2)∼5), all of the proteins determine the complexes in which they participate according to a specific order. Each protein takes part in label propagation 3) and label assignment 4). Proteins with the same label are then grouped together into a complex. The label memory, assigning order, label propagation and assignment, and post-processing are now described in detail.

3.1 Label Memory and Initialization

A protein can participate in more than one complex. A label memory that saves multiple labels and their corresponding belonging coefficients is used to represent the membership of a protein to different complexes. For protein u, the memory L_u is a set of pairs (l_i, c_i), which denotes that protein u belongs to complex l_i with a belonging coefficient c_i. If there is more than one pair in the memory, the protein may overlap multiple complexes.

Each protein is first assigned a unique label with a belonging coefficient of 1.0. Each protein is considered as an initial complex containing only itself before propagation begins.

3.2 Label Propagation Rule

After initialization, each protein determines its label memory, one by one. For each protein u, each of its neighboring proteins first propagates a label to u. If v is a neighbor of protein u, the label propagated from protein v to u is $l_{v\to u}$. This is the label with the maximum belonging coefficient in the memory of protein v. If there is more than one label with the same maximum belonging coefficient in protein v's memory, one of the labels is randomly propagated. The propagating intensity (PI) $p_{v\to u}$ of the label $l_{v\to u}$ propagated from v to u is calculated from the interaction weight as follows:

$$p_{v\to u} = \frac{|N_+(u) \cap N_+(v)|}{|N_+(u)| \times |N_+(v)|},$$

where $N_+(u)$ is the protein set consisting of u and all of its neighbors $N(u)$. The propagating intensity is higher if there are more common neighbors between the two proteins. Fewer common interacting partners yield a lower intensity. This information can be used to reduce the influence of noise in a PPI network, as false positive interactions will generally correlate with low common neighborhoods.

The protein u receives a set of labels S_u propagated from all of its neighbors:

$$S_u = \{l_{v\to u} | v \in N(u)\}.$$

As different neighbors may send the same label, we sum the propagating intensities of the same propagated label:

$$p_i = \sum_{v\in N(u)} (p_{v\to u} \times \delta(l_i, l_{v\to u})),$$

where $\delta(a, b)$ is an indicator function that equals 1 if $a = b$, and 0 otherwise.

The received labels and corresponding propagating intensities of protein u are given by P_u:

$$P_u = \{(l_i, p_i) | l_i \in S_u\}.$$

3.3 Label Assignment Rule

After propagation, the label memory of protein u is updated by exploiting the propagated labels with high

intensities. In P_u, all of the labels with propagating intensities greater than the threshold $thre_u$ are used to update the label memory of protein u:

$$L_u \leftarrow \{ \arg_{(l_i,p_i)\in l'_u} (p_i \geqslant thre_u) \}.$$

The threshold $thre_u^{\text{ave}}$ is an adaptive threshold for protein u based on the average value of p_i in P_u:

$$thre_u^{\text{ave}} = \frac{1}{|P_u|} \sum_{(l_i,p_i)\in P_u} p_i. \quad (1)$$

The average value adaptively extracts the represented labels from the received labels. In addition, the intensities of all of the propagated labels may be very low due to the noise in the network. To filter the noise, the label memory will not be updated when the adaptive threshold is smaller than a minimum value τ. The higher the minimum value, the greater the inhibition of noise. However, at the same time, it will reduce the number of the complexes. In order to make a balance, τ is set to 0.05.

3.4 Propagating Order

The propagating order is crucial to propagation. CPL updates the label information of each protein according to the protein's weight. The weight w_u of protein u is the sum of the weights of the interactions with which the protein is involved:

$$w_u = \sum_{v\in N(u)} p_{v\to u}.$$

We find that CPL using the ascending order of protein weight gives a better performance. It means that proteins with a lower weight are updated first.

3.5 Post-Processing

After all of the proteins are updated, each protein obtains the labels denoting the underlying complexes in which it may participate. Each protein is allocated to the complexes according to the saved labels in its label memory. An elementary set of complexes C with respect to the network is obtained.

It should be noted that there may be complexes that are subsets of others or are disconnected in C. A refining step is used to split disconnected complexes into a set of connected complexes and to remove any complexes that are subsets of others. Complexes in C with less than three proteins are removed. The remaining complexes in C are the final output of CPL.

4 Experimental Results

We implement the CPL algorithm① in Java. We investigate some of the strategies proposed in the algorithm. The algorithm is compared with several state-of-the-art methods, and the accuracy and the functional homogeneity of the complexes predicted by these methods are studied. Some of the putative complexes detected by CPL are discussed.

4.1 Datasets and Evaluation Methods

The CPL algorithm is tested using the widely used PPI networks of *Saccharomyces cerevisiae* (yeast) from Biology General Repository for Interaction Dataset (BioGRID)[25], the Database of Interacting Proteins (DIP)[26] and the Munich Information Center for Protein Sequences (MIPS)[27]. The details of these networks are listed in Table 1. The BioGRID network that we use contains only the physical interactions from the original version of BioGRID. These networks contain similar numbers of proteins. The BioGRID network contains the most interactions, approximately five times as many as the MIPS network.

Table 1. Three PPI Networks Used in the Experiments

Network	Date	Number of Proteins	Number of Interactions
BioGRID (Physical)	2012/08/31	5 640	59 748
DIP	2012/08/18	5 046	22 449
MIPS	2006/05/18	4 554	12 526

We use the known complexes in the CYC2008 catalogue② as our gold standard for comprehensive comparisons, which is reported in [28]. The catalogue contains 408 protein complexes validated by small-scale experiments and reported in the literature. These complexes cover 1 628 proteins. We use CYC2008 because it represents an up-to-date set of the known protein complexes in yeast. It has a better coverage of the yeast genome and is more representative than the corresponding MIPS catalogue.

We evaluate the accuracy of each predictive approach by matching the predicted complexes to the gold standard. Two evaluation metrics are used to measure the matching between the predicted and the known complexes.

1) *Precision, Recall and F-Measure* (*PRF*) *Metric.* The overlapping score *OS* between a predicted complex p and a known complex b is defined as:

①http://nclab.hit.edu.cn/CPL, Sept. 2014.

②http://wodaklab.org/cyc2008/, Sept. 2014.

$$OS(p,b) = \frac{|p \cap b|^2}{|p| \times |b|}.$$

Complexes p and b are considered as a match if $OS(p,b) \geqslant w$, where w is the matching threshold and is set to 0.2 as in the literature[5]. The predicted complex set is denoted with P and the complex set in the gold standard is B. Each complex in each set consists of a set of proteins. Let N_{cp} be the number of predicted complexes that match at least one known complex. Let N_{cb} be the number of known complexes that match at least one predicted complex. The PRF components[5] are defined as:

$$precision = \frac{N_{\text{cp}}}{|P|},$$

$$recall = \frac{N_{\text{cb}}}{|B|},$$

$$F\text{-}measure = \frac{2 \times precision \times recall}{precision + recall}.$$

2) *Composite Score.* The composite score[7] is also used to evaluate the matching between the predicted and the known complexes. The composite score is the sum of three sub-metrics, the maximum matching ratio (MMR), the fraction of matched known complexes in the gold standard (Fraction) and the geometric accuracy (Acc). The composite score is described in detail in [7].

4.2 Effectiveness of the Strategies Proposed in CPL

We investigate the use of the adaptive propagating threshold, assigning order of ascending protein weights and propagating intensity based on interaction weights in CPL. PRF and composite score are both used to test the performance.

4.2.1 *Adaptive Propagating Threshold*

We propose using the average-based propagating threshold $thre_u^{\text{ave}}$ in (1), which is an adaptive threshold for protein u. We study its effectiveness by comparing $thre_u^{\text{ave}}$ to the max-based threshold $thre_u^{\text{max}}$[29], where:

$$thre_u^{\text{max}} = (\max_{(l_i,p_i)\in P_u} p_i) \times r.$$

The parameter r ranges from 0.1 to 1.0, with an interval of 0.1. Table 2 compares the results of using the different thresholds in CPL to predict complexes from the three PPI networks.

The best results obtained using the max-based thresholds are mainly distributed from $r = 0.2$ to $r = 0.4$. Using the average-based threshold yields results similar to the best results obtained from the max-based thresholds. Although careful parameter adjustments of the max-based threshold may result in better

Table 2. Comparison of the Use of Different Propagating Thresholds in CPL Applied to the Three Networks

Network	Measure	$thre^{\text{max}}$											$thre^{\text{ave}}$
		$r = 0.0$	$r = 0.1$	$r = 0.2$	$r = 0.3$	$r = 0.4$	$r = 0.5$	$r = 0.6$	$r = 0.7$	$r = 0.8$	$r = 0.9$	$r = 1.0$	
BioGRID	Precision	0.171	0.298	0.318	0.328	*0.332*	0.321	0.322	0.293	0.288	0.293	0.312	0.388
	Recall	0.419	0.721	0.738	*0.748*	0.743	0.733	0.723	0.676	0.632	0.586	0.576	0.706
	F-measure	0.243	0.421	0.445	0.456	*0.459*	0.446	0.445	0.409	0.395	0.391	0.405	0.500
	Acc	0.464	0.609	0.658	0.688	0.706	0.709	0.715	*0.718*	0.708	0.703	0.690	0.631
	MMR	0.216	0.330	0.362	0.383	0.394	0.404	*0.411*	0.406	0.394	0.395	0.393	0.373
	Fraction	0.289	0.551	0.615	0.640	0.650	*0.657*	0.647	0.613	0.556	0.547	0.529	0.632
DIP	Precision	0.170	0.242	*0.252*	0.251	0.241	0.229	0.227	0.225	0.217	0.219	0.220	0.302
	Recall	0.485	0.623	*0.647*	0.610	0.591	0.566	0.551	0.510	0.493	0.471	0.466	0.620
	F-measure	0.252	0.348	*0.362*	0.355	0.343	0.326	0.322	0.313	0.302	0.299	0.299	0.406
	Acc	0.406	0.559	0.587	0.598	*0.604*	0.601	0.602	0.589	0.584	0.575	0.566	0.604
	MMR	0.238	0.318	*0.338*	0.337	0.337	0.330	0.332	0.326	0.317	0.314	0.306	0.363
	Fraction	0.336	0.502	*0.539*	0.522	0.525	0.517	0.493	0.473	0.451	0.424	0.409	0.566
MIPS	Precision	0.162	0.214	*0.216*	0.211	0.210	0.202	0.203	0.196	0.182	0.181	0.182	0.229
	Recall	0.407	*0.495*	0.488	0.461	0.453	0.431	0.400	0.380	0.358	0.336	0.326	0.496
	F-measure	0.231	0.298	*0.299*	0.290	0.287	0.275	0.269	0.259	0.241	0.235	0.234	0.313
	Acc	0.378	0.444	0.466	0.480	0.485	*0.492*	0.489	0.482	0.476	0.471	0.464	0.491
	MMR	0.214	0.273	*0.278*	0.276	0.271	0.265	0.265	0.260	0.256	0.249	0.242	0.297
	Fraction	0.301	*0.422*	0.417	0.409	0.400	0.377	0.365	0.336	0.309	0.294	0.282	0.452

Note: $thre^{\text{max}}$ and $thre^{\text{ave}}$ are the max-based and the average-based threshold for the whole network respectively. The values colored in red under $thre^{\text{max}}$ are the best results using the max-based thresholds. The performance of CPL using an average-based threshold is colored in blue. PRF and composite scores are both shown.

prediction, it is difficult to determine the best value of r. Using the adaptive threshold based on the average intensity has a comparatively good detection performance.

4.2.2 *Propagating Order of Ascending Protein Weights*

We study the use of the shuffle, the ascending and the descending orders of protein weights in CPL. The results are compared in Fig.2. Figs. 2(a)∼2(c) compare the PRF scores for the complexes predicted by each method in each network, and Figs. 2(d)∼2(f) compare the composite scores.

The results show that the ascending order generally performs better than the shuffle order and the descending order performs the worst. We conclude that CPL yields more accurate complexes when the label memories of proteins with smaller weights are updated earlier in the propagation. Proteins with higher weights tend to have a greater influence on the surrounding proteins. Propagation may be skewed if the higher weight proteins are updated prematurely.

4.2.3 *Propagating Intensity Based on Interaction Weight*

We study the effect of the propagating intensity on CPL's predictive ability by comparing the CPL algorithm with CPL_{uw}, which does not use the propagating intensity. In CPL_{uw}, the propagating intensities of all of the interactions in the network are set to 1. CPL and CPL_{uw} are compared in Fig.3. CPL performs better than CPL_{uw} on all three of the networks. The results suggest that using propagating intensities enhances propagation and the denser the network, the greater the effect of using the propagating intensities. Intensities based on interaction weights may reduce the effect of noise in the network.

4.3 Comparing CPL with Other Methods

We study the performance of CPL by comparing it with the state-of-the-art methods SLCP2, ProRank+, ClusterONE, MCL and CFinder. We set the parameters of these methods based on their recommendations. We cannot obtain the results for CFinder identifying complexes from the BioGRID network, as the calculation requires more memory than our 4 GB computer.

The characteristics of the complexes predicted by the methods are presented in Table 3, which shows the number of predicted complexes and the number of proteins covered by the predicted complexes. CPL detects 1 316 complexes covering 3 779 proteins from the BioGRID network, 1 192 complexes covering 3 670 proteins from the DIP network and 975 complexes covering 3 136 proteins from the MIPS network. CPL detects fe-

Fig.2. Comparison of the use of different propagating orders in CPL to detect complexes from the three networks. (a)∼(c) Comparisons of the PRF metrics. (d)∼(f) Comparisons of the composite scores. (a) and (d) are BioGRID network results. (b) and (e) are DIP network results, and (c) and (f) are MIPS network results.

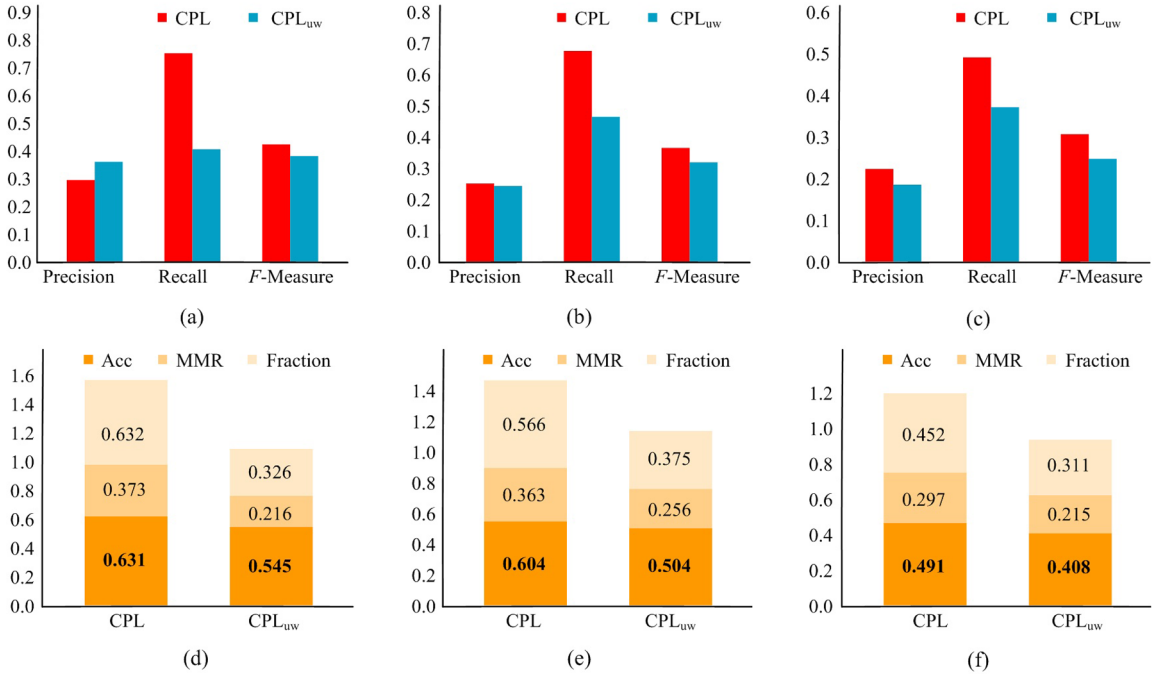


Fig.3. Comparison of the complexes predicted by CPL and CPL$_{uw}$ from the three networks. (a)∼(c) Comparison of the PRF metrics. (d)∼(f) Comparison of the composite scores. (a) and (d) are BioGRID network results, (b) and (e) are DIP network results, and (c) and (f) are MIPS network results.

wer complexes from the MIPS network than from the BioGRID network. A similar pattern is seen in the other methods, except for MCL. The BioGRID network may have a more complicated connectivity than the other networks. The complexes predicted by CFinder from the MIPS network cover 1 387 proteins, which is approximately one third of the proteins in the network and is 44% of the proteins covered by CPL. CFinder may require a more stringent clique distribution in the network than the other methods.

We also examine the overlapping proteins in the predicted complexes. Table 3 shows the number of overlapping proteins in the predicted complexes and the number of overlapping proteins in CYC2008 that are detected by the different methods. The results show that SLCP2 and MCL do not predict any overlapping proteins. CPL predicts 2 409 overlapping proteins from the BioGRID network, 1 387 from the DIP networks and 84 from the MIPS network. There are 190 overlapping proteins from the BioGRID network, 137 from the DIP network and 71 from the MIPS network and they are consistent with the overlapping proteins in CYC2008, which is more than any of the other overlapping methods. The results suggest that the proposed CPL method handles the overlapping protein problem well.

4.3.1 Accuracy of the Complexes Predicted by CPL

We investigate the matching between the complexes predicted by the different methods and the known complexes in the gold standard. As shown in Fig.4, CPL yields the best recall rates, a PRF metric, of all of the

Table 3. Characteristics of the Complexes Predicted by Various Methods from the Three Networks

Method	BioGRID				DIP				MIPS			
	#Com	#Pro	#OV^{Pred}	#OV^{Know}	#Com	#Pro	#OV^{Pred}	#OV^{Know}	#Com	#Pro	#OV^{Pred}	#OV^{Know}
CPL	1 316	3 779	2 409	**190**	1 192	3 670	1 378	**137**	975	3 136	84	**71**
SLCP2	810	3 721	0	0	783	2 980	0	0	653	2 396	0	0
ProRank+	568	4 623	4 392	179	267	1 304	525	53	235	1 079	402	21
ClusterONE	954	4 368	1 804	97	931	3 661	1 249	68	762	3 146	848	26
MCL	204	5 640	0	0	1 198	5 046	0	0	1 096	4 554	0	0
CFinder	–	–	–	–	201	2 216	329	31	178	1 387	192	23

Note: #Com: number of predicted complexes, #Pro: number of proteins covered by the predicted complexes, #OV^{Pred}: number of overlapping proteins in the predicted complexes, #OV^{Know}: number of overlapping proteins in CYC2008 that are predicted.

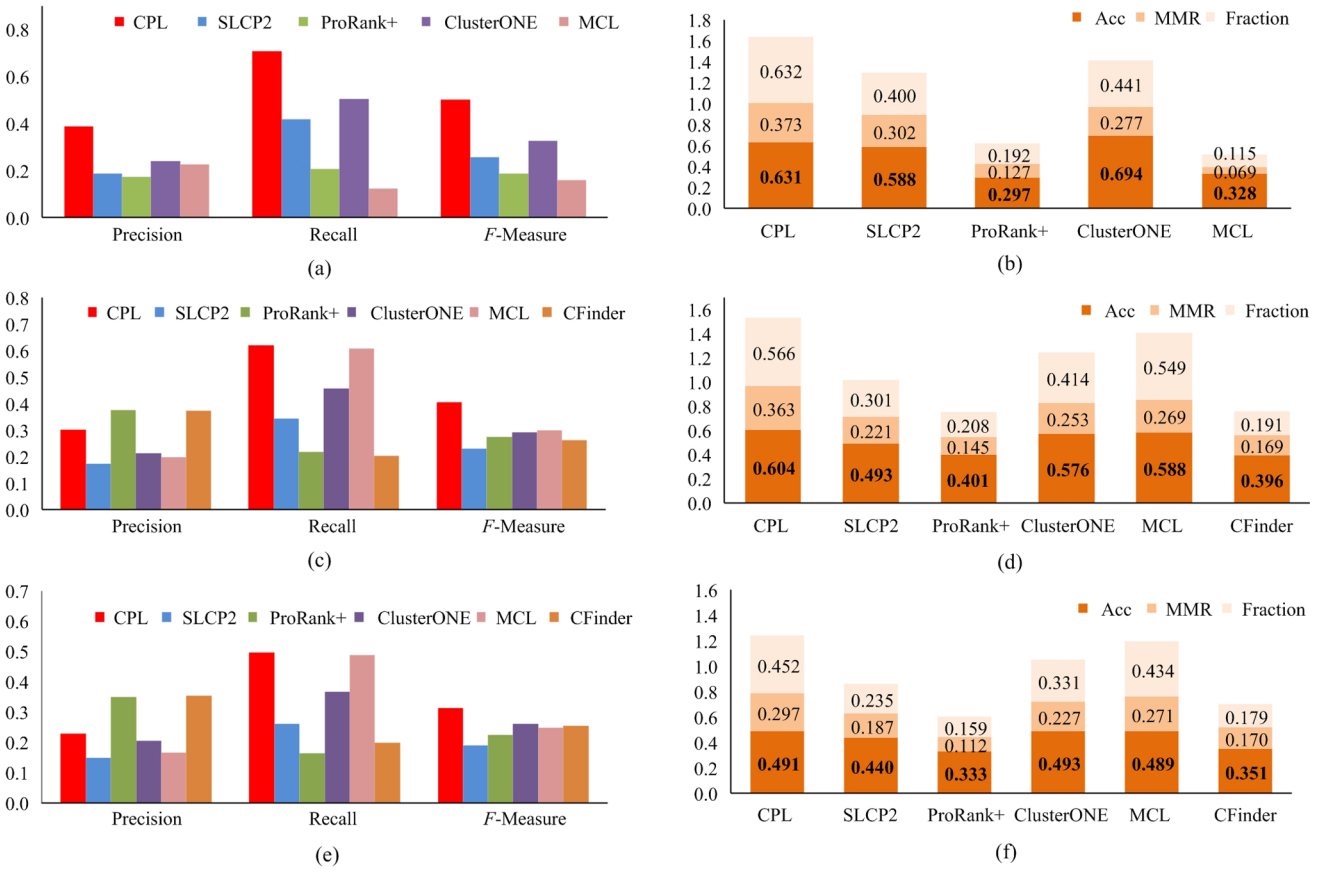


Fig.4. Comparison of CPL with other methods. (a)∼(b) Application of the methods to the BioGRID network. (c)∼(d) Application of the methods to the DIP network. (e)∼(f) Application of the methods to the MIPS network. (a), (c), (e): comparison of the PRF metrics. (b), (d), (f): comparison of the composite scores.

tested methods, indicating that it finds the most known complexes. In Fig.4(a), for example, CPL's recall is 0.706, which is almost six times that of MCL. MCL has comparative recall values when applied to the DIP and the MIPS networks, but has poor precision scores. It can also be found that CPL provides less precision scores than ProRank+ and CFinder on DIP and MIPS datasets. It may be due to that CPL could not handle the incompleteness of the dataset, as these two networks are sparser than the BioGRID one. Generally, CPL has the best F-measure scores on all three networks.

CPL also outperforms the other methods on the composite score, shown in Figs. 4(b), 4(d) and 4(f). CPL's composite score when applied to the BioGRID network is 1.636, which is approximately three times that of MCL. The results suggest that CPL can identify complexes from PPI networks with a high level of accuracy.

4.3.2 *Functional Homogeneity of the Complexes Predicted by CPL*

As the gold standard datasets are incomplete, a predicted complex that does not match any known complexes may still be valid. We therefore investigate the biological relevance of the predicted complexes on the basis of the functional homogeneity of the constituent proteins. The reason is that the proteins in a complex tend to be responsible for a specific molecular function or biological process, or are located in the same cellular compartment[5,7]. The gene ontology (GO) corpus[30] of yeast is downloaded from the *Saccharomyces* Genome database (SGD) (dated on August 11, 2010).

We use GO::TermFinder (Version 0.83)[31] to compute the p-value for each predicted complex. A predicted complex that has at least one function annotation with a p-value equal to or smaller than a threshold p is considered to have functional homogeneity. The number of complexes in a set of predicted complexes that are functionally homogeneous is used to evaluate the performance of the prediction method. Table 4 shows the number of functionally homogeneous complexes predicted by each method. We set p to 1.0E−2, 1.0E−10 and 1.0E−20 and investigate the effect on all three aspects of GO. Although CPL predicts fewer functionally homogeneous complexes than SLCP2 in some cases under $p = 1.0\text{E}{-20}$, it predicts more complexes

Table 4. Comparison of CPL with Other Methods Using Functional Homogeneity

p	Method	BioGRID			DIP			MIPS		
		MF	BP	CC	MF	BP	CC	MF	BP	CC
1.0E−2	CPL	**1 080**	**1 098**	**942**	**864**	**754**	**583**	**684**	**596**	**454**
	SLCP2	574	424	291	503	370	267	408	326	203
	ProRank+	359	260	170	193	192	153	173	180	144
	ClusterONE	381	459	332	297	375	291	306	393	310
	MCL	39	63	41	228	295	229	225	342	253
	CFinder	–	–	–	80	110	90	95	129	94
1.0E−10	CPL	**488**	**485**	**498**	**421**	**262**	**230**	**307**	**153**	**94**
	SLCP2	359	149	95	315	109	80	219	64	39
	ProRank+	165	69	52	83	55	52	56	34	33
	ClusterONE	65	112	106	53	64	76	27	52	48
	MCL	4	9	7	28	45	53	13	30	26
	CFinder	–	–	–	14	18	23	13	19	16
1.0E−20	CPL	328	**236**	**255**	**340**	**139**	**101**	**268**	**66**	26
	SLCP2	**338**	98	50	302	98	50	211	55	**28**
	ProRank+	143	36	22	61	24	18	42	7	9
	ClusterONE	20	30	43	11	23	28	2	7	6
	MCL	3	4	3	7	7	16	0	5	5
	CFinder	–	–	–	3	3	7	2	7	3

Note: MF: molecular function, BP: biological process, CC: cellular compartment.

with functional homogeneity than the other methods in most cases. The results suggest that the complexes derived by CPL are highly biologically relevant.

In summary, the results show that CPL performs well in detecting protein complexes and overlapping proteins, and that the predicted complexes of CPL are of good quality, as measured by functional homogeneity. This good performance may be due to the self-adaptive process of the propagation, which exploits the interactions in the network alone and does not need predefined descriptions of the complexes.

4.4 Putative Complexes Predicted by CPL

Some of the putative complexes predicted by CPL are shown in Table 5. The listed complexes are not reported in CYC2008. The proteins in bold are annotated with the corresponding term in the GO corpus. Each complex is given an overlapping score between the predicted complex and the set of proteins annotated by a certain GO term. Although the listed complexes are not reported in CYC2008, their overlapping scores are higher than 0.6, indicating that they are highly likely to be biologically relevant. These complexes should be investigated in further biological experiments.

5 Conclusions and Future Work

In this paper, we presented the novel framework CPL, which generates protein complexes by propagating labels through a PPI network. The main characteristic of CPL is its self-adaptive label propagation, which is independent of topological structures and quality functions. We investigated the performance of CPL when using the average-based adaptive propagating threshold, propagating intensity based on interaction weights, and assigning order based on protein weights. We also compared CPL with several state-of-the-art methods. The experimental results show that the complexes detected by CPL match more known complexes in the gold standard with a higher accuracy than the other methods, and also have better functional homogeneity. The proposed CPL algorithm offers a new way to detect complexes from a PPI network.

Although CPL performs well, it still has some limitations. Some of the predicted complexes are redundant, which may be due to false positives in the high-throughput PPI networks. One way to further improve the performance of CPL is to use the structural PPI data from X-ray crystallography, NMR spectroscopy, and other high-resolution techniques. The reason is that it could provide detailed information about the residues at an interacting interface. However, the development of accurate, complete structural PPI sets is still in its early stages. Although much structural data is available from the Protein Data Bank[32], many of the structures are monomeric and do not show native packing interactions[33]. Computational approaches, such as docking and homology-based methods, have recently been proposed for predicting structural interactions[33-34]. In the future, we will consider both structural information and graph-based network about PPIs, to further reduce the noise existing in the high-throughput PPI networks.

Table 5. Putative Complexes Predicted by CPL and Their GO Annotations

Complex	OS	Term	p-Value	Network
YPL201C, YER062C, YIL053W	1.00	Glycerol biosynthesis	3.09E−10	DIP, MIPS
YLR306W, YLR128W, YDR139C	1.00	Protein neddylation	4.63E−09	BioGRID
YOR257W, YJL019W, YNL188W	1.00	Half bridge of the spindle pole body	1.54E−09	MIPS
YFR029W, YDR160W, YJL156C	1.00	Response to an amino acid stimulus	9.26E−11	BioGRID
YNL223W, YNR007C, YBL078C, YHR171W, YLR450W	0.80	C-terminal protein lipidation	1.56E−12	BioGRID
YLL001W, YKR036C, YIL065C, YJL112W, YBL029W	0.80	Mitochondrial fission	4.65E−13	BioGRID
YLR376C, YIL132C, YHL006C, YDR078C, YLR046C	0.80	Error-free DNA repair	6.77E−13	BioGRID
YNL106C, YIL002C, YOR109W, YIR006C	0.75	Phosphoinositide 5-phosphatase	1.97E−09	MIPS
YHL003C, YKL008C, YMR298W, YNL107W	0.75	Ceramide biosynthesis	1.23E−09	MIPS
YDR078C, YHL006C, YLR376C, YLR046C	0.75	Error-free DNA repair	3.95E−09	DIP, MIPS
YDR507C, YCL024W, YKL048C, YGR021W	0.75	Septin checkpoint	1.26E−08	BioGRID
YPR145W, YGR124W, YBL039C	0.67	Asparagine synthase activity	5.63E−07	BioGRID
YLR284C, YOR180C, YGR263C	0.67	Dodecenoyl-CoA delta-isomerase	5.63E−07	DIP
YCR048W, YNR019W, YLR242C	0.67	Sterol O-acyltransferase activity	7.88E−07	MIPS
YML106W, YMR271C, YDR058C	0.67	Orotate phosphoribosyltransferase	5.63E−07	BioGRID
YLR354C, YGR043C, YIR034C	0.67	Transaldolase activity	4.50E−07	BioGRID
YJR148W, YHR208W, YHR152W	0.67	Branched chain amino acid transaminase	5.63E−07	DIP
YBL039C, YJR103W, YDR133C	0.67	CTP synthase activity	4.50E−07	MIPS
YGL253W, YFR053C, YPR042C	0.67	Fructose transport	5.52E−06	BioGRID
YER062C, YIL053W, YPL201C	0.67	Glycerol-1-phosphatase activity	6.76E−07	DIP, MIPS
YEL041W, YJR049C, YOR009W	0.67	NAD^+ kinase activity	7.88E−07	BiGRID, MIPS
YDL138W, YDL194W, YDR277C	0.67	Glucose binding	7.88E−07	BioGRID
YDL182W, YDL131W, YNL247W	0.67	Homocitrate synthase activity	5.63E−07	BioGRID
YAL054C, YLR153C, YLR049C	0.67	Acetate-CoA ligase activity	6.76E−07	MIPS

Note: Proteins in bold are annotated by the corresponding GO terms. OS: overlapping score.

References

[1] Spirin V, Mirny L A. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences*, 2003, 100(21): 12123-12128.

[2] Chen B, Fan W, Liu J *et al.* Identifying protein complexes and functional modules — From static PPI networks to dynamic PPI networks. *Briefings in Bioinformatics*, 2014, 15(2): 177-194.

[3] Geva G, Sharan R. Identification of protein complexes from co-immuno-precipitation data. *Bioinformatics*, 2011, 27(1): 111-117.

[4] Ji J, Zhang A, Liu C *et al.* Survey: Functional module detection from protein-protein interaction networks. *IEEE Knowledge and Data Engineering*, 2014, 26(2): 261-277.

[5] Li X, Wu M, Kwoh C K *et al.* Computational approaches for detecting protein complexes from protein interaction networks: A survey. *BMC Genomics*, 2010, 11(Suppl 1): S3.

[6] Wang J, Li M, Deng Y *et al.* Recent advances in clustering methods for protein interaction networks. *BMC Genomics*, 2010, 11(Suppl 3): S10.

[7] Nepusz T, Yu H, Paccanaro A. Detecting overlapping protein complexes in protein-protein interaction networks. *Nature Methods*, 2012, 9(5): 471-472.

[8] Becker E, Robisson B, Chapple C E *et al.* Multifunctional proteins revealed by overlapping clustering in protein interaction network. *Bioinformatics*, 2012, 28(1): 84-90.

[9] Chen B, Shi J, Zhang S *et al.* Identifying protein complexes in protein-protein interaction networks by using clique seeds and graph entropy. *Proteomics*, 2013, 13(2): 269-277.

[10] Habibi M, Eslahchi C, Wong L. Protein complex prediction based on k-connected subgraphs in protein interaction network. *BMC Systems Biology*, 2010, 4(1): 129.

[11] Zhang C, Liu S, Zhou Y. Fast and accurate method for identifying high-quality protein-interaction modules by clique merging and its application to yeast. *Journal of Proteome Research*, 2006, 5(4): 801-807.

[12] Altaf-Ul-Amin M, Shinbo Y, Mihara K *et al.* Development and implementation of an algorithm for detection of protein complexes in large interaction networks. *BMC Bioinformatics*, 2006, 7(1): 207.

[13] Bader G D, Hogue C W V. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 2003, 4(1): 2.

[14] Wu M, Li X, Kwoh C K *et al.* A core-attachment based method to detect protein complexes in PPI networks. *BMC Bioinformatics*, 2009, 10(1): 169.

[15] Adamcsek B, Palla G, Farkas I J *et al.* CFinder: Locating cliques and overlapping modules in biological networks. *Bioinformatics*, 2006, 22(8): 1021-1023.

[16] Palla G, Derényi I, Farkas I, Vicsek T. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 2005, 435(7043): 814-818.

[17] Wang S, Wu F. Detecting overlapping protein complexes in PPI networks based on robustness. *Proteome Science*, 2013, 11(Suppl 1): S18.

[18] Enright A, Van Dongen S, Ouzounis C. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*, 2002, 30(7): 1575-1584.

[19] Pizzuti C, Rombo S. A co-clustering approach for mining large

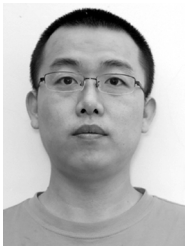
protein-protein interaction networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2012, 9(3): 717-730.
[20] Anirban M, Sumanta R, Moumita D. Detecting protein complexes in a PPI network: A gene ontology based multi-objective evolutionary approach. *Molecular BioSystems*, 2012, 8(11): 3036-3048.
[21] Eileen M H. Detection of overlapping protein complexes using a protein ranking algorithm. *In Proc. the 9th Int. Conference on Innovations in Information Technology,* March 2013, pp.233-236.
[22] Zaki N, Berengueres J, Efimov D. Detection of protein complexes using a protein ranking algorithm. *Proteins: Structure, Function, and Bioinformatics*, 2012, 80(10): 2459-2468.
[23] Wang Y, Qian X. Functional module identification in protein interaction networks by interaction patterns. *Bioinformatics*, 2014, 30(1): 81-93.
[24] Raghavan U N, Albert R, Kumara S. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 2007, 76(3): 036106.
[25] Stark C, Breitkreutz B J, Reguly T *et al.* BioGRID: A general repository for interaction datasets. *Nucleic Acids Research*, 2006, 34(Suppl 1): D535-D539.
[26] Salwinski L, Miller C S, Smith A J *et al.* The database of interacting proteins: 2004 update. *Nucleic Acids Research*, 2004, 32(Database Issue): D449-D451.
[27] Mewes H W, Amid C, Arnold R *et al.* MIPS: Analysis and annotation of proteins from whole genomes. *Nucleic Acids Research*, 2004, 32(Database Issue): D41-D44.
[28] Pu S, Wong J, Turner B *et al.* Up-to-date catalogues of yeast protein complexes. *Nucleic Acids Research*, 2009, 37(3): 825-831.
[29] Wu Z H, Lin Y F, Gregory S *et al.* Balanced multi-label propagation for overlapping community detection in social networks. *Journal of Computer Science and Technology*, 2012, 27(3): 468-479.
[30] Hong E L, Balakrishnan R, Dong Q *et al.* Gene ontology annotations at SGD: New data sources and annotation methods. *Nucleic Acids Research*, 2008, 36(Suppl 1): D577-D581.
[31] Boyle E I, Weng S, Gollub J *et al.* GO::TermFinder — Open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. *Bioinformatics*, 2004, 20(18): 3710-3715.
[32] Berman H M, Westbrook J, Feng Z *et al.* The protein data bank. *Nucleic Acids Research*, 2000, 28(1): 235-242.
[33] Naveed H, Han J J. Structure-based protein-protein interaction networks and drug design. *Quantitative Biology*, 2013, 1(3): 183-191.
[34] Zhang Q C, Petrey D, Deng L *et al.* Structure-based prediction of protein-protein interactions on a genome-wide scale. *Nature*, 2012, 490(7421): 556-560.

Qi-Guo Dai received his B.S. degree in computer science from Hubei University of Automobile Technology in 2006 and M.S. degree in computer science from Beijing University of Technology in 2010. He is a Ph.D. candidate in the School of Computer Science and Technology, Harbin Institute of Technology, China. His research interests include bioinformatics and machine learning.

Mao-Zu Guo received his B.S. and M.S. degrees from Harbin Engineering University, China, in 1988 and 1991, respectively, and Ph.D. degree from Harbin Institute of Technology, China, in 1998, all in computer science. He is currently a professor in the School of Computer Science and Technology, Harbin Institute of Technology, China. His research interests include bioinformatics and machine learning.

Xiao-Yan Liu received her B.S and M.S degrees in computer science from Harbin Engineering University, China, and Ph.D. degree in engineering mechanics from Harbin Institute of Technology, China. She is currently an associate professor in the School of Computer Science and Technology at Harbin Institute of Technology, China. Her research interests include bioinformatics and knowledge-based systems.

Zhi-Xia Teng received the B.S. degree in information management and information system from Northeast Forestry University, Harbin, in 2005. Now she is a Ph.D. candidate in computer science and technology at Harbin Institute of Technology, China. Her research interests include protein function prediction, protein network, and biological information mining.



Chun-Yu Wang received his B.S. and M.S. degrees in computer science from Harbin Institute of Technology, China. Now he is a lecturer and a Ph.D. candidate in computer science and technology at Harbin Institute of Technology, China. His research interests include bioinformatics and machine learning.