Journal of Bioinformatics and Computational Biology
Vol. 6, No. 3 (2008) 435–466

Imperial College Press
www.icpress.co.uk

USING INDIRECT PROTEIN–PROTEIN INTERACTIONS FOR PROTEIN COMPLEX PREDICTION

HON NIAN CHUA*

Graduate School of Integrated Sciences
National University of Singapore, Singapore
g0306417@nus.edu.sg

KANG NING†,‖, WING-KIN SUNG‡, HON WAI LEONG§
and LIMSOON WONG¶

Department of Computer Science
National University of Singapore, Singapore
†*ningkang@comp.nus.edu.sg*
‡*ksung@comp.nus.edu.sg*
§*leonghw@comp.nus.edu.sg*
¶*wongls@comp.nus.edu.sg*

Received 1 August 2007
Revised 1 December 2007
Accepted 3 January 2008

Protein complexes are fundamental for understanding principles of cellular organizations. As the sizes of protein–protein interaction (PPI) networks are increasing, accurate and fast protein complex prediction from these PPI networks can serve as a guide for biological experiments to discover novel protein complexes. However, it is not easy to predict protein complexes from PPI networks, especially in situations where the PPI network is noisy and still incomplete. Here, we study the use of indirect interactions between level-2 neighbors (level-2 interactions) for protein complex prediction. We know from previous work that proteins which do not interact but share interaction partners (level-2 neighbors) often share biological functions. We have proposed a method in which all direct and indirect interactions are first weighted using topological weight (FS-Weight), which estimates the strength of functional association. Interactions with low weight are removed from the network, while level-2 interactions with high weight are introduced into the interaction network. Existing clustering algorithms can then be applied to this modified network. We have also proposed a novel algorithm that searches for cliques in the modified network, and merge cliques to form clusters using a "partial

*Current address: Institute of Infocomm Research, A*STAR, Singapore. Email: hnchua@i2r.a-star.edu.sg
‖Current address: Pathology Informatics Research Core, Department of Pathology, University of Michigan, Ann Arbor, MI, USA. Email: kning@umich.edu; kning@med.umich.edu

clique merging" method. Experiments show that (1) the use of indirect interactions and topological weight to augment protein–protein interactions can be used to improve the precision of clusters predicted by various existing clustering algorithms; and (2) our complex-finding algorithm performs very well on interaction networks modified in this way. Since no other information except the original PPI network is used, our approach would be very useful for protein complex prediction, especially for prediction of novel protein complexes.

Keywords: Protein–protein interaction; protein complex prediction; level-2 interaction; partial clique merging.

1. Introduction

To understand the organization and dynamics of cell functions, the functional modules in a protein–protein interaction (PPI) network should be identified. A protein complex is a group of two or more associated proteins, and is a form of quaternary structure. Similar to phosphorylation, complex formation often serves to activate or inhibit one or more of the associated proteins. PPI networks are rapidly becoming larger and more complete as research on proteomics and systems biology proliferates.[1] As a result, more protein complexes have been identified,[2] particularly in the model organism *Saccharomyces cerevisiae* (baker's yeast). With a wealth of PPI datasets that are constantly increasing in size, efficient and accurate intelligent tools for the identification of protein complexes are of great importance. In this paper, we focus on predicting protein complexes from PPI data.

Currently, there are several approaches to the protein complex prediction problem[3–8]:

Clique finding. Spirin and Mirny[3] proposed using clique finding and superparamagnetic clustering with Monte Carlo optimization to find clusters of proteins. They found a significant number of protein complexes that overlap with experimentally derived ones. While clique finding[3] imposes stringent search criteria and generally results in greater precision, recall is limited because (1) protein interaction networks are incomplete and (2) protein complexes may not necessarily be complete subgraphs.

Clustering. Several approaches, such as MCODE,[5] are cluster-based. MCODE makes use of local graph density to find a protein complex. PPI networks are transformed to weighted graphs in which vertices are proteins and edges represent protein interactions. The algorithm operates in three stages: vertex weighting, complex prediction, and optimal postprocessing. Each stage involves several parameters that can be fine-tuned to get better predictions. The Restricted Neighborhood Search Clustering (RNSC) algorithm[7] is another approach based on clustering. Clustering approaches[5,8] generally yield better recall, but lower precision. King *et al.*[4] showed that it is possible to isolate high-precision subsets of clusters from those produced by RNSC using postprocessing based on functional homogeneity, cluster size, and interaction density. However, recall is drastically reduced as a result. Moreover, the

Table 1. Main features of protein complex prediction algorithms.

	RNSC	MCODE	MCL
Type	Local cost-based search	Local neighborhood density search	Flow simulation
Multiple assignment of protein	No	Yes	No
Weighted edge	No	No	Yes

approach makes use of functional information, which limits its applicability in less-studied genomes such as *Homo sapiens*, *Mus musculus*, and *Arabidopsis thaliana*. Recently, a popular clustering algorithm, Markov clustering algorithm (MCL),[9] has also been shown to perform well in an evaluation of algorithms for protein clustering in PPI networks.[6] MCL partitions the graph by discriminating strong and weak flows in the graph, which is shown to be very robust against graph alternations.

In this paper, we will use RNSC,[7] MCODE,[5] and MCL[9] for comparison. These approaches have been recognized as the state of the art for the task of complex discovery, and have been recently reviewed and compared in Brohée and van Helden.[6] Table 1 summarizes the main features of these algorithms.

We know from Chua *et al.*[10] that many proteins that do not interact, but share common interaction partners, share functions and participate in similar pathways. The interactions between these proteins are referred to as "level-2 neighbors". Chua *et al.*[10] also proposed a topological weight, FS-Weight, for estimating functional association between direct and indirect interactions, that is shown to work well.

In this paper, we propose using these indirect interactions with FS-Weight to modify the existing PPI network as a preprocessing step to complex prediction. The original PPI network is expanded by including indirect interactions (relationships between pairs of proteins that do not interact, but share common interaction partners). A topological weight, FS-Weight (functional similarity weight), is then computed for both direct and indirect interactions. Interactions with weights below a threshold are removed. We also propose a new algorithm that incorporates FS-Weight for complex prediction. The algorithm employs clique finding on a modified PPI network, retaining the benefits of clique-based approaches while improving recall. The algorithm first searches for cliques in the modified network, and iteratively merges them by "partial clique merging" to form larger clusters.

In the rest of this paper, we shall refer to predicted protein clusters as *clusters*, and known protein complexes as *complexes*.

2. Indirect Interactions and FS-Weight

A PPI network can be modeled as a graph $G = (V, E)$. Each vertex $v_k \in V$ represents a protein, while each edge $\{v_i, v_j\} \in E$ represents an interaction between

the proteins v_i and v_j. For the rest of the paper, we will consider PPI networks using this representation.

2.1. *Indirect interactions*

We refer to a physical interaction in the PPI network as a level-1 interaction; and the relationship between two proteins which do not interact, but share common interaction interacting partners, as a level-2 or indirect interaction. Members in a real complex may not have physical interactions with all other members; hence, conventional methods (clique-based, density-based) may miss the detection of many members. Chua *et al.*[10] showed that a topological weight, FS-Weight, can be used to identify both level-1 and level-2 interactions that are likely to share common functions within the local (level-1 and level-2) PPI interaction neighborhood. By incorporating interaction weighting and level-2 interactions, better functional predictions can be made for proteins.

2.2. *FS-Weight*

The functional similarity weight (FS-Weight) is formulated based on the underlying hypothesis that proteins share functions as a result of two distinct ways of association: direct functional association through interactions, and indirect functional association through interactions with common proteins. Direct functional association arises from the fact that proteins interact to perform common functions. Indirect functional association, on the other hand, arises from constraints in the physical and biochemical properties by which the interaction partners of a protein may be bound. If two proteins share many common interaction partners, they are more likely to possess similar properties that allow them to meet the constraints imposed by these common interaction partners. The FS-Weight is a measure of the overlap between the interaction partners of two proteins. The higher the overlap between the interaction partners of two proteins, the higher the likelihood of them sharing common functions.

The FS-Weight between two proteins u and v is defined as

$$S_{\mathrm{FS}}(u,v) = \frac{2|N_u \cap N_v|}{|N_u - N_v| + 2|N_u \cap N_v| + \lambda_{u,v}} \times \frac{2|N_u \cap N_v|}{|N_v - N_u| + 2|N_u \cap N_v| + \lambda_{v,u}}, \tag{1}$$

where N_p refers to the set that contains p and its level-1 neighbors. $\lambda_{u,v}$ is a pseudocount included in the computation to penalize similarity weights between protein pairs when proteins have very few level-1 neighbors, and is defined as

$$\lambda_{u,v} = \max(0, n_{\mathrm{avg}} - (|N_u - N_v| + |N_u \cap N_v|)), \tag{2}$$

where n_{avg} is the average number of neighbors per protein in the PPI network.

The FS-Weight can also be extended to take into account the reliability of each individual interaction as follows:

$$S_{\mathrm{FS}}(u,v) = \frac{2\sum_{w\in(N_u\cap N_v)} r_{u,w}r_{v,w}}{\left(\sum_{w\in(N_u-N_v)} r_{u,w} + \sum_{w\in(N_u\cap N_v)} r_{u,w}(1-r_{v,w})\right) + 2\sum_{w\in(N_u\cap N_v)} r_{u,w}r_{v,w} + \lambda_{u,v}} \times \frac{2\sum_{w\in(N_u\cap N_v)} r_{u,w}r_{v,w}}{\left(\sum_{w\in(N_v-N_u)} r_{v,w} + \sum_{w\in(N_u\cap N_v)} r_{v,w}(1-r_{u,w})\right) + 2\sum_{w\in(N_u\cap N_v)} r_{u,w}r_{v,w} + \lambda_{v,u}}, \tag{3}$$

where $r_{u,w}$ refers to the estimated reliability of the interaction between u and w.

In Chua *et al.*,[10] $r_{u,w}$ is estimated based on annotated proteins in the training set during cross-validation. To avoid possible bias that may be caused by using additional information (functional annotation), we exclude reliability estimation of interactions and set all $r_{u,w}$ to 1.

2.3. *Preprocessing the protein–protein interaction network*

Proteins within a complex interact to perform some common functions. By introducing level-2 interactions that are likely to represent strong functional relations into the interaction network, we may be able to capture members with less physical involvement in the complex. By using FS-Weight to filter out interactions that are less reliable and less likely to involve function sharing, we can also reduce the impact of noise and make more robust predictions.

Using FS-Weight, we modify an existing protein–protein interaction network in the following manner: level-1 interactions in the network that have low FS-Weights (weight below a certain threshold, FS-Weight$_{\mathrm{min}}$) are removed from the PPI network, while level-2 interactions that have high FS-Weights (above or equal to FS-Weight$_{\mathrm{min}}$) are added into the PPI network. FS-Weight$_{\mathrm{min}}$ is a value that is determined empirically. The modified PPI network can then be used for protein complex prediction using existing algorithms.

3. The PCP Algorithm

Here, we also designed a novel algorithm, Protein Complex Prediction (PCP), for complex prediction using the modified PPI network from the previous section. This algorithm differs from existing approaches in the following ways: it uses the FS-Weight information during the merging of cliques (clusters); and merging based on cliques is a clear and rigid method in graph theory, and is more viable based on reliable PPI networks. PCP attempts to achieve the high precision of clique-finding

algorithms whilst providing greater recall and computational tractability, without using any external information.

The PCP algorithm involves three main steps, which are explained in detail below.

3.1. *Step 1: maximal clique finding*

We first find maximal cliques within the modified PPI network. This can be done using an exhaustive approach or a heuristic approach.

3.1.1. *Exhaustive approach*

In this approach, we find all maximal cliques within the modified PPI network. This is done using the maximal clique-finding algorithm described in Tomita *et al.*[11] This algorithm has been shown to be very efficient on sparse graphs. All cliques of at least size 2 are reported. To make sure that there is no overlap among cliques, any overlap between cliques can only be assigned to one clique. There are many ways to do this. Since FS-Weight is an estimate for the likelihood of sharing functions, a cluster with a larger average FS-Weight would more likely represent a subset of a real complex. We define the average FS-Weight of a subgraph S with edges E_s as

$$FS_{\text{avg}}(S) = \frac{\sum_{(u,v)\in E_s} FS(u,v)}{|E_s|}. \tag{4}$$

Ideally, we want to find the best way to remove overlaps so that the total average FS_{avg} of all the final nonoverlapping cliques is maximized. However, since this is an NP-hard problem, we turn to heuristics. All cliques are first sorted by decreasing FS_{avg}. The clique with the highest FS_{avg} is selected and compared with the rest of the cliques. Whenever an overlap is found with another clique, the overlapping nodes are assigned to one of the two cliques such that the two cliques have a higher average FS_{avg}. An example is given in Fig. 1.

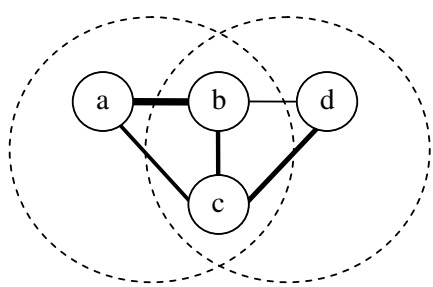


Fig. 1. Example of overlap resolution between two cliques $\{a, b, c\}$ and $\{b, c, d\}$. Line thickness depicts the relative FS-Weight scores of edges.

3.1.2. *Heuristic approach*

To improve the speed of the maximal clique finding, especially on very large PPI networks, we also propose a heuristic method for maximal clique finding from PPI networks. For each node x in a PPI network, sort all of its neighbors by decreasing degree. Starting x as a clique of size 1, go through each neighbor in the sorted order and add it to the clique if it interacts with all members of the clique. The maximum number of cliques we can have is only $|V|$, the number of nodes in PPI network. In this way, we may miss many cliques. However, since what we want is to have some strong clusters (i.e. cliques) to start with, we do not need to find all possible cliques. Moreover, the maximal cliques found will not overlap and hence no additional work is needed to remove overlaps.

To disambiguate the two clique-finding approaches, we refer to the PCP algorithm based on the heuristic clique-finding approach as PCP*.

3.2. *Step 2: computing intercluster density*

A protein complex is likely to consist of proteins forming a dense network of interactions, but may not necessarily form a complete clique. Due to the stringent definition of a clique, the resulting maximal cliques from the clique-finding step are relatively small and are likely to be partial representations of real complexes. To reconcile these smaller protein clusters into larger clusters that form a fuller representation of real complexes, we need to merge them appropriately.

We have tried to merge overlapping clusters based on the amount of overlapping vertices between them. However, the corresponding prediction results are not good, since each merge considers only overlapping vertices between two clusters, but overlooks the density of interactions between them. Hence, we define intercluster density (ICD), which is a measure of interconnectedness between two subgraphs, as a criterion for merging clusters.

The ICD essentially computes the FS-Weight density of intercluster interactions between the nonoverlapping proteins of two clusters. Since each cluster is already densely connected within itself, members that are common to both clusters also form a cluster that is densely connected. These overlapping members could be a network module that can be a subset of multiple complexes. If we consider overlapping members in the computation of ICD, we may end up merging two distinct complexes that share a common network module. Conversely, considering only nonoverlapping members ensures that the nodes in merged cliques are more uniformly interconnected. In practice, there will not be any overlap between cliques when computing ICD, since overlaps between cliques are removed during clique finding (Fig. 1). A high ICD indicates that the two clusters are highly connected. Using ICD to impose criteria for merging ensures that merged clusters retain a certain degree of interconnectedness between its members. The ICD between subgraphs S_a and S_b is defined as

$$ICD(S_a, S_b) = \frac{\sum S_{\mathrm{FS}}(i,j) | i \in (V_a - V_b), j \in (V_b - V_a), (i,j) \in E}{|V_a - V_b| \cdot |V_b - V_a|}, \tag{5}$$

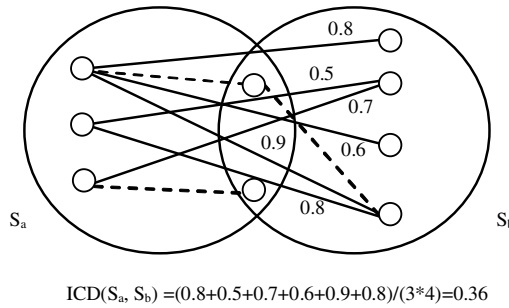


Fig. 2. Example of ICD computation. There are two clusters, and solid lines are used for ICD calculation.

where V_x is the set of vertices of subgraph S_x. An example of ICD computation is given in Fig. 2.

3.3. *Step 3: partial clique merging*

To merge cliques found in the PPI network, we define the term "partial cliques" as strongly connected subgraphs formed from the amalgamation of one or more cliques. Trivially, all cliques in the PPI network G are partial cliques. We begin with an initial graph G_p^0 in which each vertex represents a partial clique, and add an edge (u, v) between any pair of partial cliques u and v in G_p^0 if $ICD(u, v) \geq ICD_{\text{thres}}$. From G_p^0, we can again find maximal cliques among the vertices. Each clique in G_p^0 is therefore a cluster of partial cliques from G, where all pairs of partial cliques in the cluster fulfill a minimum level of interconnectedness defined by *ICD*. In other words, the vertices in each clique from G_p^0 can be merged to form a larger partial clique.

This process is then repeated to form bigger partial cliques. In each iteration i, a graph G_p^i is formed from PC^{i-1}, the partial cliques from the previous iteration, i.e. $G_p^i = (PC^{i-1}, \{(u, v) | ICD(u, v) \geq ICD_{\text{thres}}, u, v \in PC^{i-1}\})$. From G_p^i, we can again find maximal cliques among the vertices (partial cliques in G_p^{i-1}) and merge the proteins in these cliques to form bigger partial cliques. This is done until no further merge can be made. In order for the more connected partial cliques to merge first, we first perform the merge using $ICD_{\text{thres}} = 1$. The merging process is then repeatedly reinitiated while reducing ICD_{thres} by 0.1 until $ICD_{\text{thres}} \leq ICD_{\min}$. $ICD_{\min}$ is a threshold to be determined empirically. A smaller $ICD_{\min}$ will yield bigger clusters and vice versa. We refer to this merging method as "partial clique merging".

4. Experiment Settings and Datasets

4.1. *Implementation*

We implemented the preprocessing step using Perl, and the PCP (and PCP*) algorithm using C++. The implementation of the RNSC[7] algorithm was obtained from one of its authors, Igor Jurisca; while the implementation for the MCODE[5] and

MCL[9] algorithms was obtained from the main author of Ref. 6, Sylvian Brohée. The experiments were performed on a computer with a Pentium 4 central processing unit (CPU) (clock speed, 3.0 GHz), 1.0 GB of RAM, and running a Linux operating system.

4.2. *PPI datasets*

Two high-throughput PPI datasets are used in this paper. The first is obtained from the GRID database.[12] This dataset contains six protein interaction networks from the *Saccharomyces cerevisiae* (baker's yeast) genome. These include interactions characterized by the mass spectrometry technique from Ho *et al.*,[13] Gavin *et al.*,[14] Gavin *et al.*,[15] and Krogan *et al.*,[16] as well as two-hybrid interactions from Uetz *et al.*,[1] and Ito *et al.*[17] We refer to the combination of these six networks as PPI[Combined].

The second dataset is taken from a current release of the BioGRID database.[18] We only consider interactions derived from mass spectrometry and two-hybrid experiments, since these represent physical interactions. We shall refer to this dataset as PPI[BioGRID]. Table 3 presents the features of these datasets, as well as some characteristics of the clusters predicted by different algorithms.

4.3. *Protein complex datasets*

As a yardstick for prediction performance, we use protein complex data from the MIPS database.[2] These protein complexes are treated as a gold standard for analysis.

To examine whether false positives in predictions may turn out to be undiscovered annotations, we use two releases of the MIPS complex datasets — a dataset released on March 30, 2004; and a newer dataset released on May 18, 2006. We refer to two protein complex datasets as PC_{2004} and PC_{2006}, respectively. During validation, proteins that cannot be found in the input interaction network are removed from the complex data.

4.4. *Cluster scoring*

The density of a graph $G = (V, E)$ is defined as $D_G = |E|/|E|_{\max}$, where for a graph with loops, $|E|_{\max} = |V|(|V|+1)/2$; and for a graph with no loops, $|E|_{\max} = |V|(|V|-1)/2$. So, D_G is a real number ranging from 0.0 to 1.0. Resulting clusters $S = (V, E)$ from the algorithm are scored and ranked by cluster score, which is defined as the product of the density and the number of vertices in S, $(D_C \times |V|)$. This ranks larger and denser clusters higher in the results.

4.5. *Validation criteria*

4.5.1. *Matching criteria*

In order to study the relative performance of the PCP algorithm against existing algorithms, we need to define the criterion that determines whether a predicted

protein cluster matches a true protein complex. Bader and Hogue[5] defined a matching criterion using the overlap between a protein cluster S and a true protein complex C:

$$Overlap(S, C) = \frac{|V_S \cap V_C|^2}{|V_S| \cdot |V_C|}, \tag{6}$$

where V_S are the vertices of the subgraph defined by S, and V_C are the vertices of the subgraph defined by C.

In Bader and Hogue,[5] an overlap threshold of 0.2 is used to determine a match. King *et al.*[4] used a modified version of the overlap that is more stringent, but involves many empirically derived parameters which may not be applicable across different datasets. To simplify comparison, we used an overlap threshold of 0.25 to determine a match for all experiments in this work. Predicted protein clusters that match one or more true protein complexes with an overlap score above this threshold are identified as "matched predicted complexes", and the corresponding complexes are identified as "matched known complexes". Note that the number of "matched clusters", $matched_{\text{clusters}}$, may differ from the number of "matched complexes", $matched_{\text{complexes}}$, because one known complex can match one or more predicted clusters.

4.5.2. *Precision-recall analysis*

Unlike conventional prediction problems, predicted clusters rarely match real complexes perfectly. Multiple clusters may match the same complex and vice versa. While it may be possible to consider only cluster matches to plot the receiver operating characteristics (ROC) graph of each algorithm, the resulting comparison may not be meaningful since the true-positive rate does not reflect the number of correct complexes found. Hence, we choose to visualize prediction performance using precision based on cluster matches and recall based on complex matches. This provides a more realistic reflection of the usefulness of each method in a practical sense.

To measure the accuracies of prediction, the analysis on the precision and recall of different algorithms is computed. Precision and recall are defined as

$$\text{Precision} = \frac{matched_{\text{clusters}}}{predicted_{\text{clusters}}} \tag{7}$$

$$\text{Recall} = \frac{matched_{\text{complexes}}}{known_{\text{complexes}}}, \tag{8}$$

where $predicted_{\text{clusters}}$ and $known_{\text{complexes}}$ are the number of predicted clusters and the number of known (real) complexes, respectively.

The recall measure in our validation is determined by matched complexes instead of predicted clusters, and hence is not prone to bias. Moreover, the precision measure uses the number of predicted clusters as a denominator; thus, there should not be any significant bias in these validation measures. We only consider clusters and complexes of size 4 and above, since matches between clusters and complexes of

smaller sizes have relatively high probabilities of occurring by chance.[4] Note that, unlike the validation measures used in Brohée and van Helden,[6] we do not seek to evaluate the clustering properties of each algorithm. Rather, we are concerned about the actual usefulness of the algorithms in detecting clusters that match real complexes reasonably well.

4.5.3. *Precision-recall analysis based on protein membership assignment*

To avoid bias that may arise from large variations in the size of predicted complexes, we also introduce another precision-recall analysis based on protein membership assignment on some analysis. For this analysis, we defined two terms: protein–cluster pair (*PCl*) and protein–complex pair (*PCo*). Each *PCl* represents a unique protein–cluster relationship. For example, given two predicted clusters $Cl(A) = \{P_1, P_2\}$ and $Cl(B) = \{P_1, P_3\}$, we have four *PCl*s, namely $(Cl(A), P_1)$, $(Cl(A), P_2)$, $(Cl(B), P_1)$, and $(Cl(B), P_3)$. Similarly, each *PCo* represents a unique protein–complex relationship.

A *PCl* is considered to be matched if its protein belongs to some complex that matches its cluster. The definition of a match between a predicted cluster and a complex was described earlier in this section. $\text{Precision}_{\text{protein}}$ is defined as

$$\text{Precision}_{\text{protein}} = \frac{|matched_{PCl}|}{|predicted_{PCl}|}. \tag{9}$$

A *PCo* is considered to be matched if its protein belongs to some cluster that matches its complex. $\text{Recall}_{\text{protein}}$ is defined as

$$\text{Recall}_{\text{protein}} = \frac{|matched_{PCo}|}{|known_{PCo}|}. \tag{10}$$

5. Results

5.1. *Parameter determination*

The optimal parameters for RNSC, MCODE, and MCL algorithms are given by Brohée and van Helden[6] (Table 2).

There are two tunable parameters in our experiments: $\text{FS-Weight}_{\text{min}}$ and ICD_{min}. $\text{FS-Weight}_{\text{min}}$ determines the FS-Weight threshold for filtering out level-1 and level-2 interactions. ICD_{min} determines the intercluster density threshold, for which two clusters are allowed to merge during clustering for the PCP algorithm. Based on PPI[Combined] and PC_{2004}, we use level-1 interactions (without any filtering) to determine the *ICD* threshold. The FS-Weight threshold is determined on the same dataset using the PCP algorithm.

5.1.1. *ICD threshold*

We first vary ICD_{min}, the intercluster density threshold for merging clusters, between 0.1 and 0.5, and perform the predictions. The corresponding precision

Table 2. Optimal parameters for RNSC, MCODE, and MCL algorithms.

Algorithm	Parameter	Optimal value
RNSC	No. of experiments	3
	Tabu length	50
	Scaled stopping tolerance	15
MCODE	Depth	100
	Node score %	0
	Haircut	True
	Fluff	False
	% of complex fluffing	0.2
MCL	Inflation	1.8

and recall of the predictions are shown in Fig. 3(a). A lower $ICD_{\min}$ results in more clusters being merged and vice versa. We find that $ICD_{\min} = 0.1$ yields the best precision against recall and use this for the rest of our experiments.

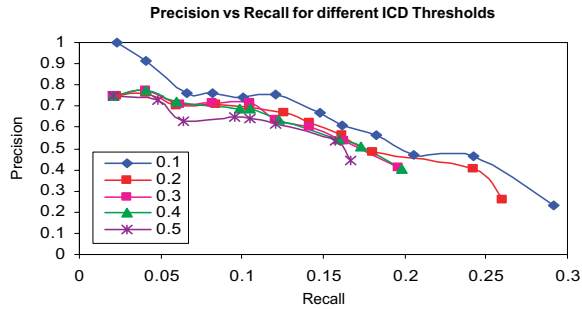
5.1.2. *FS-Weight threshold*

Chua *et al.*[10] showed that filtering level-1 and level-2 interactions with an FS-Weight threshold of 0.2 resulted in interactions that have a significantly higher likelihood of sharing functions. Here, we perform protein complex prediction using the PCP algorithm with a range of FS-Weight$_{\min}$ values to determine which value can yield the best prediction performance. The $ICD_{\min}$ is set to 0.1. The corresponding precision and recall of the predictions are shown in Fig. 3(b). We find that FS-Weight$_{\min} = 0.4$ yields the best precision against recall, and use this for the rest of our experiments.

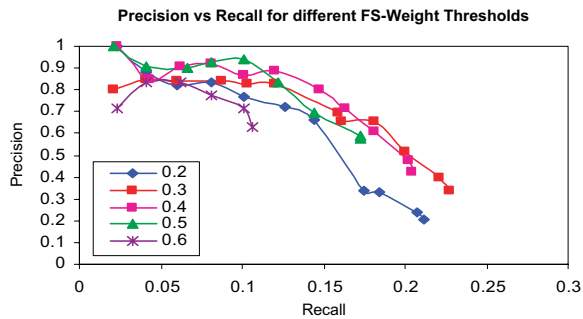
5.2. *Introduction of indirect neighbors*

The introduction of indirect neighbors is the key part of our analysis in this paper. To evaluate the performance of this process, we transform the original PPI network in different ways: (1) all level-1 interactions; (2) all level-1 and level-2 interactions; (3) all level-1 interactions, and level-2 interactions with FS-Weight $\geq$ FS-Weight$_{\min}$; and (4) level-1 and level-2 interactions with FS-Weight $\geq$ FS-Weight$_{\min}$. For (2), due to the large number of level-2 interactions, results can only be obtained for MCL and RNSC. For example, on PPI[Combined], there are 20,461 level-1 interactions. With the introduction of level-2 interactions, the number of interactions increases to 404,511. After filtering level-2 interactions based on FS-Weight, we have 23,356 interactions. Finally, upon filtering both level-1 and level-2 interactions, we are left with only 7,303 interactions.

If two proteins in an interaction belong to some common known complex, we define the interaction as an intracomplex interaction. To justify our intuition for using level-2 interactions and FS-Weight for complex prediction, we compute the fraction of interactions in the four transformed networks that are intracomplex



(a)



(b)

Fig. 3. Effect of (a) *ICD* threshold and (b) FS-Weight threshold on precision and recall values for the PPI[Combined] dataset.

interactions. Since proteins are clustered based on interactions, a higher fraction of intracomplex interactions will naturally yield more accurately predicted clusters. In Fig. 4, we present the corresponding fractions for two PPI networks, PPI[Combined] and PPI[BioGRID], using the known protein complexes in PC_{2004}. We observe that the fraction of intracomplex interactions does not change significantly after adding filtered level-2 interactions into the network. However, if both level-1 and level-2 interactions are filtered, the fraction of intracomplex interactions becomes significantly higher. Without any filtering, level-2 interactions will contain too many false positives to be useful, as reflected by the very small fraction of intracomplex interactions. This is consistent with the findings for function similarity in Chua *et al.*[10] From the observations, we believe that using a PPI network with filtered level-1 and level-2 interactions would yield the best results for protein complex prediction.

5.3. *Comparison with existing approaches*

We compared clusters predicted using four clustering algorithms — MCL, RNSC, MCODE, and PCP — on the datasets including the six PPI datasets as well as

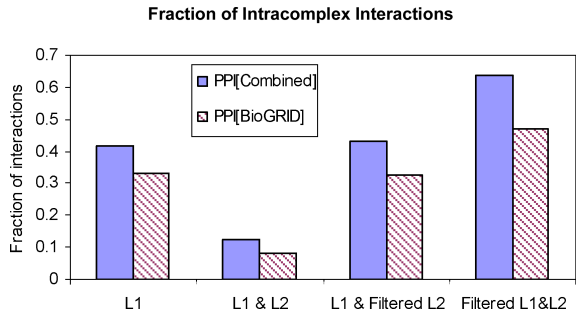


Fig. 4. Fraction of intracomplex interactions with nodes sharing some complex membership for different PPI networks.

PPI[Combined] and PPI[BioGRID]. PC_{2004} was used to represent a real protein complex against which the results from these algorithms were validated.

Table 3 summarizes some general characteristics of clusters predicted by the four clustering algorithms. Looking at the number of complexes in the six PPI networks from GRID, as well as that of PPI[Combined], we observe that most of the complexes in these six networks are overlapping. Using only level-1 interactions, clusters predicted by MCODE are larger than those predicted by other algorithms. Looking at the PPI[BioGRID] and PPI[Combined] datasets, we can see that the number of edges in the network increases drastically when we introduce level-2 interactions. However, after filtering both level-1 and level-2 interactions using FS-Weight, the number of edges reduces substantially to become even lower than the number of original level-1 interactions. We also observed that with the introduction of filtered level-2 interactions, the number of predicted clusters generally decreases while average cluster sizes increase. This is due to greater connectivity in the graph, since more edges are added among the same number of nodes. We also observe that the average cluster sizes of clusters predicted by the MCODE and MCL algorithms are larger than those predicted by the RNSC and PCP algorithms. After filtering both level-1 and level-2 interactions using FS-Weight, all algorithms produce less clusters. With the exception of MCODE, the average cluster sizes of clusters predicted by the various algorithms are also larger.

We have also studied the average density of the clusters predicted by the four different algorithms using the different networks. Generally, all algorithms predict clusters with the highest density using only level-1 interactions, followed by using level-1 and filtered level-2 interactions. Using filtered level-1 and level-2 interactions results in clusters of lower density. When level-1 and level-2 interactions without filtering are used, the clusters found have the lowest density. RNSC yields clusters with the highest density, followed by MCODE, PCP, and MCL. Interestingly, we found that the average density of real protein complexes is quite low, around 0.55, which suggests that the density of predicted clusters does not correlate with prediction accuracy.

Table 3. The features of the datasets, and the features of the clusters that are predicted by different algorithms. The original PPI network is transformed into 4 settings: (1) all level-1 interactions; (2) all level-1 and level-2 interactions; (3) all level-1 interactions, and level-2 interactions with FS-Weight $\geq$ FS-Weight$_{\min}$; and (4) level-1 and level-2 interactions with FS-Weight $\geq$ FS-Weight$_{\min}$.

						No. of clusters				Avg. cluster size			
Datasets	No. of complexes	Avg. complex size	Nodes	Setting	Edges	RNSC	MCODE	MCL	PCP	RNSC	MCODE	MCL	PCP
Gavin *et al.*[14]	**694**	7.07	1352	(1)	3210	**712**	**42**	**212**	**462**	1.90	3.90	6.38	2.93
Gavin *et al.*[15]	**716**	6.78	1430	(1)	6531	**611**	**144**	**189**	**320**	2.34	10.19	7.57	4.47
Ho *et al.*[13]	**763**	7.67	1564	(1)	3599	**995**	**10**	**314**	**547**	1.57	11.60	4.98	2.86
Krogan *et al.*[16]	**749**	4.95	2934	(1)	3959	**1764**	**20**	**630**	**1338**	1.66	3.80	4.66	2.19
Ito *et al.*[17]	**795**	7.05	2675	(1)	7084	**1387**	**68**	**544**	**871**	1.93	6.72	4.92	3.07
Uetz *et al.*[1]	**545**	2.81	909	(1)	822	**490**	**10**	**288**	**321**	1.89	3.4	3.22	2.83
PPI[Combined]	**815**	8.80	4672	(1)	20461	**2332**	**121**	**936**	**1537**	2.00	5.75	4.99	3.04
				(2)	404511	**874**	—	**209**	—	5.34	—	22.35	—
				(3)	23356	**2233**	**120**	**720**	**1499**	2.09	6.48	6.49	3.12
				(4)	7303	**699**	**92**	**259**	**417**	2.44	5.83	6.59	4.09
PPI[BioGRID]	**815**	8.82	5036	(1)	27560	**2404**	**152**	**830**	**1764**	2.20	3.98	6.38	2.85
				(2)	649133	**811**	—	**159**	—	6.21	—	31.67	—
				(3)	32383	**2331**	**142**	**681**	**1557**	2.16	5.69	7.40	3.23
				(4)	10514	**901**	**121**	**285**	**555**	2.36	5.51	7.46	3.83

Figure 5 presents the precision-recall analysis of the predictions made by the four algorithms. By varying a threshold on cluster score, we can obtain a range of recall and precision values for the predictions from each algorithm.

From Fig. 5(a) on the PPI[Combined] dataset, we observed that RNSC performs the best in precision and recall on the original network (level-1 interactions). When level-1 interactions are filtered [Fig. 5(c)], the precision and recall of both PCP and MCL algorithms increase, and are better than those of the RNSC algorithm. With the introduction of level-2 interactions [Fig. 5(b)], the precision and

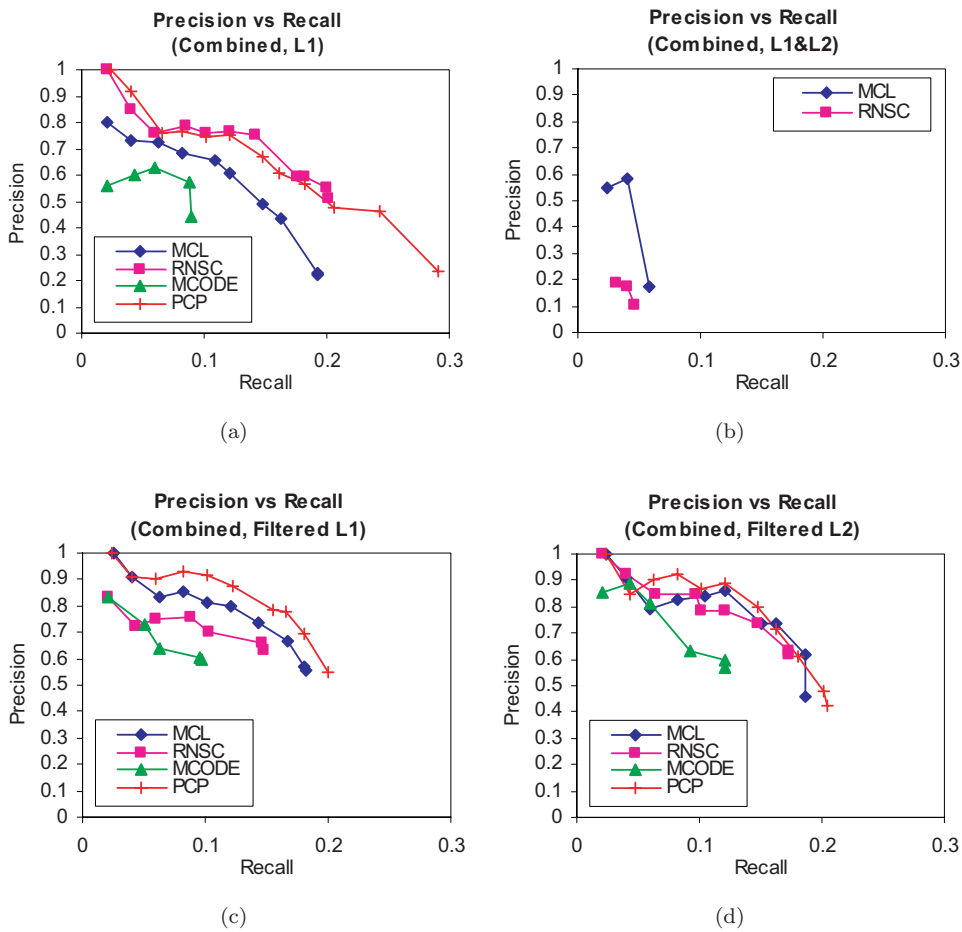
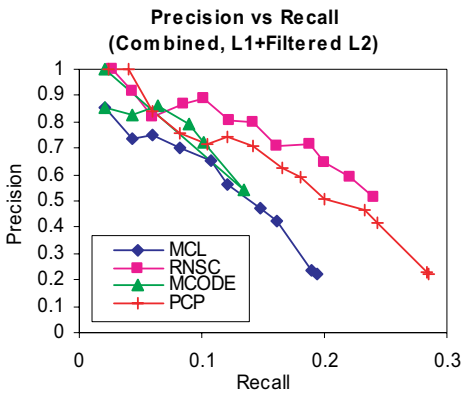
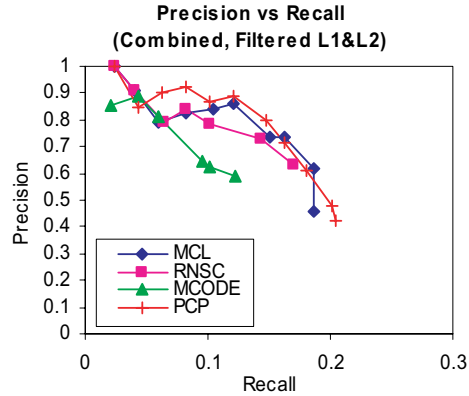


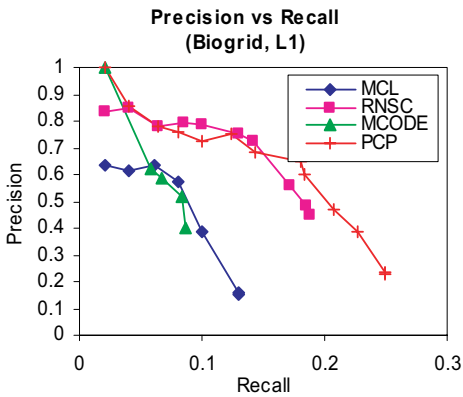
Fig. 5. The precisions and recalls of RNSC, MCODE, MCL, and PCP algorithms on PPI[Combined] with (a) original level-1 interactions, (b) level-1 and level-2 interactions, (c) filtered level-1 interactions, (d) filtered level-2 interactions, (e) original level-1 and filtered level-2 interactions, and (f) filtered level-1 and level-2 interactions; and on PPI[BioGRID] with (g) original level-1 interactions, (h) level-1 and level-2 interactions, (i) filtered level-1 interactions, (j) filtered level-2 interactions, (k) original level-1 and filtered level-2 interactions, and (l) filtered level-1 and level-2 interactions. Results are based on comparison with the PC_{2004} protein complex dataset.



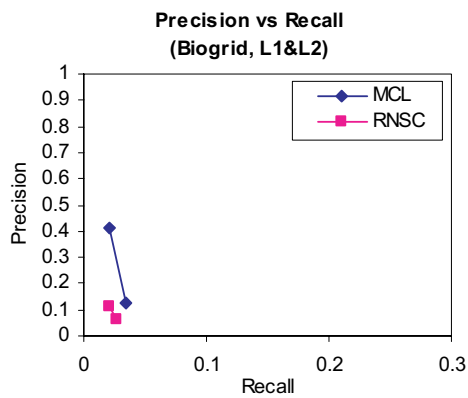
(e)



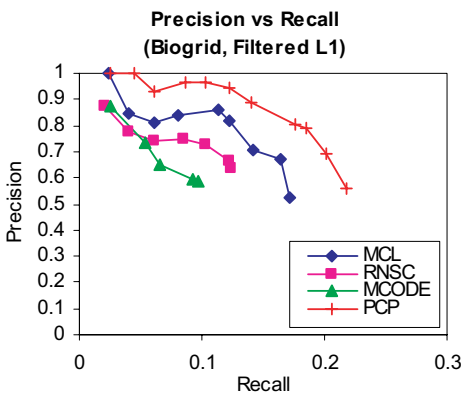
(f)



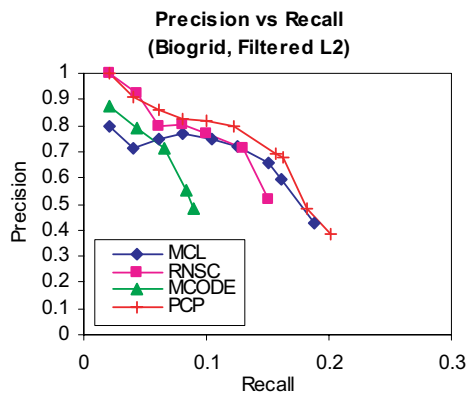
(g)



(h)

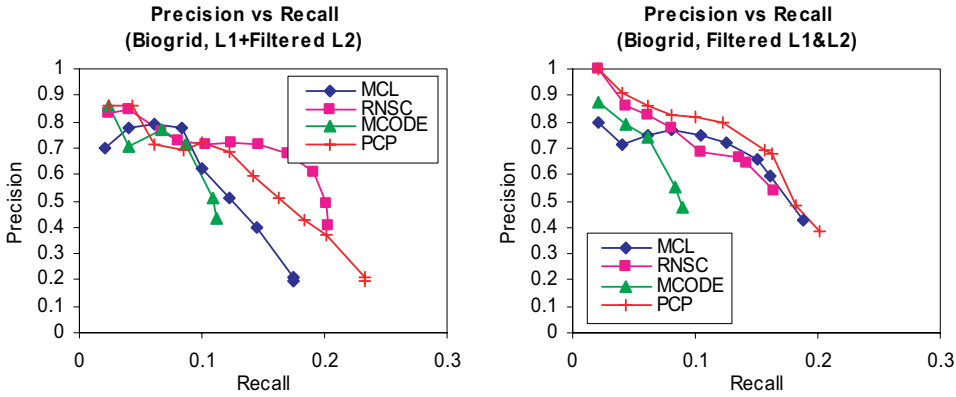


(i)



(j)

Fig. 5. (*Continued*)



(k) (l)

Fig. 5. (*Continued*)

recall decrease. When these level-2 interactions are filtered [Fig. 5(e)], precision and recall are improved in MCODE and RNSC, while PCP and MCL remain almost unchanged. When only filtered level-2 interactions alone are used [Fig. 5(d)], the precision and recall of the four algorithms are already relatively high. When filtered level-1 and level-2 interactions are used together [Fig. 5(f)], all algorithms show a significant improvement in precision, except RNSC. In all of the combinations, PCP with filtered level-1 and level-2 interactions performs the best [Fig. 5(f)]. A similar trend is observed in the bigger PPI[BioGRID] dataset [Figs. 5(g)–5(l)]. Precision is improved in most algorithms with the introduction of filtered level-2 neighbors, and further improvement is achieved when level-1 interactions are also filtered based on FS-Weight. In particular, the performance of MCODE and MCL improved substantially with the introduction of level-2 interactions and FS-Weight filtering. Again, PCP with filtered level-1 and level-2 interactions performs the best [Fig. 5(l)].

Furthermore, to illustrate the contribution of PCP to complex prediction, we compare predictions made by each algorithm natively (i.e. RNSC, MCODE, and MCL on original level-1 interactions against PCP on filtered level-1 and level-2 interactions) in Fig. 6. We observe that PCP outperforms the other algorithms significantly [Figs. 6(a) and 6(b)]. We arrived at similar conclusions using precision-recall analysis based on protein membership assignment [Figs. 6(c) and 6(d)].

For more detailed analysis, we also present the precision and recall graphs for the four algorithms on the six datasets: Gavin *et al.*,[14] Gavin *et al.*,[15] Ho *et al.*,[13] Ito *et al.*,[17] Krogan *et al.*,[16] and Uetz *et al.*[1] (Fig. 7 on original level-1 interactions, and Fig. 8 on filtered level-1 and level-2 interactions).

From Figs. 7 and 8, we find that precision and recall results are better using filtered level-1 and level-2 interactions compared to using level-1 interactions. This is especially true for the PCP and RNSC algorithms. On the original level-1

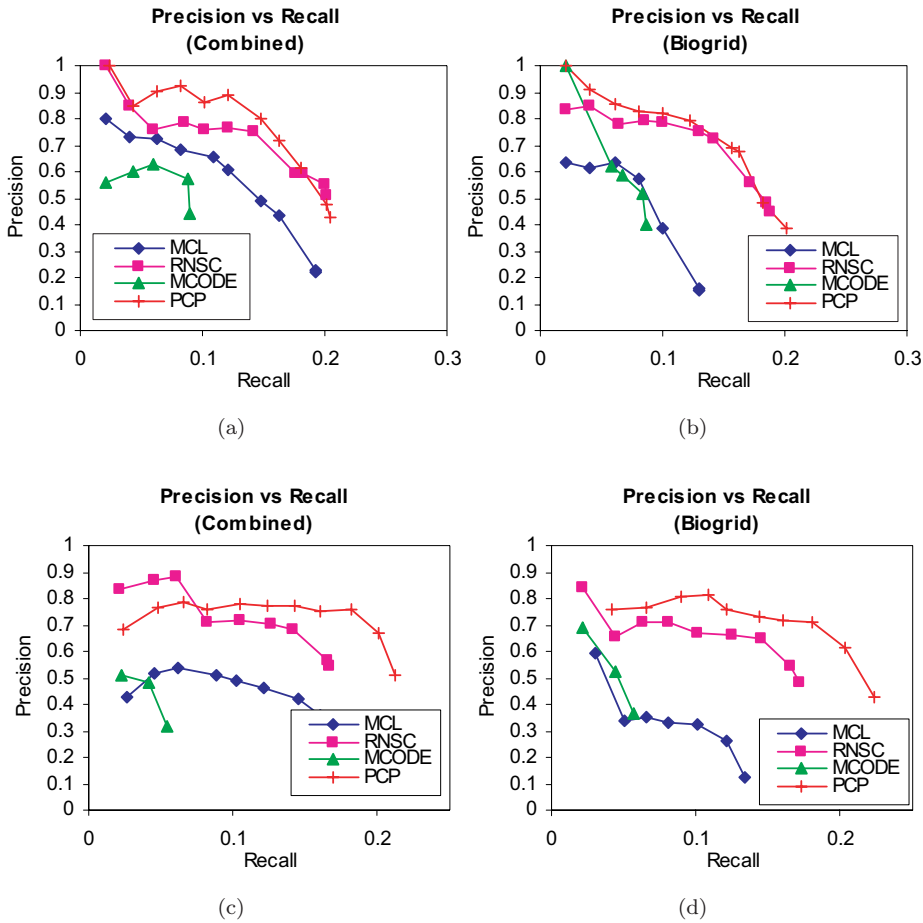


Fig. 6. Precision-recall analysis of RNSC, MCODE, MCL, and PCP algorithms on (a) PPI[Combined] and (b) PPI[BioGRID] using native settings (RNSC, MCODE, and MCL on original level-1 interactions; and PCP on filtered level-1 and level-2 interactions). Precision-recall analysis is based on protein membership assignment on the same predictions on (c) PPI[Combined] and (d) PPI[BioGRID]. Results are based on comparison with the PC_{2004} protein complex dataset.

interactions, the MCL algorithm performs the best on the Gavin *et al.*,[14] Gavin *et al.*,[15] and Ho *et al.*,[13] networks; but on the Krogan *et al.*,[16] network, the RNSC algorithm performs the best.

We also observed that, while the PCP algorithm outperforms the others on PPI[Combined] using filtered level-1 and level-2 interactions (Fig. 5), it only performs comparably to the MCL algorithm on the six individual datasets. This is likely to be a result of MCL's tendency to create bigger clusters. When given smaller interaction networks, it is able to predict correct clusters; but when using the larger combined network, it may have clustered more than one complex together, hence becoming less precise.

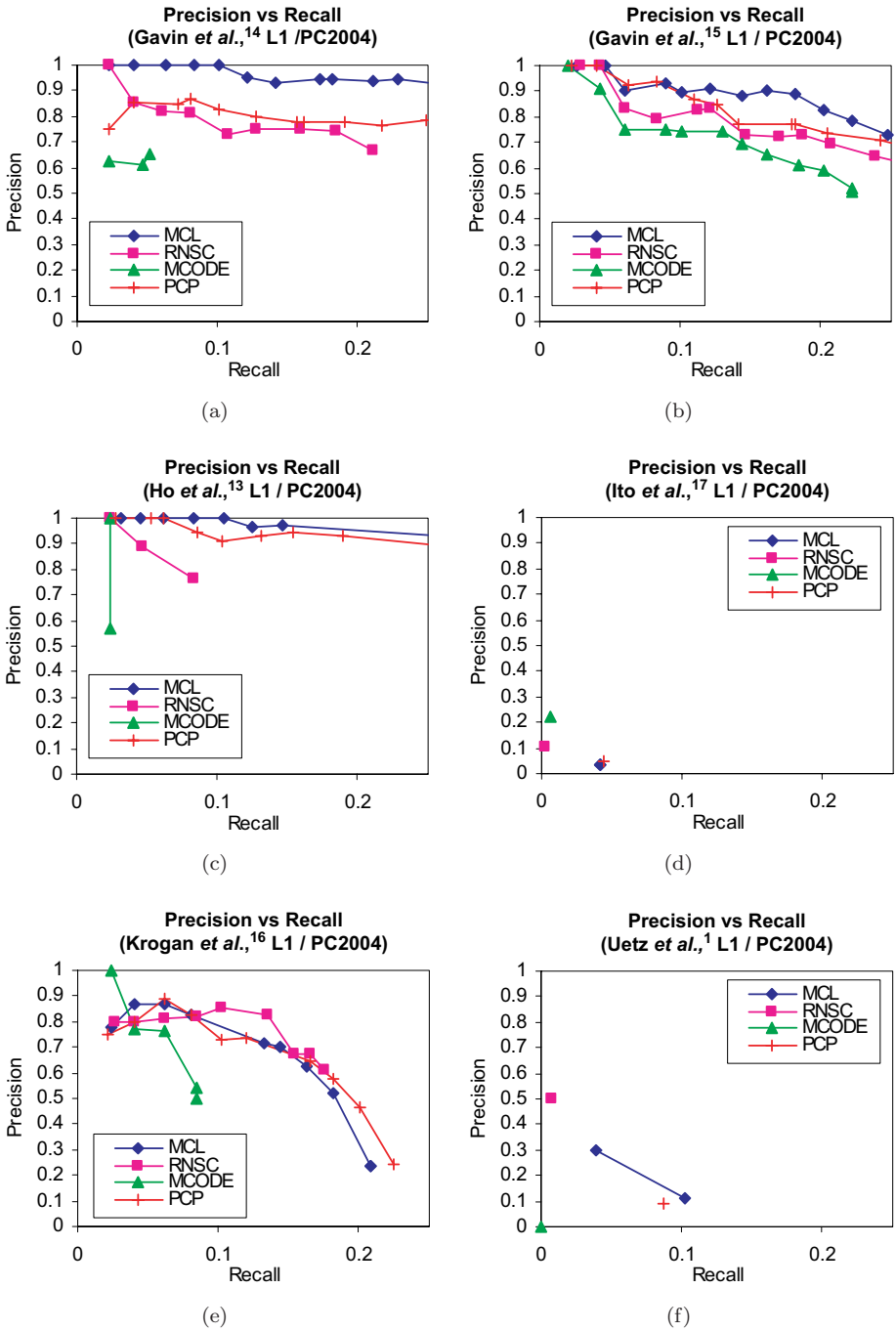


Fig. 7. The precisions and recalls of RNSC, MCODE, MCL, and PCP algorithms on 6 datasets with original level-1 interactions. Results are based on comparison with the PC_{2004} protein complex dataset.

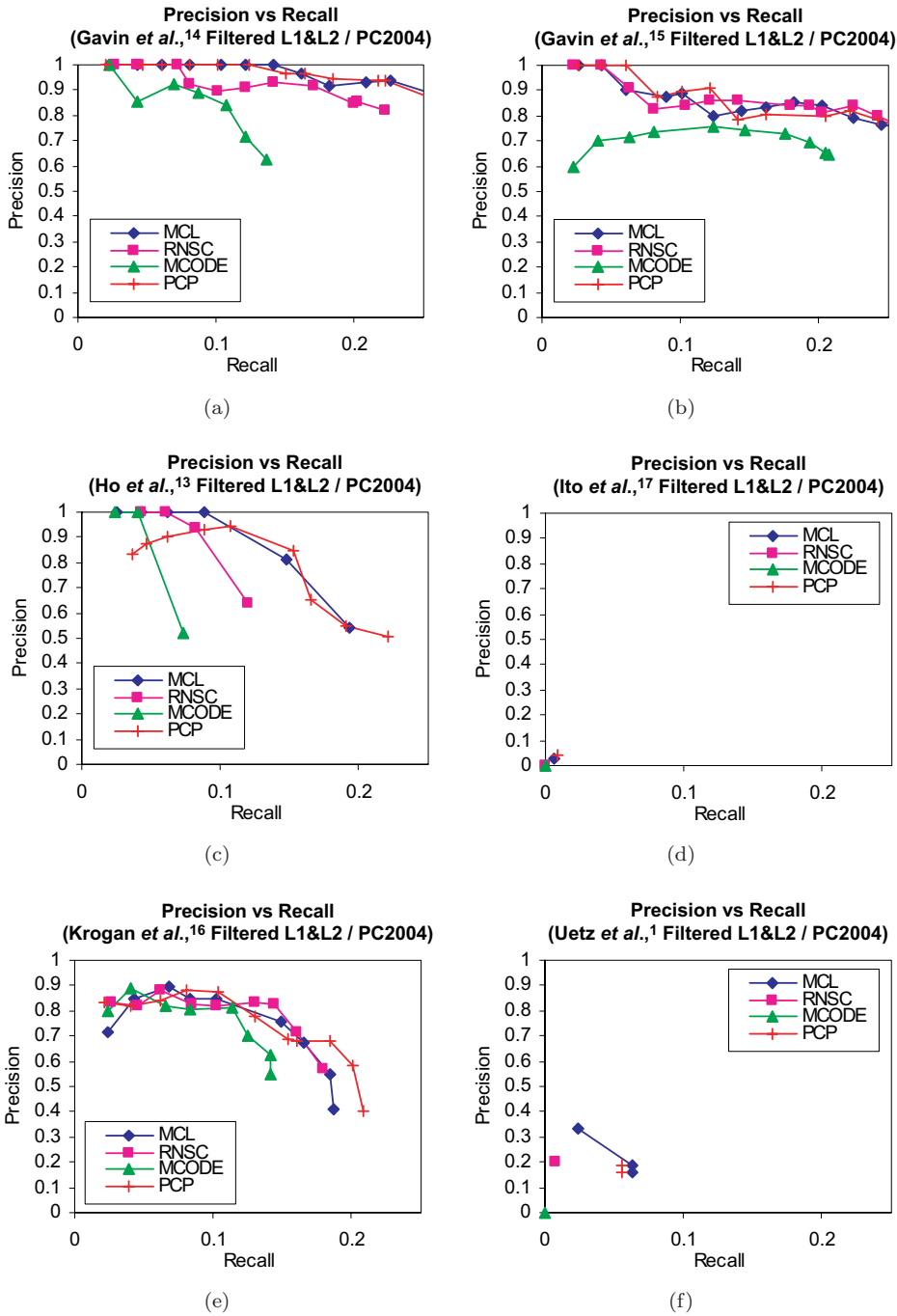


Fig. 8. The precisions and recalls of RNSC, MCODE, MCL, and PCP algorithms on 6 datasets with filtered level-1 and level-2 interactions. Results are based on comparison with the PC_{2004} protein complex dataset.

5.3.1. *Examples of predicted complexes*

We have proposed two new concepts in this paper: the introduction of indirect interactions as a preprocessing step, and the PCP clustering algorithm. To illustrate how these concepts can help to predict protein clusters that better match real complexes, we examine some examples of protein clusters predicted by the PCP algorithm based on the modified network, as well as by the RNSC and MCL algorithms based on the original network, and how they correspond to real protein complexes in the PC_{2004} dataset. Figure 9 shows two examples where PCP can predict protein clusters from PPI[Combined] that match a real complex more precisely than other algorithms. In the first example [Fig. 9(a)], PCP predicted a cluster that matches a four-member protein complex completely, while RNSC's three-member cluster has only one member (YDR121W) that matches the same complex. This is probably due to the fact that members in RNSC's cluster are well connected by level-1 interactions; but by including level-2 interactions and filtering unreliable interactions, their connections are shown not to be strong enough to be in one cluster. Therefore, PCP is able to identify the correct complex. Similarly, the cluster predicted by MCL only overlaps with two members of the complex, while the other six members of the cluster do not belong to the real complex. The second example [Fig. 9(b)] shows a five-member protein cluster predicted by PCP that is a subset of an eight-member protein complex. The best match with the same complex from RNSC is a seven-member cluster, in which only two belong to a subset of the real complex. Although PCP's predicted cluster matched five proteins and MCL also matched five proteins, the latter predicted six proteins that are not in the complex. A closer look will reveal that PCP's cluster members do not have any interactions among them, and this subset of the real protein complex can only be identified by level-2 interactions with the rest of the complex members. PCP is unable to discover the rest of the complex, as their connectivity with the other members is very weak or unknown. The protein YLL011W is missed by PCP because its local topology resulted in a low FS-Weight score; this may be because "hub proteins" like YLL011W are automatically penalized by the FS-Weight score.

5.4. *Validation on newer protein complex data*

A comparison of prediction performance validated against an old protein complex dataset and a newer, more updated standard protein complex dataset can reveal the parameter-independent identification power of the different algorithms. We have previously assessed the RNSC, MCODE, MCL, and PCP algorithms with PC_{2004}. Here, we validate the predicted clusters of PCP and other algorithms against a more recent and more updated protein complex dataset, PC_{2006}. We have used modified PPI networks (PPI[Combined] and PPI[BioGRID]) with filtered level-1 and level-2 interactions, which were shown earlier (Fig. 5) to yield the best performance for most algorithms studied. The corresponding precision-versus-recall graphs are

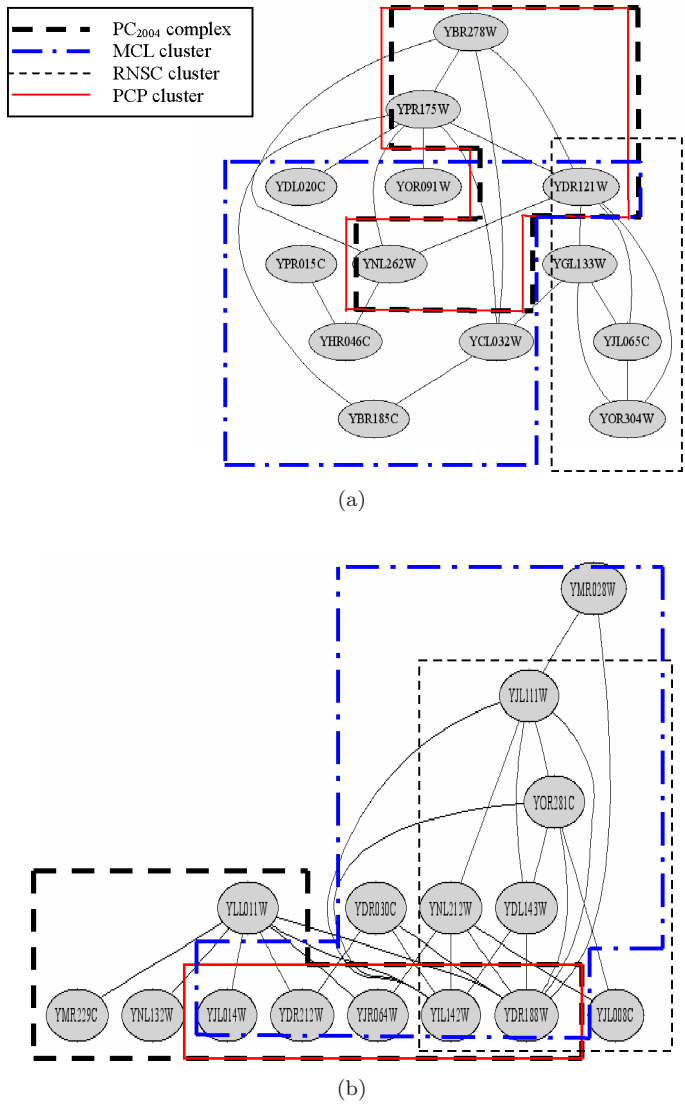


Fig. 9. Example of predicted and matched complexes. Complexes in PC_{2004} as well as the predicted clusters by MCL, RNSC, and PCP are shown in different boxes. (a) For a complex in PC_{2004} of size 4, PCP's cluster matched it perfectly, while MCL's and RNSC's clusters matched 1 and 2 of the proteins in the complex, respectively. (b) In this complex in PC_{2004} of size 8, RNSC's predicted cluster matched only 2 proteins, while PCP's predicted cluster matched 5 proteins; MCL also matched 5 proteins, but predicted 6 proteins that are not in the complex.

shown in Fig. 10. Comparing Fig. 5 against Fig. 10, we find that, against the same recall range, the precision of all algorithms studied increases substantially when validating against PC_{2006} for both PPI network datasets. A significant number of clusters which are predicted by PCP, but have been treated as false positives

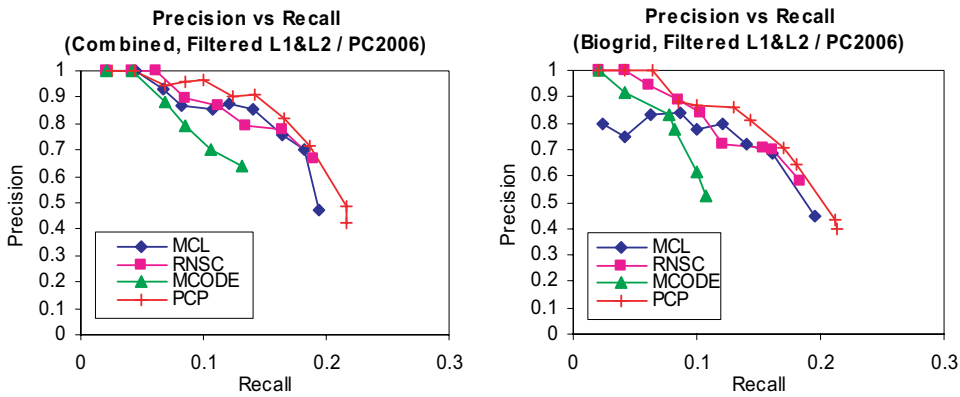


Fig. 10. The precisions and recalls of different algorithms on (a) PPI[Combined] and (b) PPI [BioGRID] with filtered level-1 and level-2 interactions. Results are based on comparison with the PC_{2006} protein complex dataset.

because they cannot be matched against any known complex in PC_{2004}, are now found to match against known complexes in PC_{2006}. This indicates that PCP has a good potential for finding novel protein complexes.

We also present two illustrative examples in Fig. 11 which show that the PCP algorithm predicted novel members to some complexes, which are later verified in the newer complex dataset. In the first example [Fig. 11(a)], PCP predicted a cluster of four proteins. The cluster is found to match well with a real four-member complex from PC_{2004} that contains all but one of the proteins in the predicted cluster. A comparison with PC_{2006}, however, reveals that the predicted cluster matched a real complex in the dataset that contains all four proteins. The protein YFL008W in PC_{2006} has level-1 interactions with the other three proteins, but since the FS-Weight of these interactions are low, PCP did not predict it to be in the same cluster. It is also interesting that in Fig. 11(b), PCP has predicted YHR033W to be in the same cluster as the other five proteins; this is consistent with PC_{2006}, but not PC_{2004}. However, the other five proteins in the new complex are not predicted by PCP, since they do not have any level-1 interaction with other proteins. We think that a more accurate prediction of this protein complex may be achieved by incorporating additional information such as function annotations. Moreover, while the YJR072C protein is predicted by PCP, it is not in the new protein complex. Since the interactions of this protein with YDR212W and YJR064W are present in quite a few other protein complexes,[8] we believe that even though this protein is not in the same complex with other proteins, it should be in the same "function unit"[3] with these proteins. Discriminating "function unit" with protein complex may need additional information such as function annotations.

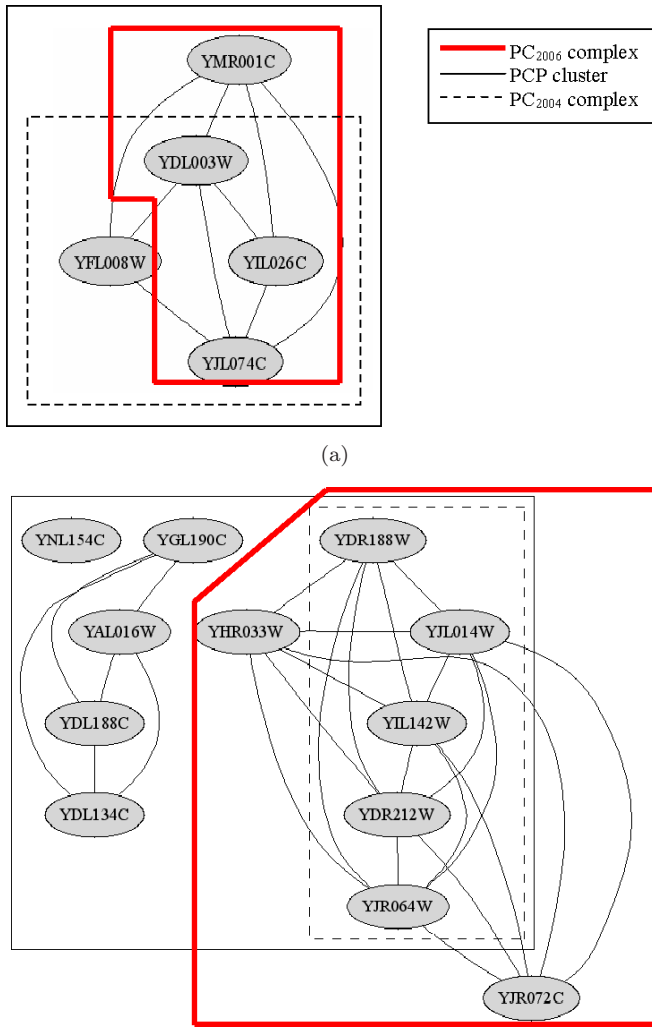


Fig. 11. Examples of predicted and matched complexes based on old and new PPI networks. Complexes in PC_{2004} and PC_{2006} and the predicted PCP clusters are shown in different boxes for comparison. (a) The complex in PC_{2004} is of size 4; while in PC_{2006}, its size is 5. PCP predicted 4 proteins in this complex correctly. (b) This complex is of size 5 in PC_{2004}, for which PCP predicted all 5 proteins correctly. In PC_{2006}, its size is 11, while the PCP algorithm predicted 6 of them correctly.

5.5. *Examples of novel complexes predicted by PCP*

Table 4 lists some clusters predicted by the PCP algorithm that match some complexes in PC_{2006}. These complexes are not present in PC_{2004}, and the matching clusters do not have any matching complexes in PC_{2004}. The Gene Ontology term

Table 4. Selected clusters identified by the PCP algorithm. Proteins with suffix (1) are those that have the GO annotation specified, and proteins with suffix (0) are those that do not have the specified GO annotation. Cluster members in bold are also members of the matching complex.

Complex	Cluster	GO annotation	Notes
YDR138W, YHR167W, YML062C, YNL139C, YNL253W	YCL011C(1), YDL084W(1), **YDR138W**(1), **YHR167W**(1), YJL006C(0), **YML062C**(1), YML112W(0), YNL004W(1), **YNL139C**(1), **YNL253W**(1), YOR191W(0)	GO:0051028 (mRNA transport)	5 overlaps Complex Size: 5 Cluster Size: 11
YCR052W, YDR303C, YER025W, YFR037C, YGR275W, YIL126W, YKR008W, YLR033W, YLR321C, YLR357W, YML127W, YMR033W, YMR091C, YPR034W	YCR020W-B(1), **YCR052W**(1), **YDR303C**(1), **YFR037C**(1), YGR056W(1), **YGR275W**(0), YHR056C(1), **YIL126W**(1), **YKR008W**(1), **YLR033W**(1), **YLR321C**(1), **YLR357W**(1), **YML127W**(1), **YMR033W**(1), **YMR091C**(0), **YPR034W**(1)	GO:0006139 (nucleobase, nucleoside, nucleotide, and nucleic acid metabolic process)	13 overlaps Complex Size: 14 Cluster Size: 16
YDR211W, YER025W, YFL039C, YGR083C, YGR159C, YLR291C, YOR260W, YOR361C, YPL237W	**YDR211W**(1), **YER025W**(1), **YGR083C**(1), YJR007W(1), YKR026C(1), YNL265C(1), **YOR260W**(1), **YPL237W**(1)	GO:0044249 (cellular biosynthetic process)	5 overlaps Complex Size: 9 Cluster Size: 8

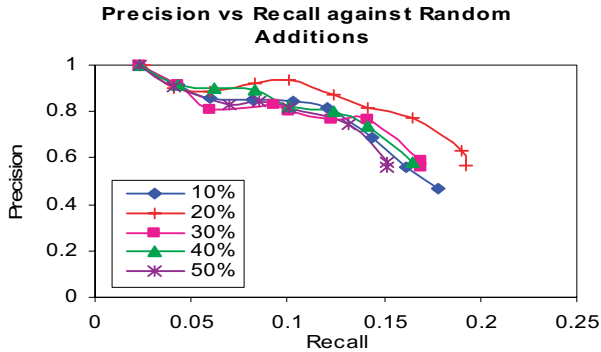
(biological process) that is annotated to the most number of proteins within each predicted cluster is also shown in the table. These examples illustrate the ability of the algorithm to predict novel complexes.

5.6. *Robustness against noise in interaction data*

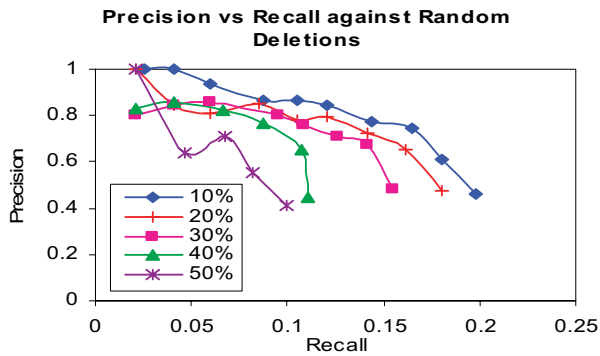
To assess the robustness of the PCP algorithm, we compute the precision and recall of clusters predicted by PCP when different types and amounts of noise are randomly introduced into the PPI[Combined] dataset.

In robustness experiments, noises are usually introduced by swapping edges or randomizing the node labels. However, these methods, which are used in estimating p-values and the uniqueness of PPI motifs, are not a good model for our purpose. We are considering errors produced by high-throughput PPI experiments. In this type of experiment, the errors should be closer to missing (not detected) edges or sticky proteins, which are modeled by random noises. Hence, to simulate such noise, we randomly add, delete, and reroute (delete and add) 10% to 50% of pseudo-interactions in the network. The precision and recall of the predicted clusters on the various perturbed datasets are shown in Fig. 12.

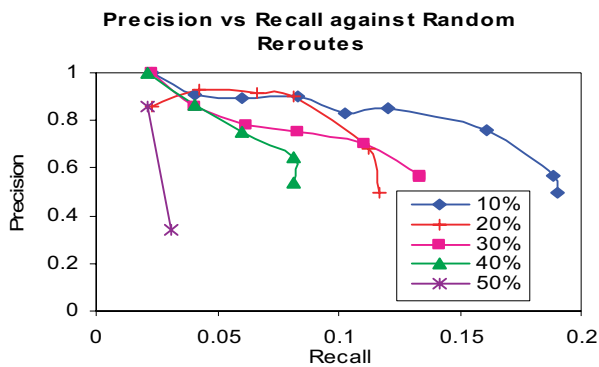
We can see from Fig. 12(a) that the precision against recall of the clusters predicted by PCP remains fairly consistent, even with random additions of interactions up to 50% of the original interactions in PPI[Combined]. This is a clear indication that the PCP algorithm is robust against spurious interactions. The filtering of the PPI network based on FS-Weight removes most of these random additions, and



(a)



(b)



(c)

Fig. 12. The precision and recall of predictions made by the PCP algorithm when different types and amounts of noise are introduced into the reliable PPI network. Three ways of perturbing the network are studied: (a) random addition, (b) random deletion, and (c) random deletion and addition (reroute).

retains only confident interactions for clustering. Random deletion of interactions has a greater impact on clustering performance, as can be seen in Fig. 12(b). This is analogous to a lack of information, leading to a reduction in recall. As FS-Weight is a local topology measure, it becomes less effective when the interaction network becomes very sparse, since there will be insufficient interactions in the local neighborhood to give a confident score. The formulation of the measure will assign low weights in these cases, which will cause many interactions to be filtered. Nonetheless, precision remains high for clusters that can be discovered. A combination of random addition and deletions results in a simultaneous reduction in precision and recall.

5.7. *Improvement of efficiency by heuristic clique finding*

As clique finding is a computationally expensive operation, the introduction of heuristics can help to make the PCP algorithm more scalable to larger interaction networks. Here, we compare the relative performance and efficiency between using PCP (based on exhaustive clique finding) and PCP* (based on heuristic clique finding) on PPI[Combined] and PPI[BioGRID] datasets.

Figure 13 shows the precision versus recall graphs for the predictions made by PCP and PCP*. Both algorithms achieved similar precision and recall performance on the datasets examined [Figs. 13(a) and 13(b)]. Similar conclusions can be derived using precision-recall analysis based on protein membership assignment [Figs. 13(c) and 13(d)]. The use of the heuristic clique-finding approach in place of the exhaustive clique-finding approach did not result in any noticeable difference in the performance of the PCP algorithm.

However, PCP and PCP* do have significant differences in their computational efficiency. Table 5 shows the amount of CPU time taken by PCP and PCP* to make predictions on different datasets. On the six small PPI networks, PCP* is only $0.1 \sim 25$ seconds faster than PCP on original level-1 interactions, and even slightly slower than PCP on some PPI networks with filtered level-1 and level-2 interactions in some cases. However, for the much larger PPI[Combined] and PPI[BioGRID] datasets, PCP* is at least two times faster than PCP on original level-1 interactions, and at least 1.5 times faster than PCP on filtered level-1 and level-2 interactions. The relative speedup achieved using PCP* will tend to increase with the size of the interaction network.

Based on these results, we can conclude that for the datasets examined, the use of the heuristic clique-finding method (PCP* algorithm) can effectively speed up the process of protein complex prediction without significant compromise in precision and recall. This is especially useful for very large PPI networks.

6. Discussion and Conclusion

Protein complexes play an important role in cells, and protein complex discovery from PPI networks remains an interesting and challenging problem in systems

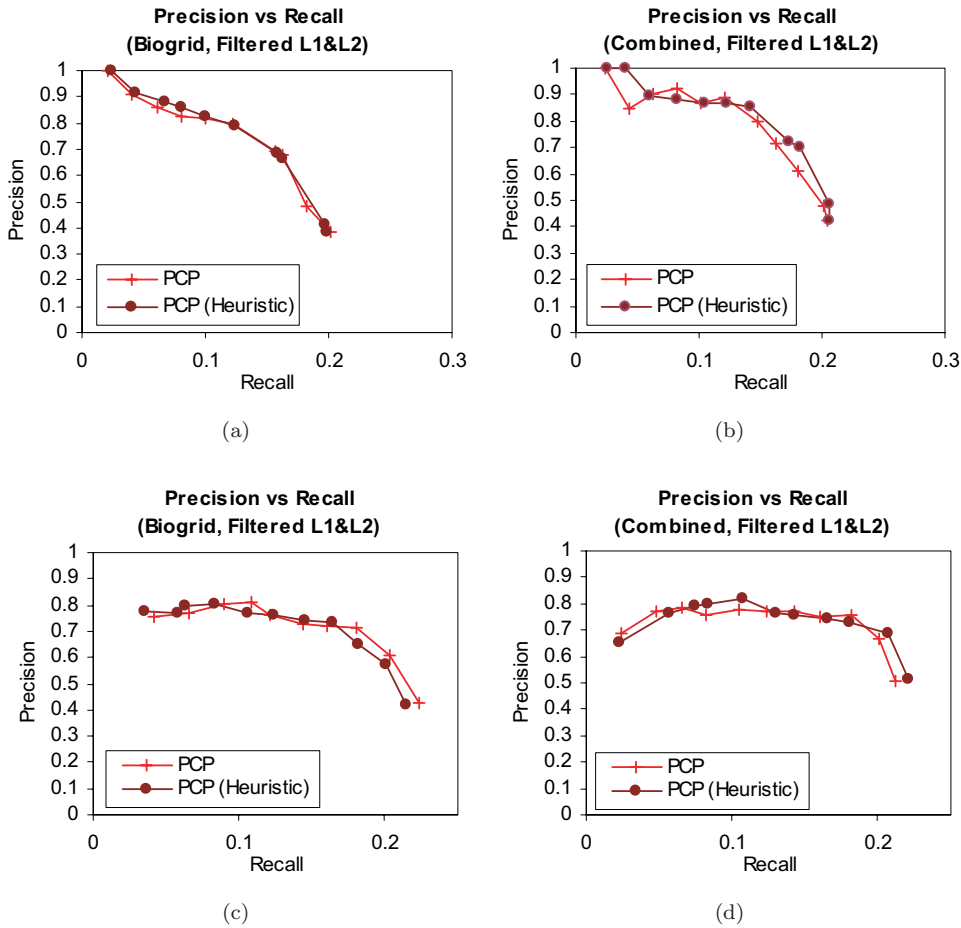


Fig. 13. Precision-recall analysis of PCP and PCP* algorithms on (a) PPI[Combined] and (b) PPI[BioGRID] using filtered level-1 and level-2 interactions; precision-recall analysis based on protein membership assignment on the same predictions on (c) PPI[Combined] and (d) PPI[BioGRID]. Results are based on comparison with the PC_{2004} protein complex dataset.

biology. Challenges to this task include (1) a rapid increase in the size of PPI data, (2) incomplete PPI data, and (3) the fact that PPI data can contain many errors.

In this paper, we proposed a preprocessing step on PPI networks before complex prediction: (1) introduce level-2 interactions; (2) weigh level-1 and level-2 interactions using FS-Weight; and (3) remove interactions with weight lower than a certain threshold. This way, we can alleviate the problem of incompleteness and noise in current PPI networks. From our experiments, we have shown that existing clustering algorithms are able to produce clusters that match protein complexes with significantly higher precision and recall using PPI networks processed in this way.

Table 5. CPU time taken by PCP and PCP* algorithms for different datasets.

			CPU time (s)			
			PCP		PCP*	
Datasets	Nodes	Edges	L1	Filtered L1&L2	L1	Filtered L1&L2
Gavin *et al.*[14]	1352	3210	21.41	7.09	15.9	5.6
Gavin *et al.*[15]	1430	6531	43.2	36.59	25.24	19.31
Ho *et al.*[13]	1564	3599	47.3	6.44	23.09	5.78
Krogan *et al.*[16]	2934	3959	44.44	12.55	34.15	5.71
Ito *et al.*[17]	2675	7084	61.48	6.34	38.1	11.43
Uetz *et al.*[1]	909	822	2.95	0.56	2.84	0.58
PPI[Combined]	4672	20461	395.83	24.89	175.45	16.85
PPI[BioGRID]	5036	27560	831.1	54.45	273.04	29.77

Based on the modified PPI network, we have also proposed the PCP clustering algorithm in which cliques are identified in the network and merged progressively using the "partial clique merging" method. We have compared the PCP algorithm with RNSC, MCODE, and MCL algorithms, and showed that PCP has superior precision and recall in complex prediction. By validating against newer MIPS complex data, we found that PCP can discover novel complex members as well as novel complexes which are only found in the newer complex dataset. Through comprehensive noise analysis, we have further shown that PCP maintains high precision even when used on significantly noisier datasets. We have also proposed a heuristic clique-finding method for the PCP algorithm. Experiments show that this method can effectively speed up the PCP algorithm, without affecting its precision and recall.

There is still one limitation that plagues previous approaches and our current approach: complexes which have subsets of proteins that are not tightly connected to the rest of the complex members cannot be identified, as illustrated in Fig. 11(b). This is inevitable, since clustering methods are highly dependent on interaction density. We are currently studying the possibility of using other biological information to represent a more reliable and complete network of relationships between proteins for complex prediction.

Acknowledgments

We thank Igor Jurisica for providing us the source codes of RNSC. We also thank Sylvian Brohée for providing us with the source codes of MCL and MCODE. This work was supported by a MOE T1 grant and an A*STAR NGS scholarship.

References

1. Uetz P *et al.*, A comprehensive analysis of protein–protein interactions in *Saccharomyces cerevisiae, Nature* **403**(6770):623–627, 2000.
2. Mewes HW, Heumann K, Kaps A, Mayer K, Pfeiffer F, Stocker S, Frishman D, MIPS: A database for genomes and protein sequences, *Nucleic Acids Res* **27**(1):44–48, 1999.

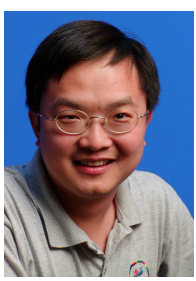
3. Spirin V, Mirny LA, Protein complexes and functional modules in molecular networks, *Proc Natl Acad Sci USA* **100**(21):12123–12128, 2003.
4. King AD, Pržulj N, Jurisica I, Protein complex prediction via cost-based clustering, *Bioinformatics* **20**(17):3013–3020, 2004.
5. Bader GD, Hogue CW, An automated method for finding molecular complexes in large protein interaction networks, *BMC Bioinformatics* **4**(2):27, 2003.
6. Brohée S, van Helden J, Evaluation of clustering algorithms for protein–protein interaction networks, *BMC Bioinformatics* **7**:488, 2006.
7. Pržulj N, Wigle DA, Jurisica I, Functional topology in a network of protein interactions, *Bioinformatics* **20**(3):340–348, 2003.
8. Asthana A, King OD, Gibbons FD, Roth FP, Predicting protein complex membership using probabilistic network reliability, *Genome Res* **14**(6):1170–1175, 2004.
9. van Dongen S, Graph clustering by flow simulation, Ph.D. thesis, University of Utrecht, The Netherlands.
10. Chua HN, Sung WK, Wong L, Exploiting indirect neighbours and topological weight to predict protein function from protein–protein interactions, *Bioinformatics* **22**(13):1623–1630, 2006.
11. Tomita E, Tanaka A, Takahashi H, The worst-case time complexity for generating all maximal cliques and computational experiments, *Theor Comput Sci* **363**:28–42, 2006.
12. Breitkreutz BJ, Stark C, Tyers M, The GRID: The General Repository for Interaction Datasets, *Genome Biol* **4**(3):R23, 2003.
13. Ho Y *et al.*, Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry, *Nature* **415**:180–183, 2002.
14. Gavin AC *et al.*, Functional organization of the yeast proteome by systematic analysis of protein complexes, *Nature* **415**(6868):141–147, 2002.
15. Gavin AC *et al.*, Proteome survey reveals modularity of the yeast cell machinery, *Nature* **440**(7084):631–636, 2006.
16. Krogan NJ *et al.*, Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*, *Nature* **440**(7084):637–643, 2006.
17. Ito T, Chiba T, Ozawa R, Yoshida M, Hattori M, Sakaki Y, A comprehensive two-hybrid analysis to explore the yeast protein interactome, *Proc Natl Acad Sci USA* **98**(8):4569–4574, 2001.
18. Stark C, Breitkreutz BJ, Reguly T, Boucher L, Breitkreutz A, Tyers M, BioGRID: A general repository for interaction datasets, *Nucleic Acids Res* **34**(Database issue):D535–D539, 2006.

Hon Nian Chua obtained his Bachelor's degree in Computer Engineering from the National University of Singapore (NUS) in 2003, and his Ph.D. degree in Bioinformatics from NUS in 2008 under the support of the A*STAR Graduate Scholarship. He is currently working as a research engineer at the Department of Data Mining, Institute for Infocomm Research, A*STAR. His current research interest is in the application of machine learning and graph-based techniques in biological and medical research.

Kang Ning is a research fellow at the Department of Pathology, University of Michigan, USA. He received his B.Sc. in Computer Science from the University of Science and Technology of China in 2003, and his Ph.D. in Computer Science from the National University of Singapore in 2008. His research interests include algorithms for combinatorial problems and computational biology, where he is particularly interested in sequence alignment, pattern discovery, and proteomics. He is currently a member of the American Society for Mass Spectrometry and the Life Sciences Society.

Wing-Kin Sung received both his B.Sc. and Ph.D. degrees in Computer Science from the University of Hong Kong in 1993 and 1998, respectively. He has over 15 years' experience in algorithm and bioinformatics research. He also teaches courses on bioinformatics for both undergraduates and postgraduates. He has been conferred the 2003 Forum on Information Technology (FIT) paper award (Japan), the 2006 National Science Award (Singapore), and the 2008 Young Research Award (National University of Singapore) for his research contributions.

Hon Wai Leong is an Associate Professor at the Department of Computer Science, National University of Singapore. He received his B.Sc. (Hon) degree from the University of Malaya, Malaysia, and his Ph.D. degree from the University of Illinois at Urbana-Champaign, USA. His main research interest is in the design and analysis of practical algorithms for combinatorial optimization problems from diverse application areas including VLSI-CAD, transportation logistics, multimedia systems, and computational biology. In computational biology, his current interests include computational proteomics, fragment assembly, sequencing by hybridization, and genome rearrangement. He is a member of the ACM, IEEE, and ISCB, and is a senior member of the Singapore Computer Society.



Limsoon Wong is a Professor in the School of Computing and the School of Medicine at the National University of Singapore. He is currently working mostly on knowledge discovery technologies, and is especially interested in their application to biomedicine. He serves on the editorial boards of the *Journal of Bioinformatics and Computational Biology* (ICP), *Bioinformatics* (OUP), and *Drug Discovery Today* (Elsevier).