BIOINFORMATICS ORIGINAL PAPER Vol. 25 no. 15 2009, pages 1891–1897
doi:10.1093/bioinformatics/btp311

Systems biology

Complex discovery from weighted PPI networks

Guimei Liu[1,*], Limsoon Wong[1] and Hon Nian Chua[2]

[1]School of Computing, National University of Singapore, Singapore and [2]Institute for Infocomm Research, Singapore

Received on December 22, 2008; revised on April 21, 2009; accepted on May 6, 2009

Advance Access publication May 12, 2009

Associate Editor: Alfonso Valenica

ABSTRACT

Motivation: Protein complexes are important for understanding principles of cellular organization and function. High-throughput experimental techniques have produced a large amount of protein interactions, which makes it possible to predict protein complexes from protein–protein interaction (PPI) networks. However, protein interaction data produced by high-throughput experiments are often associated with high false positive and false negative rates, which makes it difficult to predict complexes accurately.

Results: We use an iterative scoring method to assign weight to protein pairs, and the weight of a protein pair indicates the reliability of the interaction between the two proteins. We develop an algorithm called CMC (clustering-based on maximal cliques) to discover complexes from the weighted PPI network. CMC first generates all the maximal cliques from the PPI networks, and then removes or merges highly overlapped clusters based on their interconnectivity. We studied the performance of CMC and the impact of our iterative scoring method on CMC. Our results show that: (i) the iterative scoring method can improve the performance of CMC considerably; (ii) the iterative scoring method can effectively reduce the impact of random noise on the performance of CMC; (iii) the iterative scoring method can also improve the performance of other protein complex prediction methods and reduce the impact of random noise on their performance; and (iv) CMC is an effective approach to protein complex prediction from protein interaction network.

Contact: liugm@comp.nus.edu.sg

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Protein complexes are important for understanding principles of cellular organization and function. High-throughput experimental techniques have produced a large amount of protein interactions, which makes it possible to uncover protein complexes from protein–protein interaction (PPI) networks. A PPI network can be modeled as an undirected graph, where vertices represent proteins and edges represent interactions between proteins. Protein complexes are groups of proteins that interact with one another, so they are usually dense subgraphs in PPI networks. Several algorithms based on graph clustering, dense region finding or clique finding have been developed to discover protein complexes from PPI networks, including MCL (van Dongen, 2000), RNSC (Przulj and Wigle, 2003), MCode (Bader and Hogue, 2003), DPClus (Altaf-Ul-Amin *et al.*, 2006), CFinder (Adamcsek *et al.*, 2006), PCP (Chua *et al.*, 2008) and IPCA (Li *et al.*, 2008).

It has been noticed that protein interaction data produced by high-throughput experiments are often associated with high false positive and false negative rates due to the limitations of the associated experimental techniques and the dynamic nature of protein interaction maps, which may have a negative impact on the performance of complex discovery algorithms. Many computational approaches have been proposed to assess the reliability of high-throughput protein interaction data. Some methods estimate the overall error rate of a given PPI dataset (Deane *et al.*, 2002; Deng *et al.*, 2003; D'haeseleer and Church, 2004). More complicated methods seek to assess the reliability of individual interactions (Gilchrist *et al.*, 2004; Patil and Nakamura, 2005; von Mering *et al.*, 2002). Various information has been used in these methods, such as gene annotations, gene expression and sequence homology. Several methods use solely the topology of the protein interaction networks (Chen *et al.*, 2005; Goldberg and Roth, 2003; Saito *et al.*, 2003). CD-distance (Brun *et al.*, 2003) and FSWeight (Chua *et al.*, 2006) are two measures calculated based on the number of common neighbors of two proteins. They are initially proposed to predict protein functions, and have been shown to perform well for assessing the reliability of protein interactions (Chen *et al.*, 2006).

Chua *et al.* (2008) have shown that by using FSWeight to remove unreliable interactions and add new interactions with high FSWeight score, the performance of several clustering algorithms can be improved. In our previous work (Liu *et al.*, 2008), we proposed an iterative scoring method to assess the reliability of protein interactions and predict new interactions, and it has been shown to perform better than CD-distance and FSWeight. In this article, we study the impact of this iterative scoring method on complex discovery. We develop a simple algorithm called CMC (clustering-based on maximal cliques) that uses maximal cliques to discover complexes from weighted PPI networks. CMC first finds maximal cliques from PPI networks, and then removes or merges highly overlapped maximal cliques based on their interconnectivity.

The rest of the article is organized as follows. Section 2 introduces the iterative scoring method. The complex discovery algorithm CMC is described in Section 3, and experiment results are reported and discussed in Section 4. Finally, Section 5 summarizes the article.

2 THE ITERATIVE SCORING METHOD

Many methods have been proposed to assess the reliability of protein interactions. These methods usually assign a score to each protein pair such that the higher the score is, the more likely the two

*To whom correspondence should be addressed.

proteins interact with each other. The intuition behind the iterative scoring method is simple: if the score of an interaction reflects its reliability, then the scored interactions should better represent the actual interaction network than the initial binary ones, and we should be able to further improve score computation by recomputing the score of each protein pair using the scored interactions. Here, we use a variant of CD-distance to calculate the score of protein pairs, and we call it AdjustCD. Given a pair of proteins u and v, AdjustCD of edge (u,v) is defined as follows:

$$\text{AdjustCD}(u,v)=\frac{2|N_u\cap N_v|}{|N_u|+\lambda_u+|N_v|+\lambda_v}$$

where λ_u and λ_v are used to penalize proteins with very few neighbors [as in FSWeight (Chua *et al.*, 2006)], and they are defined as follows:

$$\lambda_u=\max\left\{0,\frac{\sum_{x\in V}|N_x|}{|V|}-|N_u|\right\}$$

$$\lambda_v=\max\left\{0,\frac{\sum_{x\in V}|N_x|}{|V|}-|N_v|\right\}$$

Based on the definition, if the degree of a vertex u is below the average degree, then it is adjusted to the average degree.

The iterative version of AdjustCD is defined as follows:

$$w^k(u,v)=\frac{\sum_{x\in N_u\cap N_v}(w^{k-1}(x,u)+w^{k-1}(x,v))}{\sum_{x\in N_u}w^{k-1}(x,u)+\lambda_u^k+\sum_{x\in N_v}w^{k-1}(x,v)+\lambda_v^k}$$

where $w^{k-1}(x,u)$ is the score of (x,u) in the $(k-1)$-th iteration. Initially, if there is an edge between x and u in the original PPI network, then $w^0(x,u)=1$, otherwise, $w^0(x,u)=0$. The two terms, λ_u^k and λ_v^k are also defined based on weighted degree:

$$\lambda_u^k=\max\left\{0,\frac{\sum_{x\in V}\sum_{y\in N_x}w^{k-1}(x,y)}{|V|}-\sum_{x\in N_u}w^{k-1}(x,u)\right\}$$

$$\lambda_v^k=\max\left\{0,\frac{\sum_{x\in V}\sum_{y\in N_x}w^{k-1}(x,y)}{|V|}-\sum_{x\in N_v}w^{k-1}(x,v)\right\}$$

It is not difficult to see that $w^1(u,v)=\text{AdjustCD}(u,v)$. CD-distance and FSWeight can be iterated in a similar way.

In our previous work (Liu *et al.*, 2008), we have shown that the iterative scoring method can improve functional homogeneity and localization coherence of top ranked interactions, and the iterative scoring method reaches the best performance when $k=2$, and the subsequent iterations do not improve the performance further. The convergence of the iterative scoring method is shown in Section 5 of Supplementary Materials. In the rest of the article, we use AdjustCD^i to denote the iterative scoring method with $k=i$.

3 THE CMC ALGORITHM

The CMC algorithm uses maximal cliques to identify dense subgraphs from PPI networks, and it consists of three steps. In the first step, it finds all the maximal cliques from the weighted PPI network; in the second step, it ranks the cliques according to their weighted density; finally, it merges or removes highly overlapped cliques.

Although enumerating all maximal cliques is NP-hard, this does not pose a problem in PPI networks because PPI networks are usually sparse. CMC uses the Cliques algorithm proposed by Tomita *et al.* (2006) to find maximal cliques. The Cliques algorithm uses a depth-first search strategy to enumerate all maximal cliques, and it can effectively prune non-maximal cliques during the enumeration process.

Next, CMC assigns a score to each clique, and ranks cliques in descending order of their score. The score of a clique C is defined as its weighted density:

$$score(C)=\frac{\sum_{u\in C,v\in C}w(u,v)}{|C|\cdot(|C|-1)}$$

where $w(u,v)$ is the weight of the interaction between u and v calculated using AdjustCD or other scoring methods. Proteins in a larger clique are more likely to have more common neighbors than proteins in a smaller clique, so the edges within a larger clique are likely to have higher weights than those in a smaller clique. Thus, if the density of two cliques is the same, the weighted density of the larger clique is likely to be higher than that of the smaller clique.

Thousands of maximal cliques may be generated from a PPI network and many of them overlap with one another. The highly overlapped cliques should be removed to reduce result size. It is also desirable to merge highly overlapped cliques to form bigger yet still dense subgraphs as complexes are not necessarily fully connected and PPI data may be incomplete. CMC uses the interconnectivity between two cliques to decide whether two overlapped cliques should be merged together or not. The interconnectivity between two cliques C_1 and C_2 is defined based on the connectivity between the non-overlapping part of the two cliques:

$$\textit{inter-score}(C_1,C_2)=\sqrt{\frac{\sum_{u\in(C_1-C_2)}\sum_{v\in C_2}w(u,v)}{|C_1-C_2|\cdot|C_2|}\cdot\frac{\sum_{u\in(C_2-C_1)}\sum_{v\in C_1}w(u,v)}{|C_2-C_1|\cdot|C_1|}}$$

Given a set of cliques ranked in descending order of their score, denoted as $\{C_1,C_2,\ldots,C_k\}$, the CMC algorithm removes and merges highly overlapped cliques as follows. For every clique C_i, CMC checks whether there exists clique C_j such that C_j has a lower score than C_i and $|C_i\cap C_j|/|C_j|\geq overlap\_thres$, where *overlap_thres* is a predefined threshold for overlapping. If such C_j exists, then CMC uses the interconnectivity score between C_i and C_j to decide whether to remove C_j or merge C_j with C_i. If inter-score$(C_1,C_2)\geq$ *merge_thres*, then C_j is merged with C_i, otherwise, C_j is removed. Here, *merge_thres* is a predefined threshold for merging. Algorithm 1 shows the pseudo-codes of the CMC algorithm. In the rest of the article, we call the dense subgraphs generated by CMC *clusters*.

4 RESULTS

In this section, we first describe the datasets and evaluation methods used in our experiments, and then study the performance of CMC and the impact of the iterative scoring method on CMC. We compare CMC with three other clustering algorithms: MCL (van Dongen, 2000), MCode (Bader and Hogue, 2003) and CFinder (Adamcsek *et al.*, 2006). CFinder is a software tool based on the CPM algorithm (Palla *et al.*, 2005). Due to the limitation of space, we show only some of the figures in the article. More results can be found in Supplementary Materials.

Algorithm 1 CMC Algorithm

Input:
G is a weighted PPI network;
overlap_thres is the overlapping threshold;
merge_thres is the merging threshold;
Output:
set of dense subgraphs (clusters) discovered from G;
Description:
1: Generate the set of maximal cliques from G using the Cliques algorithm;
2: Calculate the score of every clique, and rank them in descending order, denoted as $\mathcal{C} = \{C_1, C_2, \cdots, C_k\}$;
3: **for all** $C_i \in \mathcal{C}$ **do**
4: **for all** $C_j \in \mathcal{C}$ and C_j is after C_i **do**
5: **if** $|C_i \cap C_j|/|C_j| \geq$ *overlap_thres* **then**
6: **if** *inter-score* $(C_i, C_j) \geq$ *merge_thres* **then**
7: $C_i = C_i \cup C_j$;
8: $\mathcal{C} = \mathcal{C} - C_j$;
9: output the clusters in $\mathcal{C}$;

Table 1. Reference complex sets

			Size			Density	
Datasets	#cmplx	#proteins	Max	Avg	Median	Avg	Median
MIPS	162	1171	95	14.93	9	0.408	0.318
Aloy	63	544	34	9.22	7	0.747	0.833

4.1 Datasets

We use a combined PPI dataset containing yeast protein interactions generated by six individual experiments, including interactions characterized by mass spectrometry technique by Ho *et al.* (2002), Gavin *et al.* (2002, 2006) and Krogan *et al.* (2006), and interactions produced using two-hybrid techniques by Uetz *et al.* (1999) and Ito *et al.* (2001). This dataset contains 4671 proteins and 20 461 interactions, among which 11 487 interactions have common neighbors.

Two reference sets of protein complexes are used in our experiments. The first set comprises of hand-curated complexes from MIPS (Mewes *et al.*, 2004) and the other set is generated by Aloy *et al.* (2004). For both sets, we keep only those complexes with size no less than 4. Table 1 shows the number of complexes, number of proteins, the maximal, average and median size, and the average and median density of the complexes in the reference complex sets.

4.2 Evaluation methods

One evaluation method we use is to match the generated clusters with reference complex sets, and calculate recall (sensitivity) and precision at complex and complex-protein pair level, respectively. Let S be a cluster, C be a reference complex, V_S be the set of proteins contained in S and V_C be the set of proteins contained in C. We define the matching score between S and C as the Jaccard index between V_S and V_C.

$$match\_score(S, C) = \frac{|V_S \cap V_C|}{|V_S \cup V_C|}$$

Given a threshold *match_thres*, if $match\_score(S, C) \geq match\_thres$, then we say S and C match each other. In most of our experiments, we set *match_thres* = 0.5, which requires that $|V_s \cap V_c| \geq |V_S| + |V_C|/3$. For example, if $|V_S| = |V_C| = 8$, then the overlap between S and C should be at least 6.

Given a set of reference complexes $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$ and a set of predicted complexes $\mathcal{P} = \{S_1, S_2, \ldots, S_m\}$, recall and precision at complex level are defined as follows:

$$Recall_c = \frac{|\{C_i | C_i \in \mathcal{C} \wedge \exists S_j \in \mathcal{P}, S_j \text{ matches } C_i\}|}{|\mathcal{C}|}$$

$$Precision_c = \frac{|\{S_j | S_j \in \mathcal{P} \wedge \exists C_i \in \mathcal{C}, C_i \text{ matches } S_j\}|}{|\mathcal{P}|}$$

Recall and precision at complex–protein pair level are defined as follows:

$$Recall_p = \frac{\sum_{i=1}^{n} \max\{overlap(C_i, S_j) | \forall S_j \text{ that matches } C_i\}}{\sum_{i=1}^{n} |C_i|}$$

$$Precision_p = \frac{\sum_{j=1}^{m} \max\{overlap(S_j, C_i) \ | \forall C_i \text{ that matches } S_j\}}{\sum_{j=1}^{m} |S_j|}$$

where $overlap(C_i, S_j) = |V_{C_i} \cap V_{S_j}|$. In Section 7 of Supplementary Materials, we use an example to illustrate how to compute recall and precision at complex and complex–protein pair level.

A protein complex can only be formed if its proteins are localized within the same compartment of the cell, so we also use localization coherence of complexes to measure the quality of the complexes. We use cellular component terms from Gene Ontology (Ashburner *et al.*, 2000), and select only informative GO localization terms. A GO term is informative if no less than 30 proteins are annotated with this term and none of its descendant terms are annotated to no less than 30 proteins (Zhou *et al.*, 2002).

Let $\mathcal{L} = \{L_1, L_2, \ldots, L_k\}$ be a set of localization groups, where each group contains a set of co-localized proteins. The co-localization score of a complex is defined as the maximal fraction of proteins in this complex that are in the same localization group among those proteins with localization annotations. The co-localization score of a set of complexes $\mathcal{C}$ is defined as the weighted average score over all complexes:

$$loc\_score(\mathcal{C}) = \frac{\sum_{C \in \mathcal{C}} \max\{overlap(C, L_i) | i = 1, 2, \ldots, k\}}{\sum_{C \in \mathcal{C}} |\{p | p \in C \wedge \exists L_i \in \mathcal{L}, p \in L_i\}|}$$

4.3 Efficiency analysis

Our experiments were conducted on a PC with Intel(R) Core^{TM}2 Duo CPU of 2.33 GHz and 3.25 GB RAM. The third and fourth columns in Table 2 show the total running time of CMC and the time for finding maximal cliques under different PPI dataset settings. The last two columns shows the total number of maximal cliques generated and the number of final clusters. The *overlap_thres* is set to 0.5 and *merge_thres* is set to 0.25. We also exclude clusters with size <4 from output. In Table 2, 'Original' means the original PPI network; 'Weighted' means the PPI network is weighted using AdjustCD[2] and interactions with a score of 0 are removed; 'Weighted+top-20000' means that the top-20000 new interactions

ranked by AdjustCD^2 are added to the weighted network; and 'Original+random 204610' means that 204 610 random interactions are added to the original PPI network.

Table 2 shows that even with 10 times of additional interactions in the original network, finding maximal cliques still takes <2 s on our machine. This indicates that even though finding maximal cliques is NP-hard, it is feasible to apply it on PPI networks as PPI networks are usually sparse.

4.4 The effect of *overlap_thres* and *merge_thres*

In this experiment, we study the effect of the two thresholds *overlap_thres* and *merge_thres* on the performance of CMC. Figure 1 shows recall and precision of the clusters generated by CMC under different parameter settings on Aloy reference set. The PPI network is weighted using AdjustCD^2, and no further interactions are removed except those with a score of 0. No new interactions are added. The matching threshold *match_thres* is set to 0.50.

Figure 1 shows that CMC is more sensitive to *merge_thres* than to *overlap_thres*. For recall and precision at complex level, it reaches the best performance when *merge_thres* = 0.25. At complex–protein pair level, when *merge_thres* is decreased from 1 to 0.25, precision decreases slightly for the top ranked clusters, but when more clusters are included, precision increases. When *merge_thres* = 0.15, precision at both complex level and complex–protein pair level is lower than that when *merge_thres* = 0.25. We also observed that localization coherence score decreases with the decrease of *merge_thres*. This may be caused by the fact that when *merge_thres* is smaller, more cliques are merged together and clusters are getting larger. When *overlap_thres* = 0.5 and *merge_thres* = 1, co-localization score is 0.910. It decreases to 0.866 when *merge_thres* = 0.25, and drops to 0.827 when *merge_thres* = 0.15. Average cluster size increases from 5.88 to 7.79 when *merge_thres* is changed from 1 to 0.25.

The results on MIPS reference set is similar to that on Aloy, when the PPI network is weighted using AdjustCD^2 (see Fig. 1 in Supplementary Materials). When the PPI network is weighted using AdjustCD^1, CMC achieves the best performance when *merge_thres* = 0.15, while on the unweighted PPI network, CMC achieves the best performance when *merge_thres* = 0.5 (see Fig. 2 and 3 in Supplementary Materials). In the remaining experiments, we always set *overlap_thres* to 0.5, and set *merge_thres* to 0.25 if the PPI network is weighted using AdjustCD^2, 0.15 if the PPI network is weighted using AdjustCD^1 and 0.5 if the PPI network is unweighted.

Table 2. Running time (in seconds)

	#PPIs	Total	MaxClq	#max_clqs	# clusters
Original	20461	0.734 s	0.219 s	2877	134
Weighted	11487	0.391 s	0.110 s	2877	205
Weighted + top-20000	31487	7.172 s	0.625 s	20689	464
Original + random 204610	225071	4.437 s	1.328 s	6369	1218

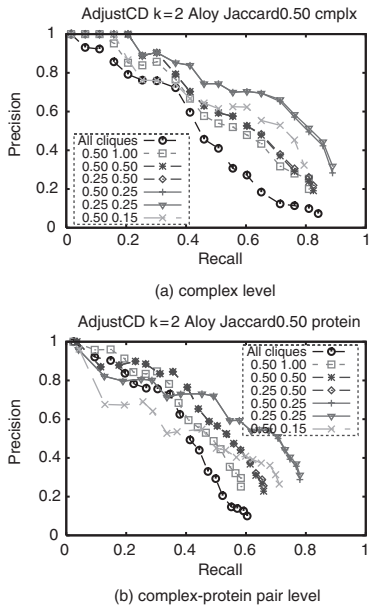

Fig. 1. Precision versus recall on reference complex set Aloy under different parameter settings. The first number in the legend of the two figures is *overlap_thres* and the second number is *merge_thres*. The PPI network is weighted using AdjustCD^2.

4.5 Purifying and expanding PPI networks

The weights of interactions reflect their reliabilities. Interactions with low scores are likely to be false positives, and protein pairs that are not included in the original PPI network but have high scores may be false negatives. In this experiment, we study whether removing interactions with low scores and adding new interactions with high scores will improve the performance of CMC.

Figure 2 shows recall and precision of the clusters generated by CMC on Aloy reference set when different amounts of PPIs are added or removed. We can see that adding interactions with high scores does not have a significant impact on precision and recall except that when too many new interactions are added, precision drops slightly. This is probably because merging overlapped cliques together already has the effect of adding missing interactions, and these missing edges are likely to have high scores since the corresponding protein pairs share some common neighbors.

When new interactions are added, localization coherence score decreases; when interactions with low scores are removed, localization coherence score increases. For example, when top 10 000 interactions are added, co-localization score decreases from 0.866 to 0.784. When only top 6000 interactions in the original network are retained, that is, the bottom 5487 interactions with low score are removed, co-localization score increases from 0.866 to 0.899. More results on MIPS reference set and when the PPI network is weighted using AdjustCD^1 can be found in Supplementary Materials (Figs 4 and 5, and Table 2).

We have also studied the impact of removing and adding interactions on the performance of MCL, MCode and CFinder. MCL and CFinder show similar performance to CMC, and they achieve the best performance when only top 6000 interactions are retained and no new interactions are added (see Figs 6–9 in the Supplementary Materials). However, MCode shows a big improvement when new interactions are added. It has the best performance when 3000 new interactions are added (see Figs 10 and 11 in the Supplementary Materials).

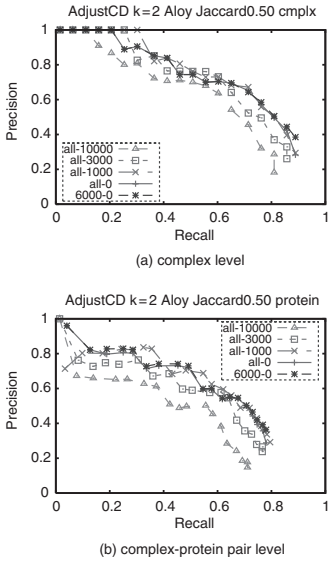

Fig. 2. Precision versus recall on reference complex set Aloy when different amount of interactions are removed or added. The score of the protein pairs are calculated using AdjustCD[2]. In the legend of the two figures, 'all-*N*' means all the interactions in the original network with a score larger than 0 are retained and top *N* new interactions that are not in the original PPI network are added; '6000-0' means only the top 6000 interactions in the original network are retained. *overlap_thres* = 0.5, *merge_thres* = 0.25, *match_thres* = 0.50.

4.6 The impact of the iterative scoring method and comparison of four clustering methods

In this experiment, we study the impact of the iterative scoring method on the performance of the four clustering algorithms under different k values. For MCL, we set inflation to 1.8; for MCode, we set depth to 100, node score percentage to 0, and percentage for complex fluffing to 0.2 as suggested by Brohee and van Helden (2006). For CFinder, we set k-clique size to 4. For the other parameters, we use their default values.

Table 3 shows the performance of the four clustering methods when the interactions are weighted using AdjustCD under different k values. The matching threshold is set to 0.50. Under this matching threshold, if we match the clusters generated by CMC with random complexes generated by swapping members among reference complexes, the precision and recall are always 0 over 1000 runs. The maximal recalls of CMC under different matching thresholds are shown in Table 3 of the Supplementary Materials.

All the algorithms have considerable improvement on weighted networks than on the original unweighted network. CMC and CFinder both show similar performance in the two cases when $k = 2$ and $k = 20$, which is better than their respective performance when $k = 1$. MCL and MCode both show similar performance in the two cases when $k = 1$ and $k = 2$, which is better than their respective performance when $k = 20$. Therefore, $k = 2$ is a safe choice for all the four clustering methods.

CMC shows the highest recall among the four clustering algorithms, but its precision is lower than CFinder and MCL when it reaches its maximal recall. However, under the same recall, CMC has higher precision than other algorithms (see Fig. 12 in

Table 3. The impact of the iterative scoring method on the performance of four clustering methods

Scoring method: AdjustCD					*match_thres*=0.50							
					Aloy (#complexes: 63)				**MIPS (#complexes: 162)**			
Clustering methods	k	#clusters	Avg size	loc_ score	#matched clusters	Precision	#matched complexes	Recall	#matched clusters	Precision	#matched complexes	Recall
CMC	0	172	9.83	0.823	53	0.308	53	0.841	42	0.244	55	0.340
	1	121	9.42	0.897	50	**0.413**	49	0.778	41	**0.339**	51	0.315
	2	148	8.50	0.899	57	0.385	**56***	**0.889**	44	0.297	**56***	**0.346**
	20	146	8.78	0.891	56	0.384	**56***	**0.889**	43	0.295	**56***	**0.346**
CFinder	0	103	13.84	0.528	39	0.379	38	0.603	34	0.330	40	0.247
	1	76	12.86	0.724	38	**0.500**	38	0.603	30	**0.395**	34	0.210
	2	95	11.66	0.713	44	0.463	**43**	**0.683**	36	0.379	46	0.284
	20	95	11.77	0.718	44	0.463	**43**	**0.683**	37	0.389	**49**	**0.302**
MCL	0	372	9.40	0.638	27	0.073	27	0.429	30	0.081	37	0.228
	1	120	10.18	0.848	49	0.408	49	0.778	40	0.333	**51**	**0.315**
	2	116	10.31	0.856	52	**0.448**	**52**	**0.825**	41	**0.353**	**51**	**0.315**
	20	110	10.75	0.849	49	0.445	49	0.778	37	0.336	47	0.290
MCode	0	61	7.31	0.849	20	0.328	20	0.317	18	0.295	22	0.136
	1	103	7.42	0.913	35	0.340	**35**	**0.556**	30	0.291	**39**	**0.241**
	2	88	8.67	0.897	34	**0.386**	34	0.540	29	**0.330**	**39**	**0.241**
	20	82	10.28	0.838	29	0.354	29	0.460	23	0.280	32	0.198

For CMC, MCL and CFinder, we retain only the top 6000 interactions, and no new interactions are added. For MCode, we retain all the interactions with non-zero score and add top 3000 new interactions with the highest score. The 2nd column is the number of iterations k of the iterative scoring method, and $k = 0$ means the PPI network is unweighted. The 3rd column is the number of clusters generated, the 4th and 5th column is the average size and co-localization score of generated clusters.

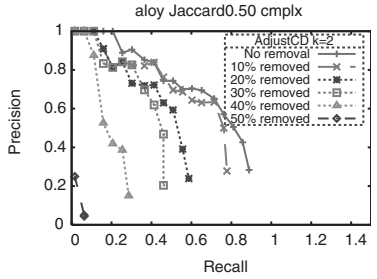

Fig. 3. Precision versus recall at complex level on Aloy reference set when different amount of interactions are randomly removed. Score method: AdjustCD^2, *overlap_thres* = 0.5, *merge_thres* = 0.15, *match_thres* = 0.50.

Supplementary Materials). The average co-localization score of the clusters generated by CMC is higher than those generated by CFinder and MCL. It is possible that some of the unmatched clusters generated by CMC are unknown complexes. We have annotated the clusters generated by CMC using GO terms. Among the 85 clusters that do not match any complex in MIPS or Aloy, 67 of them have common GO terms that are annotated to at least half of their members (see CMC_unmatchedclusters.xls in Supplementary Materials).

We studied the number of times that complexes and clusters are being matched. For the CMC algorithm, on Aloy dataset, the matching between complexes and clusters is almost one-to-one. There is only one or two clusters (complexes) are matched with two complexes (clusters). On the MIPS dataset, there are more matched complexes than matched clusters. We found that almost all complexes are matched to at most one clusters except one or two are matched with two clusters, and most of the clusters are also matched with at most one complex, except that one cluster is matched with five or six complexes, and about five clusters are matched with more than one but less than five complexes. The statistics of the other three clustering methods are pretty similar.

We have also compared the iterative scoring method with two other scoring methods (Friedel *et al.*, 2008; Krogan *et al.*, 2006). These two methods calculate confidence scores from tandem affinity purification data using machine learning techniques, and they then use MCL to find complexes. The results show that CMC is relatively stable with respect to different scoring methods, and it performs better than MCL. MCL is more sensitive to different scoring methods, and it performs better when the interaction networks are scored using AdjustCD^2 (see Section 6 in the Supplementary Materials).

4.7 Random addition and deletion

In this experiment, we study the robustness of CMC by randomly removing and adding interactions. After randomly removing and adding interactions, we use AdjustCD to score the resulting PPI network, and retain only the top 6000 interactions. No new interactions with high score are added. Figure 3 shows precision and recall at complex level when different numbers of interactions are randomly removed, and the PPI network is scored using AdjustCD^2. When more than 20% interactions are removed, the performance of CMC decreases significantly. CMC shows similar behavior when PPI networks are weighted using AdjustCD^1 and AdjustCD^{20} (Fig. 13 in Supplementary Materials).

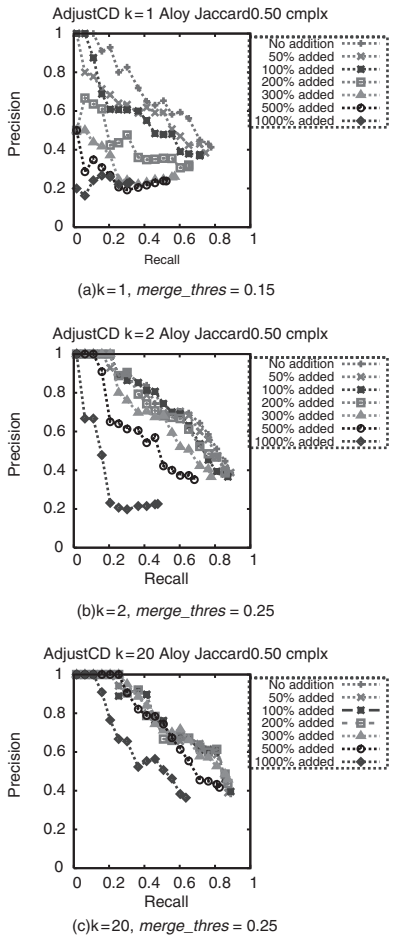

Fig. 4. Precision versus recall at complex level on Aloy reference set when different amount of interactions are randomly added. *overlap_thres* = 0.5, *match_thres* = 0.5.

Figure 4 shows precision and recall at complex level when different amounts of random interactions are added to the PPI network. When $k = 1$, the performance of CMC drops dramatically when the amount of random interactions added is equal to the number of interactions in the original PPI network. The situation gets better when $k = 2$. When $k = 20$, the performance of CMC drops greatly only when the amount of random noise added is 10 times of the number of interactions in the original PPI network. It shows that the iterative scoring method makes the CMC algorithm more robust to random noise. The same results are also observed for other clustering algorithms (see Figs 17–22 in the Supplementary Materials). We have compared the edge weights on the original network and noisy networks when they are weighted using AdjustCD under different k values. The results show that when $k \geq 20$, the edge weights of the noisy networks are very close to that on the original network when <500% noise are added (Table 4 in Supplementary Materials).

5 CONCLUSION AND DISCUSSION

In this article, we use an iterative method to weight PPI networks and develop a maximal clique-based algorithm CMC to discover

complexes from weighted PPI networks. The PPI weighting method plays several roles in the CMC algorithm. First, it assigns low scores to unreliable interactions, thus reducing their impact on complex discovery. This is why CMC is more robust to random noise when PPI networks are iteratively weighted. Second, it helps rank clusters properly by favoring larger clusters and clusters with fewer external connections. Our experiment results show that iterating AdjustCD can yield scores that better indicate the reliability of interactions, thus improving the performance of CMC considerably. The iterative scoring method also makes CMC more robust to random noise, and can improve the performance of other complex prediction algorithms as well, especially when there is lots of noise.

We observe that the average density of the clusters generated by CMC is higher than that of the clusters generated by MCL. The average density of the complexes in MIPS is much lower than that of the complexes in Aloy reference set (Table 1). All the four algorithms have much higher precision and recall on Aloy dataset than on MIPS dataset. This indicates that it is difficult to uncover complexes with low density even with MCL, which is expected to be able to uncover clusters with low density. In the future work, we will explore some heuristics to expand cliques and try to find clusters with relatively low density.

Funding: Singapore Ministry of Education grant R-252-000-274-112 and Singapore National Research Foundation grant NRF-G-CRP-2997-04-082(d).

Conflict of Interest: none declared.

REFERENCES

Adamcsek,B. *et al.* (2006) Cfinder:locating cliques and overlapping modules in biological networks. *Bioinformatics*, **22**, 1021–1023.

Aloy,P. *et al.* (2004) Structure-based assembly of protein complexes in yeast. *Science*, **303**, 2026–2029.

Altaf-Ul-Amin,M. *et al.* (2006) Development and implementation of an algorithm for detection of protein complexes in large interaction networks. *BMC Bioinformatics*, **7**, 207.

Ashburner,M. *et al.* (2000) Gene ontology: tool for the unification of biology. *Nat. Genet.*, **25**, 25–29.

Bader,G. and Hogue,C. (2003) An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, **4**, 2.

Brohee,S. and van Helden,J. (2006) Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics*, **7**, 488.

Brun,C. *et al.* (2003) Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. *Genome Biol.*, **5**, R6.

Chen,J. *et al.* (2005) Discovering reliable protein interactions from high-throughput experimental data using network topology. *Artif. Intell. Med.*, **35**, 37–47.

Chen,J. *et al.* (2006) Increasing confidence of protein-protein interactomes. In *Proceedings of the 17th International Conference on Genome Informatics*, Yokohama, Japan, pp. 284–297.

Chua,H. *et al.* (2006) Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions. *Bioinformatics*, **22**, 1623–1630.

Chua,H. *et al.* (2008) Using indirect protein-protein interactions for protein complex predication. *J. Bioinform. Comput. Biol.*, **6**, 435–466.

Deane,C.M. *et al.* (2002) Protein interactions: two methods for assessment of the reliability of high throughput observations. *Mol. Cell. Proteomics*, **1**, 349–356.

Deng,M. *et al.* (2003) Assessment of the reliability of protein-protein interactions and protein function prediction. In *Proceedings of Pacific Symposium on Biocomputing*, Lihue, Hawaii, pp. 140–151.

D'haeseleer,P. and Church,G. (2004) Estimating and improving protein interaction error rates. In *Proceedings of IEEE Computational Systems Bioinformatics Conference*, Stanford, CA, USA, pp. 216–223.

Friedel,C.C. *et al.* (2008) Bootstrapping the interactome: Unsupervised identification of protein complexes in yeast. In *Proceedings of the 12th Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, Singapore, pp. 3–16.

Gavin,A.-C. *et al.* (2002) Functional organization of the yeast proteome by systematic analysis of protein complexes *Nature*, **415**, 141–147.

Gavin,A.-C. *et al.* (2006) Proteome survey reveals modularity of the yeast cell machinery. *Nature*, **440**, 631–636.

Gilchrist,M. *et al.* (2004) A statistical framework for combining and interpreting proteomic datasets. *Bioinformatics*, **20**, 689–700.

Goldberg,D.S. and Roth,F.P. (2003) Assessing experimentally derived interactions in a small world. *Appl. Math. Biochem.*, **100**, 4372–4376.

Ho,Y. *et al.* (2002) Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. *Nature*, **415**, 180–183.

Ito,T. *et al.* (2001) A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl Acad. Sci. USA*, **98**, 4569–4574.

Krogan,N.J. *et al.* (2006) Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. *Nature*, **440**, 637–643.

Li,M. *et al.* (2008) Modifying the DPclus algorithm for identifying protein complexes based on new topological structures. *BMC Bioinformatics*, **9**, 398.

Liu,G. *et al.* (2008) Assessing and predicting protein interactions using both local and global network topological metrics. In *Proceedings of the 19th International Conference on Genome Informatics*. Gold Coast, Australia, pp. 138–149.

Mewes,H.W. *et al.* (2004) MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acids Res.*, **32**(Database issue), 41–44.

Palla,G. *et al.* (2005) Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, **435**, 814–818.

Patil,A. and Nakamura,H. (2005) Filtering high-throughput protein-protein interaction data using a combination of genomic features. *BMC Bioinformatics*, **6**, 100.

Przulj,N. and Wigle,D. (2003) Functional topology in a network of protein interactions. *Bioinformatics*, **20**, 340–348.

Saito,R. *et al.* (2003) Construction of reliable protein-protein interaction networks with a new interaction generality measure. *Bioinformatics*, **19**, 756–763.

Tomita,E. *et al.* (2006) The worst-case time complexity for generating all maximal cliques and computational experiments. *Theor. Comput. Sci.*, **363**, 28–42.

Uetz,P. *et al.* (1999) A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, **403**, 623–627.

van Dongen,S. (2000) Graph clustering by flow simulation. PhD Thesis, University of Utrecht, Utrecht, The Netherlands.

von Mering,C. *et al.* (2002) Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, **417**, 399–403.

Zhou,X. *et al.* (2002) Transitive functional annotation by shortest-path analysis of gene expression data. *Proc. Natl Acad. Sci. USA*, **99**, 12783–12788.