Protein Complex Detection via Effective Integration of Base Clustering Solutions and Co-complex Affinity Scores

Min Wu, Le Ou-Yang, and Xiao-Li Li

Abstract—With the increasing availability of protein interaction data, various computational methods have been developed to predict protein complexes. However, different computational methods may have their own advantages and limitations. Ensemble clustering has thus been studied to minimize the potential bias and risk of individual methods and generate prediction results with better coverage and accuracy. In this paper, we extend the traditional ensemble clustering by taking into account the co-complex affinity scores and present an Ensemble Hierarchical Clustering framework (EnsemHC) to detect protein complexes. First, we construct co-cluster matrices by integrating the clustering results with the co-complex evidences. Second, we sum up the constructed co-cluster matrices to derive a final ensemble matrix via a novel iterative weighting scheme. Finally, we apply the hierarchical clustering to generate protein complexes from the final ensemble matrix. Experimental results demonstrate that our EnsemHC performs better than its base clustering methods and various existing integrative methods. In addition, we also observed that integrating the clusters and co-complex affinity scores from different data sources will improve the prediction performance, e.g., integrating the clusters from TAP data and co-complex affinities from binary PPI data achieved the best performance in our experiments.

Index Terms—Protein complex, ensemble clustering, hierarchical clustering, weighted consensus matrix.

1 Introduction

Protein complexes are of great importance for understanding the structural and functional architecture of the cells [1]. Specifically, protein complexes can help us to understand protein interactions [2], functions, diseases [3], etc. Nowadays, many important protein complexes have been detected by the wet-lab experiments. However, as these small-scale experimental techniques are time-consuming and tedious, there are still many protein complexes that have not been detected. Therefore, we are highly motivated to detect protein complexes with computational methods.

Recently, high-throughput screening (HTS) experiments have provided us with a large amount of protein-protein interaction (PPI) data. It thus becomes more prevalent to detect protein complexes in PPI networks where nodes are proteins and edges are protein interactions. For example, various graph clustering algorithms and tools, such as MCODE [4], CFinder [5], MCL [6], RNSC [7], IPCA [8], COACH [9], HC-PIN [10], ClusterONE [11], DCU [12] and ClusterViz [13], have been designed for detecting protein complexes from PPI networks [14], [15]. On the other hand, another track of methods [16], [17], [18], [19], [20] were proposed to detect protein complexes on tandem affinity purification (TAP) data as two large-scale TAP data were released in 2006.

- *Min Wu is with the Institute for Infocomm Research (I²R), A*Star, Singapore 138632. Email: wumin@i2r.a-star.edu.sg*
- *Le Ou-Yang is with Dept. of Electronic Engineering at City University of Hong Kong. Email: ouyangle@mail2.sysu.edu.cn*
- *Xiao-Li Li is with the Institute for Infocomm Research (I²R), A*Star, Singapore 138632. Email: xlli@i2r.a-star.edu.sg*

Manuscript received April 19, 2005; revised September 17, 2014.

As more genomic and proteomic data are becoming available, data integration is thus a promising strategy to improve the coverage and accuracy for predicting protein complexes. For example, DECAFF [21] and CPredictor [22] exploited functional information of proteins together with PPI data for protein complex detection. MATISSE [23] and TS-OCD [24] integrated gene expression data with PPI data to identify protein complexes. However, the above mentioned methods usually integrate a single data source (e.g., functional annotations or gene expression profiles) with PPI data. Later on, heterogeneous data sources are integrated for protein complex identification [25], [26]. In [25], the authors integrated PPI data with other heterogeneous data sources (i.e., functional association from STRING database and PubMed abstracts) and built a composite protein network. They further weighted each edge in the network using a supervised maximum-likelihood approach and then detected protein complexes from the weighted composite protein network. In [26], the authors integrated four diverse data sources and constructed a final co-complex score matrix for proteins using a supervised learning method. They applied the hierarchical clustering on the final score matrix to generate clusters as protein complexes.

On the other hand, with various methods proposed above for protein complex detection, we are thus able to generate diverse clustering results. Since each computational method is designed to focus on one aspect of the data and neglect other properties of the data, the clustering results generated by different methods may have different qualities and nature [27]. Ensemble clustering, which aims to combine the clustering results, is thus promising to improve the detection for protein complexes

[28], [29], [30]. For example, to effectively utilize the information provided by different clustering results, Greene *et al.* [29] proposed an agglomerative algorithm to produce a disjoint hierarchy of "meta-clusters" and predicted protein complexes from these results; Ou-Yang *et al.* [30] introduced a weighted ensemble clustering method to reconstruct a consensus matrix and identified protein complexes based on the complex information inherent in this consensus matrix.

Given a clustering method, its co-cluster matrix demonstrates which protein pairs are co-clustered in the same clusters/complexes by this method. Consensus matrix is further constructed to measure the co-cluster relationship among proteins by combining the co-cluster matrices of various clustering methods. A simple way to combine the co-cluster matrices is to treat them equally [31]. Such a simple consensus matrix may not be accurate to measure the co-cluster propensity among proteins as different clustering methods generate co-cluster matrices with different quality. Some heuristics, e.g., the intra-cluster distance [28] and Normalized Mutual Information (NMI) scores [32], were then proposed to assign different weights to individual clustering methods and build the weighted consensus matrices. However, these weights for clustering methods are not able to measure the quality of co-cluster matrices accurately as they rely solely on the cluster topology [28] or consistency to other clustering methods [32]. It is thus highly desirable to design a direct and accurate weighting scheme to build the weighted consensus matrices.

In addition, the above co-cluster matrices and consensus matrices are based on the clustering results from various clustering methods. Such result-level integration may miss the underlying co-complex information which exist in the original data sources. It is thus necessary to integrate both the clustering results and the co-complex information (e.g., co-complex affinity scores) directly derived from various data sources to facilitate the detection of protein complexes.

To address the above issues, we propose an Ensemble Hierarchical Clustering framework (EnsemHC) to detect protein complexes. First, we construct co-cluster matrices by leveraging the clustering results and the co-complex evidences from two different data sources, i.e., PPI and TAP data. Second, we integrate the co-cluster matrices to derive a final ensemble matrix via an iterative weighting scheme. Third, we apply the hierarchical clustering [26] to generate protein complexes from the final ensemble matrix. Experimental results demonstrate that our proposed EnsemHC method performs much better than its base clustering methods. EnsemHC also performs better than various existing integrative methods like ENMF [29], EC-BNMF [30] and InteHC [26].

2 METHODS

In this section, we will introduce our proposed EnsemHC method in details.

2.1 Weighted Co-cluster Matrices

Given n proteins and a set of base clustering solutions, $E = \{C^1, C^2, \cdots, C^{|E|}\}$, each clustering solution C^e ($1 \leq e \leq |E|$) groups the n proteins into $|C^e|$ clusters, i.e., $C^e = \{c_1^e, \cdots, c_{|C^e|}^e\}$. With respect to a clustering solution/result C^e, we define its co-cluster matrix M^e in the following Equation (1).

$$M^e(i,j) = \begin{cases} 1, & \text{if } \exists c_k^e \in C^e \text{ such that} \{i,j\} \subseteq c_k^e\ ; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

In the co-cluster matrix M^e, $M^e(i,j) = 1$ means that two proteins i and j are co-clustered in C^e. Currently, we can drive co-complex information directly from the raw data sources. For example, TAP data can measure the co-complex propensity for proteins based on its purification records [33]. Therefore, we can further integrate these data sources together with the co-cluster matrix in Equation (1) to better understand the co-complex relationship among proteins. In Equation (2), we refined the Equation (1) by taking into account the co-complex affinity scores. In particular, τ_{ij} is the co-complex affinity score between proteins i and j, e.g., the C2S score derived from TAP data [19], the FSweight score from PPI networks [26], [34], [35], etc.

$$M^e(i,j) = \begin{cases} \frac{1+\tau_{ij}}{2}, & \text{if } \exists c_k^e \in C^e \text{ such that} \{i,j\} \subseteq c_k^e\ ; \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

2.2 Iterative Weighting for Final Ensemble Matrix

Each clustering solution provides a co-cluster matrix as shown in Equation (2). Thus, we can obtain a final score matrix M via weighted sum, i.e., $M = \sum_e w_e \cdot M^e$. The weight for each clustering solution would be very important to compute the final ensemble matrix and generate protein complexes.

A simple weighting scheme is to equally set the weights for different clustering solutions, i.e., $w_e = 1/|E|, 1 \leq e \leq |E|$ [31]. However, such equal weights may lead to poor results when the base clustering solutions differ much in performance. Another straightforward weighting scheme is to learn the weights using the supervised information (e.g., the benchmark protein complexes). The drawback of such supervised weighting scheme is that it is not applicable to those species without benchmark complexes.

Next, we will introduce a novel weighting scheme, which will evaluate the difference among various clustering solutions in an unsupervised manner.

We assume that there is a set LC of latent clusters. Given a base clustering solution C^e, we will map it to the latent clusters in LC to assess its quality and determine its weight w_e. Generally, a higher percentage of clusters in C^e matching with latent clusters indicates that we will have a higher portion of real co-complex interactions in M^e. Hence, M^e is likely to be more reliable and thus we should assign a higher weight w_e to C^e. The PPV value [19], [26] in Equation (3) measures the percentage of the clusters in C^e that are matched by clusters in LC. Thus, we intuitively set w_e as $PPV(C^e, LC)$ to compute the final ensemble matrix.

$$PPV(C^e, LC) = \frac{\sum_j \max_i |b_i \cap c_j^e|}{\sum_j |\cup_i (b_i \cap c_j^e)|}, \quad (3)$$

where $c_j^e \in C^e$ and $b_i \in LC$.

Algorithm 1 demonstrates our iterative weighting scheme for the base clustering solutions. Initially, we set equal weights for the base clustering solutions and calculate an ensemble matrix M in Line 2. We generate initial latent clusters LC by applying hierarchical clustering on the calculated ensemble matrix M in Line 3 (the hierarchical clustering will be introduced in next subsection). Then, we derive a new weight vector in Line 5 and calculate the Pearson correlation between the new weight vector (W') and the previous weight vector (W) in Line 6. If they are similar enough (e.g., their Pearson correlation is larger than 0.9 in our experiments), we will output W. Otherwise, we will keep calculating the ensemble matrix and generating the latent clusters in Line 8.

Algorithm 1 Iterative Weighting for Base Clustering Solutions

Input: Base clustering solutions $E = \{C^1, C^2, \cdots, C^{|E|}\}$.
Output: Weight vector for base clustering solutions: $W = \{w_1, w_2, \cdots, w_{|E|}\}$.
1: $\forall C^e \in E$, construct co-cluster matrix M^e for C^e;
2: $W = \{\frac{1}{|E|}, \cdots, \frac{1}{|E|}\}$; $M = \frac{1}{|E|}\sum_e M^e$;
// Line 2: set equal weights for base solutions
3: $LC = HC(M)$;
// Line 3: derive latent clusters via hierarchical clustering
4: **while(true)**
5: Calculate W': $\forall C^e \in E$, $w'_e = PPV(C^e, LC)$;
// Line 5: $W' = \{w'_1, w'_2, \cdots, w'_{|E|}\}$
6: **if** $pearson(W, W') < \sigma$
7: $W = W'$;
8: $M = \sum_e w_e \times M^e$; $LC = HC(M)$;
9: **else**
10: **break;**
11: **end while**

Basically, the latent clusters are generated by integrating various base clustering results and they are supposed to be better than the base clustering results. It thus makes sense to assign a higher weight to a base clustering, which is more consistent with the latent clusters. Algorithm 1 will then help us to iteratively obtain a better weighting and further refine the quality of the latent clusters.

2.3 Hierarchical Clustering on Final Ensemble Matrix

The hierarchical clustering algorithm is applied to detect protein complexes on the final ensemble matrix M. First, it considers all singleton proteins as initial clusters. Second, it iteratively merges two clusters with the highest similarity in each iteration. The detailed procedure for the hierarchical clustering is illustrated in Algorithm 2. In addition, the similarity between clusters is defined in Equation (4) and the quality function for a clustering $C = \{c_1, c_2, \cdots, c_n\}$ [26] is defined in Equation (5) as follows.

$$sim(c_i, c_j) = \frac{1}{|c_i| \times |c_j|} \sum_{p \in c_i, q \in c_j} M(p, q), \quad (4)$$

$$Q(C) = \frac{\sum_{i=1}^{n} \frac{1}{\sqrt{|c_i|}} \sum_{p,q \in c_i} M(p, q)}{\sum_{i=1}^{n} \sqrt{|c_i|} \times (|c_i| - 1)}. \quad (5)$$

In the below Algorithm 2, the hierarchical clustering will keep running, i.e., it runs from the start with all individual proteins as clusters to the end with all the proteins as a whole cluster. During this process, we will keep track of the quality scores for the generated clusters and we finally will output the set of clusters with the maximal quality score (Line 13).

Algorithm 2 $HC(M)$: Hierarchical Clustering for Protein Complexes

Input: M, the final ensemble matrix;
L, the set of proteins in a given species (e.g., yeast)
Output: C, the set of predicted protein complexes.
1: $C = \{\{p\} | \forall p \in L\}$, $C_{max} = \phi$, Q_{max} = 0;
2: **while(true)**
3: $(c_i^*, c_j^*) = \arg\max_{c_i, c_j} sim(c_i, c_j)$;
// Line 3: find two most similar clusters
4: $c_{merge} = c_i^* \cup c_j^*$;
// Line 4: merge these two clusters
5: $C = C + \{c_{merge}\} - \{c_i^*\} - \{c_j^*\}$;
// Line 5: remove two original clusters
6: **if** $Q(C) > Q_{max}$
7: $C_{max} = C$, $Q_{max} = Q(C)$;
8: **end if**
9: **for each** $c_k \in C$
10: $sim(c_k, c_{merge}) = \frac{|c_i^*| \times sim(c_i^*, c_k) + |c_j^*| \times sim(c_j^*, c_k)}{|c_i^*| + |c_j^*|}$
// Line 10: update the similarity scores
11: **end for**
12: **end while**
13: $C = C_{max}$;

As we know, the above hierarchical clustering generates non-overlapping clusters. Hence, our EnsemHC has a simple additional process to include overlapping proteins after the hierarchical clustering. Given a PPI network G and a cluster c_k, we will augment c_k and include proteins into c_k, which are in the PPI network G and connect to more than half of the proteins in c_k [9]. For the clusters which are generated by the Algorithm 2 from either PPI data or TAP data, we will augment them using the same PPI network (i.e., DIP data in our experiments).

3 Results

In this section, we first introduce the experimental data and evaluation metrics. Then, we extensively compare our EnsemHC with various methods for detecting protein complexes.

3.1 Experimental data and evaluation metrics

In this study, we perform experiments on two different data sources, i.e., PPI data and TAP data. The PPI data is downloaded from the DIP database [36], which involves with 17,201 interactions among 4,930 proteins. We also collect the clustering results of 10 state-of-the-art methods on DIP data, namely, CMC [35], COACH [9], ClusterONE [11], DPClus [37], MCL [6], MCODE [4], RNSC [7], RRW [38], SPICi [39] and PLW [40]. The TAP data is consolidated from both [33] and [41], with 6,498 purifications involving

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2016.2552176, IEEE/ACM Transactions on Computational Biology and Bioinformatics

2,996 bait proteins and 5,405 prey proteins. Similarly, we collect the predicted complexes of 5 existing methods on TAP data, namely, BT [18], C2S [19], CACHET [20], Hart [16] and Pu [17]. We use PPI clusters and TAP clusters to denote the base clustering results derived from PPI and TAP data, respectively.

We utilize the *sensitivity* (Sn), *positive predictive value* (PPV), *Accuracy* (Acc) [19] and FRAC [11] to evaluate the predicted protein complexes. Given a benchmark complex r_i and a predicted complex c_j, the *sensitivity*, PPV and *Accuracy* are defined in Equation (6).

$$Sn = \frac{\sum_i \max_j T_{i,j}}{\sum_i |r_i|}, PPV = \frac{\sum_j \max_i T_{i,j}}{\sum_j |\cup_i (r_i \cap c_j)|},$$
$$Accuracy = \sqrt{Sn \times PPV}, \quad (6)$$

where $T_{i,j}$ is the number of proteins shared by r_i and c_j, i.e., $|r_i \cap c_j|$. Fraction of matched complexes (i.e., FRAC) [11] is an indicator for prediction coverage, which measures the percentage of benchmark protein complexes that are matched by the predicted protein complexes. Given r_i and c_j, they are matched if $\frac{|r_i \cap c_j|^2}{|r_i||c_j|} \geq \omega$ (ω is usually set to 0.2 and we also fix it to be 0.2 in our experiments). The definition of FRAC is shown in Equation (7), where R is the set of benchmark complexes and P is the set of predicted complexes. In particular, the CYC2008 catalogue [42] with 408 complexes is used as the benchmark for evaluation in this study. All the experimental data and results, as well as a binary executable, are available in our website http://www1.i2r.a-star.edu.sg/%7exlli/EnsemHC/.

$$FRAC = \frac{|\{r_i | r_i \in R \wedge \exists c_j \in P, c_j \ matches \ r_i\}|}{|R|}. \quad (7)$$

In addition, in our experiments, we set τ_{ij} in Equation (2) as 1, FSweight [26] and C2S score [19], respectively. Correspondingly, we have 3 types of co-cluster matrix M^e, i.e., binary co-cluster matrix, co-cluster matrix with FSweight and co-cluster matrix with C2S score. Next, we will show the results from different co-cluster matrices.

3.2 Comparison between Equal Weighting and Iterative Weighting

We first evaluate the performance of equal weighting and our iterative weighting for detecting protein complexes.

Among 10 methods on DIP PPI data, their performance is quite different from each other as shown in Table 1 in Subsection 3.4. For example, PLW and MCL both achieve an accuracy 0.624, while DPClus and MCODE are 0.309 and 0.339, respectively. Our iterative weighting can effectively adjust the weights for individual methods based on their own characteristics (i.e., their PPV values to the latent clusters). As such, we observe that iterative weighting performs much better than equal weighting for PPI clusters as shown in Figure 1.

Figure 2 shows the comparison of the two weighting schemes for TAP clusters. For the co-cluster matrices with C2S scores, iterative weighting performs better than equal weighting as shown in Figures 2(a) and 2(c). For the co-cluster matrix with FSweight scores, iterative weighting

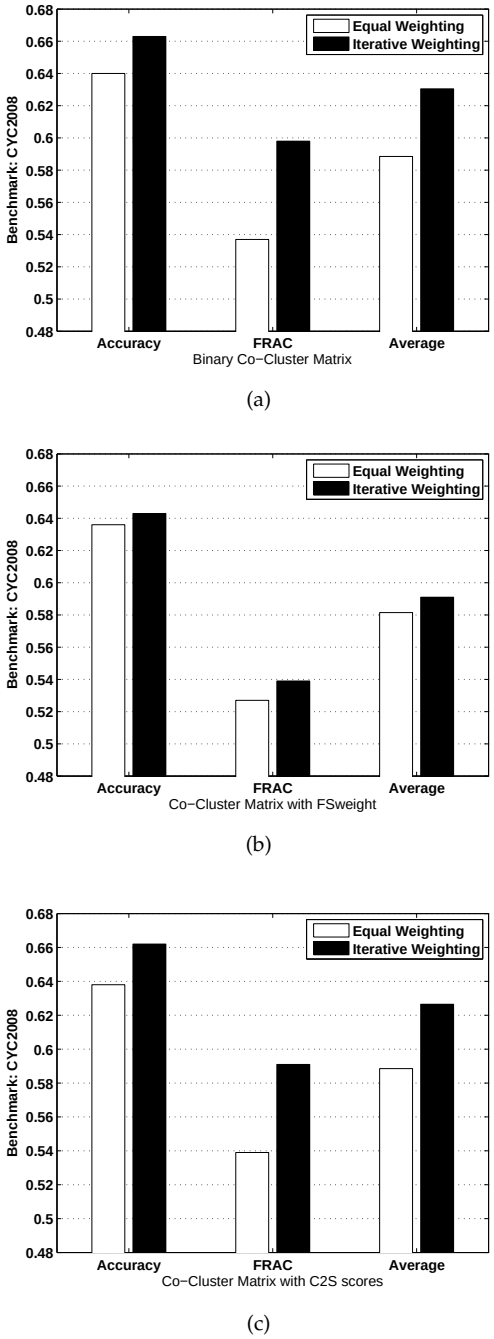
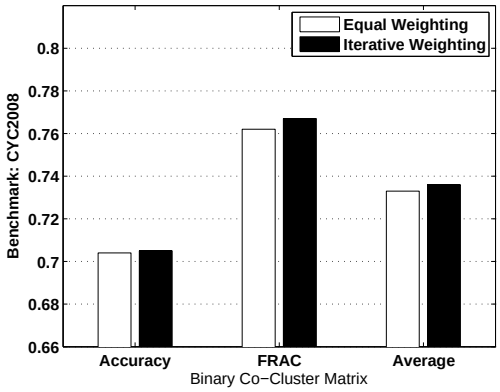


Fig. 1: Comparison of two weighting schemes on PPI clusters.

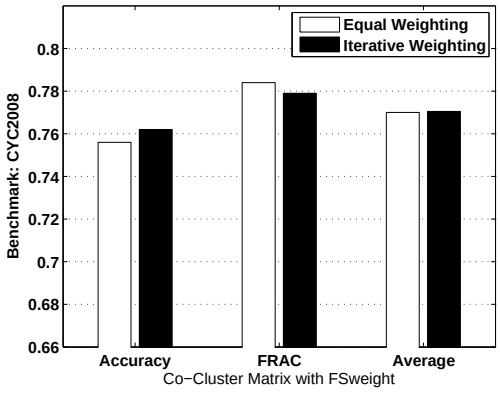
and equal weighting have comparable performance in terms of the average of Accuracy and FRAC. In particular, we also learnt the weights for individual methods in a supervised manner by using the information of benchmark complexes (please refer to our supplementary for more details and results for the supervised weighting scheme). For these 5 methods on TAP data (i.e., BT, C2S, CACHET, Hart and Pu), their weights learnt by the supervised method are 1.03, 1.02, 1.08, 1.05 and 1.0, respectively. That is, the equal weighting is actually very close to the supervised weighting. Overall, our iterative weighting achieves even better results

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2016.2552176, IEEE/ACM Transactions on Computational Biology and Bioinformatics

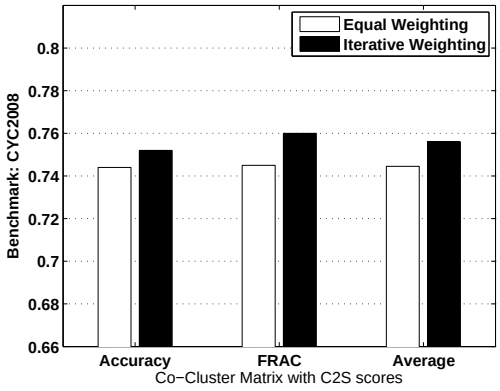
in Figures 2(a) and 2(c) and comparable results in Figure 2(b), demonstrating that it is effective to assign the weights for individual clustering methods.



(a)



(b)



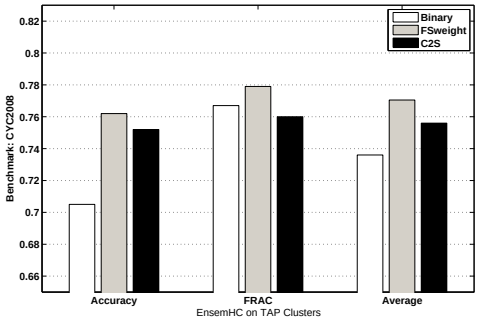
(c)

Fig. 2: Comparison of two weighting schemes on TAP clusters.

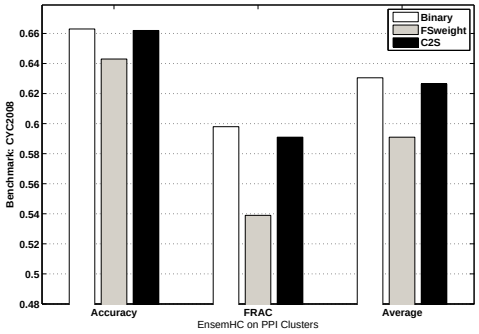
In addition to the equal weighting, we also compare our iterative weighting with an existing weighting scheme based on the Normalized Mutual Information (NMI) [32]. Given a set of clustering solutions $E = \{C^1, C^2, \cdots, C^{|E|}\}$, the NMI score between C^i and C^j demonstrates their consistency. The weight w_e for the clustering solution C^e is defined as its total NMI scores to other clustering solutions (we denote this weighting scheme as NMI weighting). NMI weighting is even worse than equal weighting for PPI clusters. However, our iterative weighting scheme achieves comparable results when its initial weights are set as equal weights and NMI weights as shown in Figure S1 in the supplementary. This observation indicates that our iterative weighting is robust to its initial weights.

3.3 Comparison of various evidences for Co-cluster matrices

Figure 3 compares the results of our EnsemHC on the co-cluster matrices with different co-complex evidences, e.g., FSweight scores based on PPI network topology and C2S scores based on the TAP purifications. We have two interesting findings from Figure 3 as follows.



(a)



(b)

Fig. 3: Comparison of various evidences for co-cluster matrices.

First, the co-cluster matrices with additional evidences perform much better than binary co-cluster matrices for TAP clusters. Meanwhile, the results of EnsemHC on PPI clusters are quite different — binary co-cluster matrices can achieve very good performance. Let's consider a simple example to explain the above results, where the final ensemble matrices are derived from binary co-cluster matrices via equal weighting. Figure 4 shows the distribution of the scores in the final ensemble matrices from TAP clusters (left figure) and PPI clusters (right figure). For the final ensemble matrix from TAP clusters, there are about 2,800 co-cluster pairs with score 1.0 (i.e., predicted by all the 5 methods on TAP data). Recall that the hierarchical clustering in Algorithm 2 will merge two proteins with the maximum

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2016.2552176, IEEE/ACM Transactions on Computational Biology and Bioinformatics

score into a cluster. As we have a large number of pairs with the maximum score 1.0, the merge operation would thus be arbitrary. In this situation, the additional evidences (e.g., FSweight and C2S scores) would help to guide the merge operations towards a better clustering. On the other hand, the number of pairs with score 1.0 in the ensemble matrix derived from PPI clusters is much smaller (less than 200) and the top co-cluster pairs (e.g. top 2,800 pairs) in this matrix already have different scores. This would explain why binary co-cluster matrices for PPI clusters can achieve good performance.

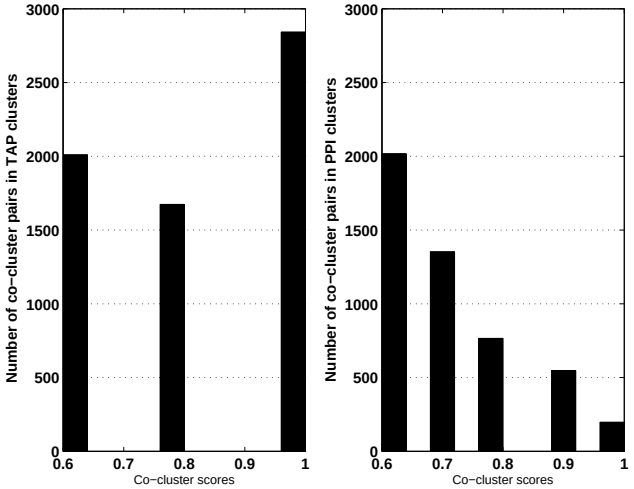


Fig. 4: The distribution of the co-cluster scores in the final ensemble matrices.

Second, the co-cluster matrices integrated with the evidences from other data sources would be more promising. For example, FSweight is better than C2S on TAP clusters in Figure 3(a), while C2S is better than FSweights on PPI clusters in Figure 3(b). In Figure 3(a), co-clusters matrices for TAP clusters would be redundant with C2S scores which are also derived from TAP data. Meanwhile, FSweight scores are likely to be complement to these co-cluster matrices and thus achieve better performance for protein complex detection. This would explain why FSweight is better than C2S on TAP clusters. The case for the PPI clusters in Figure 3(b) is similar. Hereafter, EnsemHC on TAP clusters refers to the case we integrate co-cluster matrices with FSweight scores, while EnsemHC on PPI clusters refers to the case we integrate co-cluster matrices with C2S scores.

3.4 Comparison with various base clustering methods

Table 1 shows the performance of various base methods on DIP PPI data. Here, as mentioned above, the results of our EnsemHC on DIP data are generated based on the co-cluster matrices for PPI clusters with the C2S scores as additional evidence.

Among 10 base clustering methods as shown in Table 1, PLW and MCL achieve the same best Accuracy and COACH has the best FRAC. Our EnsemHC achieves an Accuracy score 0.662, which is 6.09% higher than PLW. Meanwhile, EnsemHC's FRAC is 0.591, 12.14% higher than COACH.

TABLE 1: Comparison between EnsemHC and various methods on DIP data.

Methods	# complexes	# proteins	Acc	FRAC	Avg
EnsemHC	1155	3303	**0.662**	**0.591**	**0.627**
ClusterONE	342	1366	0.557	0.328	0.442
CMC	423	1831	0.613	0.458	0.535
COACH	746	1838	0.617	0.527	0.572
DPClus	301	1177	0.309	0.066	0.188
MCL	600	4101	0.624	0.404	0.514
MCODE	58	754	0.339	0.0931	0.216
RNSC	541	2095	0.605	0.375	0.49
RRW	248	1174	0.538	0.348	0.443
SPICi	412	2113	0.591	0.382	0.487
PLW	576	1747	0.624	0.444	0.534

Table 2 shows the comparison among various methods on TAP data. Similarly, EnsemHC on TAP data refers to the co-cluster matrices derived from TAP clusters with FSweight scores as additional evidence. EnsemHC achieves an Accuracy 0.762 and FRAC 0.779, 0.66% and 23.65% higher than C2S, respectively.

TABLE 2: Comparison between EnsemHC and various methods on TAP data.

Methods	# complexes	# proteins	Acc	FRAC	Avg
EnsemHC	1869	4768	**0.762**	**0.779**	**0.771**
BT	409	1692	0.73	0.598	0.664
C2S	1035	5094	0.757	0.63	0.694
CACHET	449	1110	0.666	0.512	0.589
Hart	390	1689	0.725	0.593	0.659
Pu	400	1913	0.738	0.591	0.665

In addition, we observe that the results of EnsemHC on TAP clusters in Table 2 are much better than that on PPI clusters in Table 1. Given a pair of proteins not co-clustered, we will actually not consider their FSweight or C2S scores in the co-cluster matrices. After such an intersection-based integration in Equation (2), the performance of EnsemHC is still mainly determined by the quality of its input clusters. As TAP clusters in Table 2 have much higher Accuracy and FRAC than PPI clusters in Table 1, it is reasonable that EnsemHC on TAP clusters are much better than that on PPI clusters. In this study, we focus on the co-clustered protein pairs with an intersection-based integration of clustering results and other co-complex affinity scores. In the future, it would be interesting to investigate an union-based integration for protein complex detection.

3.5 Comparison with ensemble clustering methods

Table 3 shows the comparison between our EnsemHC and two ensemble clustering methods, i.e., ENMF [29] and EC-BNMF [30]. The parameter settings for ENMF and EC-BNMF are introduced in our supplementary materials.

TABLE 3: Comparison among EnsemHC, ENMF and EC-BNMF.

Input Data	Methods	Acc	FRAC	Avg
DIP Clusters	EnsemHC	0.662	**0.591**	**0.627**
	EC-BNMF	**0.677**	0.561	0.619
	ENMF	0.650	0.583	0.617
TAP Clusters	EnsemHC	**0.762**	**0.779**	**0.771**
	EC-BNMF	0.737	0.593	0.665
	ENMF	0.710	0.541	0.626

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2016.2552176, IEEE/ACM Transactions on Computational Biology and Bioinformatics

On DIP clusters, EC-BNMF achieves the best Accuracy, while EnsemHC performs the best in term of FRAC and the average of Accuracy and FRAC. On TAP clusters, our EnsemHC achieves the highest Accuracy, FRAC and their average score. For example, EnsemHC on TAP clusters achieves an Accuracy 0.762, which is 3.4% and 7.3% higher than EC-BNMF (0.737) and ENMF (0.710), respectively.

3.6 Comparison with other integrative methods

InteHC [26] is an integrative method which combines 4 data sources, i.e., PPI data, gene expression profiles and gene ontology annotations, for protein complex prediction. In addition, the authors in [25] predicted protein complexes leveraging both data-level integration and result-level integration. They first built a composite network by integrating PPI data, functional associations from STRING database and co-occurrence information for proteins from PubMed abstracts. They applied 6 clustering methods on the composite network and designed a voting-based aggregative strategy to combined their results to generate the final set of protein complexes. Next, we denote the method in [25] as "Combined" and show the comparison among EnsemHC, InteHC and Combined in Table 4 (we used EnsemHC on TAP clusters, which achieved the best performance, to compare with InteHC and Combined).

TABLE 4: Comparison among EnsemHC, Combined and InteHC.

Methods	# complexes	# proteins	Acc	FRAC	Avg
EnsemHC	1869	4768	0.762	**0.779**	**0.771**
InteHC	860	2580	**0.769**	0.711	0.740
Combined	228	1173	0.462	0.25	0.356

As shown in Table 4, EnsemHC performs better than InteHC and Combined in terms of the average of Accuracy and FRAC. We also observed that Combined generates only 228 protein complexes as it requires that each complex should have at least 4 proteins. Meanwhile, the number of complexes predicted by EnsemHC and InteHC is much larger as they generate quite a number of complexes with 2 or 3 proteins. For fair comparison, we further removed those complexes with 2 or 3 proteins for EnsemHC and InteHC. As such, EnsemHC has 261 complexes and InteHC has 246 (i.e., all the three methods have a comparable number of predicted complexes). Figure 5 shows the comparison among them and EnsemHC consistently performs better than InteHC and Combined in term of Accuracy, FRAC and the average.

Lastly, as we mentioned in the Introduction section, both InteHC and Combined integrate heterogeneous data sources using supervised learning methods, while our EnsemHC works in an unsupervised manner. Moreover, InteHC integrates 4 data sources and Combined integrates 3 data sources including a very comprehensive STRING database. Meanwhile, EnsemHC only integrates the TAP clusters with the FSweight scores derived from PPI data. Nevertheless, EnsemHC still perform much better than InteHC and Combined in terms of prediction accuracy and coverage. Therefore, EnsemHC is more promising than InteHC and Combined for integrative detection of protein complexes.

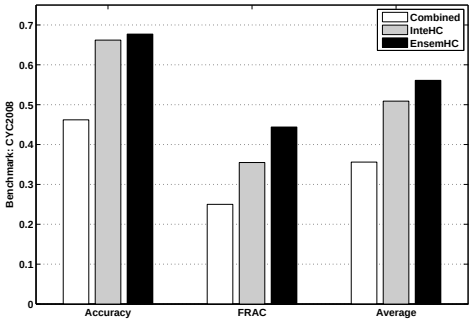


Fig. 5: Comparison among EnsemHC, InteHC and Combined.

3.7 Complexity and running time for EnsemHC

The computational complexity of EnsemHC is $O(K \times N^3)$, where K is the number of times running the hierarchical clustering in Algorithm 1 and N is the dimension of the final ensemble matrix (i.e., the number of proteins in the data). Considering that K is usually small in our study and N is not that huge (N is around 6,000 in yeast species), the running time of EnsemHC is still affordable to predict the complexes in the yeast species. For example, on a PC with 3.4GHz CPU (8 cores) and 8G RAM, it takes about 295 seconds to generate complexes by integrating TAP clusters and FSweight scores, and 247 seconds by integrating DIP PPI clusters and C2S scores. In addition, Table S5 in our supplementary file shows that K is equal to 2 when we integrated 10 basic clustering solutions on DIP PPI data with C2S scores, i.e., we only ran the hierarchical clustering twice in the Algorithm 1.

4 Conclusion

In this paper, we present an Ensemble Hierarchical Clustering framework (EnsemHC) to detect protein complexes. First, we construct co-cluster matrices by leveraging the clustering results and the evidences for co-complex relationships from PPI and TAP data. Second, we integrate the co-cluster matrices to derive a final ensemble matrix via an iterative weighting scheme. Third, we apply the hierarchical clustering [26] to generate protein complexes from the final ensemble matrix. Experimental results demonstrate that our EnsemHC performs much better than its base clustering methods. EnsemHC also performs better than existing integrative methods like ENMF [29], EC-BNMF [30], Combined [25] and InteHC [26].

In this study, we focus on the co-clustered protein pairs with an intersection-based integration of clustering results and other co-complex evidences from PPI and TAP data. Such an intersection-based integration would probably limit the performance of EnsemHC. In the future, it would be interesting to investigate an union-based integration towards better protein complex detection.

REFERENCES

[1] K. Mitra, A.-R. Carvunis, S. K. Ramesh, and T. Ideker, "Integrative approaches for finding modular structure in biological networks," *Nature Reviews Genetics*, vol. 14, no. 10, pp. 719–732, 2013.

[2] H. Yu, A. Paccanaro, V. Trifonov, and M. Gerstein, "Predicting interactions in protein networks by completing defective cliques," *Bioinformatics*, vol. 22, no. 7, pp. 823–829, 2006.

[3] K. Lage, E. O. Karlberg, Z. M. Størling, P. I. Olason, A. G. Pedersen, O. Rigina, A. M. Hinsby, Z. Tümer, F. Pociot, N. Tommerup, Y. Moreau, and S. Brunak, "A human phenome-interactome network of protein complexes implicated in genetic disorders," *Nature Biotechnology*, vol. 25, no. 3, pp. 309–316, 2007.

[4] G. D. Bader and C. W. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," *BMC Bioinformatics*, vol. 4, no. 1, p. 2, 2003.

[5] B. Adamcsek, G. Palla, I. Farkas, I. Derényi, and T. Vicsek, "Cfinder: locating cliques and overlapping modules in biological networks," *Bioinformatics*, vol. 22, no. 8, pp. 1021 – 1023, 2006.

[6] A. J. Enright, S. Van Dongen, and C. A. Ouzounis, "An efficient algorithm for large-scale detection of protein families," *Nucleic Acids Res.*, vol. 30, no. 7, pp. 1575–1584, 2002.

[7] A. King, N. Pržulj, and I. Jurisica, "Protein complex prediction via cost-based clustering," *Bioinformatics*, vol. 20, no. 17, pp. 3013–3020, 2004.

[8] M. Li, J.-e. Chen, J.-x. Wang, B. Hu, and G. Chen, "Modifying the dpclus algorithm for identifying protein complexes based on new topological structures," *BMC bioinformatics*, vol. 9, no. 1, p. 398, 2008.

[9] M. Wu, X. Li, C.-K. Kwoh, and S.-K. Ng, "A core-attachment based method to detect protein complexes in ppi networks," *BMC Bioinformatics*, vol. 10, no. 1, p. 169, 2009.

[10] J. Wang, M. Li, J. Chen, and Y. Pan, "A fast hierarchical clustering algorithm for functional modules discovery in protein interaction networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 8, no. 3, pp. 607–620, 2011.

[11] T. Nepusz, H. Yu, and A. Paccanaro, "Detecting overlapping protein complexes in protein-protein interaction networks," *Nature Methods*, vol. 9, no. 5, pp. 471–472, 2012.

[12] B. Zhao, J. Wang, M. Li, F.-X. Wu, and Y. Pan, "Detecting protein complexes based on uncertain graph model," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 11, no. 3, pp. 486–497, 2014.

[13] J. Wang, J. Zhong, G. Chen, M. Li, F. Wu, and Y. Pan, "Clusterviz: a cytoscape app for clustering analysis of biological network," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 12, no. 4, pp. 815–822, 2015.

[14] X. Li, M. Wu, C.-K. Kwoh, and S.-K. Ng, "Computational approaches for detecting protein complexes from protein interaction networks: a survey," *BMC Genomics*, vol. 11, no. Suppl 1, p. S3, 2010.

[15] J. Ji, A. Zhang, C. Liu, X. Quan, and Z. Liu, "Survey: Functional module detection from protein-protein interaction networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 261–277, 2014.

[16] G. T. Hart, I. Lee, and E. M. Marcotte, "A high-accuracy consensus map of yeast protein complexes reveals modular nature of gene essentiality," *BMC bioinformatics*, vol. 8, no. 1, p. 236, 2007.

[17] S. Pu, J. Vlasblom, A. Emili, J. Greenblatt, and S. J. Wodak, "Identifying functional modules in the physical interactome of saccharomyces cerevisiae," *Proteomics*, vol. 7, no. 6, pp. 944–960, 2007.

[18] C. C. Friedel, J. Krumsiek, and R. Zimmer, "Bootstrapping the interactome: unsupervised identification of protein complexes in yeast," *Journal of Computational Biology*, vol. 16, no. 8, pp. 971–987, 2009.

[19] Z. Xie, C. K. Kwoh, X.-L. Li, and M. Wu, "Construction of co-complex score matrix for protein complex prediction from ap-ms data," *Bioinformatics*, vol. 27, no. 13, pp. i159–i166, 2011.

[20] M. Wu, X.-l. Li, C.-K. Kwoh, S.-K. Ng, and L. Wong, "Discovery of protein complexes with core-attachment structures from tandem affinity purification (tap) data," *Journal of Computational Biology*, vol. 19, no. 9, pp. 1027–1042, 2012.

[21] X. Li, C. Foo, and S. Ng, "Discovering protein complexes in dense reliable neighborhoods of protein interaction networks," in *International Conference on Computational Systems Bioinformatics (CSB)*, 2007, pp. 157–168.

[22] B. Xu and J. Guan, "From function to interaction: a new paradigm for accurately predicting protein complexes based on protein-to-protein interaction networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 4, pp. 616–627, 2014.

[23] I. Ulitsky and R. Shamir, "Identification of functional modules using network topology and high-throughput data," *BMC Systems Biology*, vol. 1, no. 1, p. 8, 2007.

[24] L. Ou-Yang, D.-Q. Dai, X.-L. Li, M. Wu, X.-F. Zhang, and P. Yang, "Detecting temporal protein complexes from dynamic protein-protein interaction networks," *BMC bioinformatics*, vol. 15, no. 1, p. 335, 2014.

[25] C. H. Yong, G. Liu, H. N. Chua, and L. Wong, "Supervised maximum-likelihood weighting of composite protein networks for complex prediction," *BMC systems biology*, vol. 6, no. Suppl 2, p. S13, 2012.

[26] M. Wu, Z. Xie, X. Li, C.-K. Kwoh, and J. Zheng, "Identifying protein complexes from heterogeneous biological data," *Proteins: Structure, Function, and Bioinformatics*, vol. 81, no. 11, pp. 2023–2033, 2013.

[27] J. Song and M. Singh, "How and when should interactome-derived clusters be used to predict functional modules and protein function?" *Bioinformatics*, vol. 25, no. 23, pp. 3143–3150, 2009.

[28] S. Asur, D. Ucar, and S. Parthasarathy, "An ensemble framework for clustering protein–protein interaction networks," *Bioinformatics*, vol. 23, no. 13, pp. i29–i40, 2007.

[29] D. Greene, G. Cagney, N. Krogan, and P. Cunningham, "Ensemble non-negative matrix factorization methods for clustering protein-protein interactions," *Bioinformatics*, vol. 24, no. 15, pp. 1722–1728, 2008.

[30] L. Ou-Yang, D.-Q. Dai, and X.-F. Zhang, "Protein complex detection via weighted ensemble clustering based on bayesian nonnegative matrix factorization," *PloS ONE*, vol. 8, no. 5, p. e62158, 2013.

[31] A. L. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 6, pp. 835–850, 2005.

[32] Y. Yang and K. Chen, "Temporal data clustering via weighted clustering ensemble with different representations," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 23, no. 2, pp. 307–320, 2011.

[33] A.-C. Gavin, P. Aloy, P. Grandi, R. Krause, M. Boesche, and *et al.*, "Proteome survey reveals modularity of the yeast cell machinery," *Nature*, vol. 440, no. 7084, pp. 631–636, 2006.

[34] H. N. Chua, K. Ning, W.-K. Sung, H. W. Leong, and L. Wong, "Using indirect protein–protein interactions for protein complex prediction," *Journal of bioinformatics and computational biology*, vol. 6, no. 03, pp. 435–466, 2008.

[35] G. Liu, L. Wong, and H. N. Chua, "Complex discovery from weighted ppi networks," *Bioinformatics*, vol. 25, no. 15, pp. 1891–1897, 2009.

[36] L. Salwinski, C. S. Miller, A. J. Smith, F. K. Pettit, J. U. Bowie, and D. Eisenberg, "The database of interacting proteins: 2004 update," *Nucleic Acids Res.*, vol. 32, no. suppl 1, pp. D449–D451, 2004.

[37] M. Altaf-Ul-Amin, Y. Shinbo, K. Mihara, K. Kurokawa, and S. Kanaya, "Development and implementation of an algorithm for detection of protein complexes in large interaction networks," *BMC Bioinformatics*, vol. 7, no. 1, p. 207, 2006.

[38] K. Macropol, T. Can, and A. K. Singh, "Rrw: repeated random walks on genome-scale protein networks for local cluster discovery," *BMC bioinformatics*, vol. 10, no. 1, p. 283, 2009.

[39] P. Jiang and M. Singh, "Spici: a fast clustering algorithm for large biological networks," *Bioinformatics*, vol. 26, no. 8, pp. 1105 – 1111, 2010.

[40] D. L. Wong, X.-L. Li, M. Wu, J. Zheng, and S.-K. Ng, "Plw: Probabilistic local walks for detecting protein complexes from protein interaction networks," *BMC genomics*, vol. 14, no. Suppl 5, p. S15, 2013.

[41] N. J. Krogan, G. Cagney, H. Yu, G. Zhong, X. Guo, and *et al*, "Global landscape of protein complexes in the yeast saccharomyces cerevisiae," *Nature*, vol. 440, no. 7084, pp. 637–643, 2006.

[42] S. Pu, J. Wong, B. Turner, E. Cho, and S. J. Wodak, "Up-to-date catalogues of yeast protein complexes," *Nucleic Acids Res.*, vol. 37, no. 3, pp. 825–831, 2009.