The Protein Journal (2020) 39:681–702
https://doi.org/10.1007/s10930-020-09922-z

# Identification of Protein Complexes Based on Core-Attachment Structure and Combination of Centrality Measures and Biological Properties in PPI Weighted Networks

**Abdolkarim Elahi[1] · Seyed Morteza Babamir[1]**

Accepted: 28 September 2020 / Published online: 11 October 2020

**Abstract**

In protein interaction networks, a complex is a group of proteins that causes a biological process to take place. The correct identification of complexes can help to better understand function of cells used for therapeutic purposes, such as drug discoveries. This paper uses core-attachment structure, centrality measures, and biological properties of proteins to identify protein complex with the aim of enhancing prediction accuracy compared to related work. We used the *inherent* organization of complex to the identification in this article, while most methods have not considered such properties. On the other hand, clustering methods, as the common method for identifying complexes in protein interaction networks have been applied. However, we want to propose a method for more accurate identification of complexes in this article. Using this method, we determined the core center of each complex and its attachment proteins using the centrality measures, biological properties and weight density, whereby the weight of each interaction was calculated using the protein information in the gene ontology. In the proposed approach to weighting the network and measuring the importance of proteins, we used our previous work. To compare with other methods, we used datasets DIP, Collins, Krogan, and Human. The results show that the performance of our method was significantly improved, compared to other methods, in terms of detecting the protein complex. Using the p-value concept, we show the biological significance of our predicted complexes. The proposed method could identify an acceptable number of protein complexes, with the highest proportion of biological significance in collaborating on the functional annotation of proteins.

**Keywords** Protein complex · Protein interaction network · Centrality measures · Essential protein · Core and attachment algorithm

## 1 Introduction

Proteins are responsible for many activities and behaviors in the cell. Generally, a specific biological process is conducted during interaction with each other. With the advent of laboratory technologies with mass output, large protein interactions have been created. The protein–protein interaction (PPI) network is a network of physical interactions among proteins [1], in which the proteins are considered as vertices, and the physical interactions between the pairs of proteins are considered as graph edges. PPI networks play a very important role in biological processes, including discrimination, protein folding, signal transduction, transcription and translation [2].

The identification of protein complexes in the PPI network has greatly contributed to the understanding of the biological activity mechanism, the essential functional units and the network architecture of protein interactions. A protein complex is a group of proteins interacting at a specific time and place to perform a specific biological process. To date, various computational methods with different perspectives for clustering in protein–protein interaction networks have been proposed. These methods use laboratory datasets and information to detect protein complexes.

Usually, in these methods, protein–protein interaction networks are modeled using a graph, while a dense subgraph is considered as a protein complex. Based on this, most of these methods use concepts from graph theory. That said,

✉ Seyed Morteza Babamir
babamir@kashanu.ac.ir

[1] Department of Computer, University of Kashan, Kashan, Iran

methods that only try to solve problems using concepts from graph theory are usually not very precise, due to neglecting the biological aspects of the protein complexes. For this reason, in recent years, researchers have been trying to provide better methods by interfering with some biological information.

In general terms, complex detection methods can be grouped into two groups of graph-based methods and methods based on a combination of graph theory and biological properties. The MCL method makes complexes in PPI networks based on random walk simulation [3]. The restricted neighborhood search clustering (RNSC) method first detects a high-density subgraph as a complex, then, by using a functional similarity between the proteins, it abandons the inappropriate subgraphs. The Cfinder [4] and CMC [5] methods find and integrate cliques (complete subgraphs, which are connected by an edge [6]) in order to identify protein complexes. Some methods, such as MCODE, predict condensed subgraphs as protein complexes [7].

According to studies conducted on yeast complexes, protein complexes are composed of two major core and attachment proteins [8]. The core plays the main role, and the attachment proteins play a role in helping the core proteins. There is more interaction in a protein complex between the core proteins than other proteins, with the core having a higher density than the other parts of the complex. Considering the high precision of this class of methods, compared to other methods, our proposed method is based on core and attachment proteins. Srihari et al. proposed a method called MCL-Caw where: (1) only the proteins in generated complexes are categorized in core and attachment proteins based on their connections and (2) other proteins as noisy ones are removed [9]. Peng et al. presented a method called WPNCA (weighted PageRank-Nibble and Core Attachment to separate the weighted PPI network into several connected subgraphs, then, complexes in each subgraph are obtained based on the core-attachment structure [10]. The most well-known methods in this category are Core [11], COACH [12] and PEWCC [13].

The COACH method works in two steps. In the first step, it discovers the highly connected areas of the network, i.e., the primary core then extends these areas using highly connected neighbors. For this purpose, a set of candidate cores is generated for each vertex in the graph, which is aligned with the same cores for maximum density. This process is performed to produce candidate cores from a vertex $v$. Next; it uses the Core-removal algorithm in which, through the division of the subgraph of vertex $v$ ($G_v$) to core-free complexes, continues the work through the connected vertices. If the density of $G_v$ is less than a threshold of user density, these complexes are divided into non-core components. The vertices of the core return back to the subgraph complex that is sufficiently dense. When the Core-removal algorithm is completed, post-processing of the discovered cores is used to create the maximum density. At this point, the candidate cores generated from $v$ are produced. Candidate cores are collected from one vertex only among a general set of primary cores. In the redundancy filtering step, each candidate core is ranked against each initial core by a neighbor closeness measure. After discovering the initial core for each vertex, the primary cores in the predicted protein complexes are expanded. For each initial core, $G=(V_c,E_c)$, a set of direct neighbors of $N_c$ is used. Then, the algorithm adds all vertices of $v \in N_c$, which are connected to more than half of the vertices in $C$. This size is introduced as the closeness of $v$ in $C$ [12]. Problems with this method include the random selection of the central vertex in building the core subgraph during the elementary step, the inadequate selection of attachment proteins, the lack of weight in the interactions between proteins, and the presence of noises in the protein interaction network, that said, these problems are solved by the proposed method.

The PEWCC method also comprises two main steps for determining the reliability of interactions, eliminating noise interactions and discovering complexes using a clustering coefficient. In the first step, a new criterion called PE is introduced to measure the correctness of the interaction between the two proteins. Then, considering the low amount of PE, the unreliable edge of the protein pair is removed from the network. In the second step, to detect the complexes with a weight clustering coefficient for each $v_i$ protein, the neighborhood graph is first created, then the weighted clustering coefficient is calculated, followed by the calculation of the degree of each vertex in the neighborhood graph. Next, sequences of proteins in the neighborhood graph are arranged from minimum to maximum. If the protein $v_i$ with the smallest degree and its interactions from the neighboring graph contain only three proteins, the method is stopped and, sequences of proteins with the highest-weighted clustering factor are returned as the validated protein score [13].

In the proposed method, by combining the centrality measure and biological properties, the initial vertex is chosen more precisely. Then, with a noisy edge filter, the core complex is constructed with a higher density. Further, using more weight density and better selection of attachment proteins, the identification of protein complexes becomes more accurate. In this paper, the input network is weighed using gene ontology. Low reliability of interaction between two proteins is removed. Then, to detect essential proteins as seed at the center of the initial cores, the noisy proteins are filtered. Based on the elemental centers and weight density, cores of complexes and attachment proteins are formed. Of course, we used the two initial steps of the proposed approach from our previous paper, which includes weighting the input network and discovering the initial proteins [14]. In that paper, gene ontology and GFD-Net plugin in Cytoscape software were used for network weighting, removal of false positive interactions and the increase of the accuracy of the detection of essential proteins. We also employ a multiple classifier that includes centrality measures for local and global properties of the network and biological features.

In the following, by filtering noisy interactions and using weight density, the primary cores are formed. Also, using the weight density, we eliminate the redundant primary cores and then, according to their overlap, the set of final cores is obtained. In the next step, attachment proteins are determined using weights of interactions and affinity of an attachment protein to the cores. The proposed method and known methods for detecting protein complexes, such as *MCODE* [7], *DCU* [15], *ClusterOne* [16], *HC-PIN* [17] and *CMC* [5], were applied to datasets (networks) DIP [18], Collins [19] and Krogan [20] and the predicted complexes were compared with real complexes CYC2008 [21] and MIPS [22]. The benchmark dataset MIPS has a hierarchical structure so that complexes may consist of several subcomplexes until five levels.

## 2 Materials and Methods

In this paper, a new method called CEBCOA (CEntrality measure and Biological properties in COre and Attachment) is introduced. This method has the following basic steps:

(1) Calculating the semantic similarity of the interaction of protein pairs based on gene ontology.
(2) Removing interactions with a semantic similarity below a certain limit.
(3) Discovering the essential proteins as a seed for the center of complexes.
(4) Finding the initial cores.
(5) Removing redundancy cores.
(6) Adding attachment proteins to each core to form a predicted complex.

During these steps, we use three metrics precision, recall, and F-measure, which we explain them before dealing with the steps.

*Precision* and *Recall* are defined as Relations 1 and 2 [23] where $\varphi = \{P1, P2 \ldots P_k\}$ is the set of the complexes discovered by a method, $R = \{R1, R2 \ldots R_m\}$ is the set of real complexes, and $\alpha\ (R, P)$ shows the overlap between these two complexes.

$$\text{Precision} = |\{\text{P}|\text{P} \in \varphi, \exists \text{R} \in \Re, \alpha(\text{R}, \text{P}) \geq \theta\}|/|\varphi| \qquad (1)$$

$$\text{Recall} = \{R|R \in \mathfrak{R}, \exists P \in \varphi, \alpha(R, P) \geq \theta\}/(|\mathfrak{R}|) \qquad (2)$$

To determine the amount of overlap between real complexes and the discovered ones, threshold $\theta$ is applied whose value is between 0 and 1. A value of 1 means that the discovered complex *P* and the real complex *R* should have the same and an equal number of proteins. A value of zero means that the discovered complex and the real one have no common and similar proteins. According to [24] $\theta$ is considered to be 0.5. *F-measure* is defined in terms of *Precision* and *Recall* as Relation 3.

$$\text{F-measure} = 2 * \text{Precision} * \text{Recall}/\text{Precision} + \text{Recall} \qquad (3)$$

In fact, F-measure is the mean of the harmonics of *Precision* and *Recall* [25]. As with the previous two metrics, the *F-measure* also has a value between 0 and 1 for different methods where 0 and 1 denote the less and most efficiency of the method, respectively.

### 2.1 Step 1

In this step, we describe (1) semantic similarity and (2) topology similarity and use the combination of these two similarities.

#### 2.1.1 Step 1–1

To calculate the semantic similarity between a pair of proteins based on gene ontology, the networks used in this paper are nondirectional and weighted; with a graph $G = (V, E, W)$, it is shown that $V = \{v_1, v_2, \ldots, v_n\}$ is a set of vertices, $E = \{e_1, e_2, \ldots, e_m\}$ is the set of edges, and $W$ is the weight of each edge. Each edge $(v_i, v_j)$ with weight $W_{i,j}$ shows the power of interaction between vertices $v_i$ and $v_j$ in terms of semantic similarity through gene ontology.

Since there may be a large number of false positive and false negative interactions among proteins, the accuracy of predicting protein complexes may be low. To meet this challenge, we considered the weighting of the PPI networks, which topological methods are usually used to this end. In addition, we applied the biological information of the network (the conversion of a network to a graph was described in Sect. 2–1-1). To weighting a network, we calculated the semantic similarity between the protein pairs based on the gene ontology (GO).

In this study semantic similarity is used as a type of weighting. Since the GFD-Net tool [26] provides the right answer, whether or not the network is sparse, we use it. The number of common annotations of GO of the two proteins is presented as their same functionality and also as their interaction. These calculations are used to weight network datasets using the GFD-Net plugin in the Cytoscape tool. Cytoscape's GFD-Net application has been designed to visualize and analyze functional dissimilarity in protein (gene) networks, which can weight a protein network based on GO and the quantity of their function difference. A gene network may contain one or more biological functions. The GFD-Net assumes that each gene in a network is similar to all the genes connected in the network. Accordingly, GFD-Net selects the most functional common in the network context for each gene so the least semantic dissimilarity is selected among all the genes connected in the gene network. The gene function is achieved by an annotation in GO. In the first step, GFD-Net selects the annotations of GO for each gene. In the second step, a part of the GO graph (DAG) is constructed in which there are genetic dependencies and in the third step, the most common function of each gene is determined. In this step, for each vertex in the GO DAG search space, the nearest annotations of each gene are searched in the input network and a GO term (function) is obtained for each gene. Since more than one-third of protein networks are sparse and small, GFD-Net can perform more accurate calculations of protein similarity and have a global view of the network.

According to [26] Relation (4) is used to compute the average dissimilarity of gen annotations where $R'$ denotes dissimilarity of two gene annotations and is computed as Relation (5). Notations $p(\delta)$ and $p(\gamma)$ as two gene annotations are represented as $g_\propto t_\gamma$ and $g_\beta t_\delta$ in $R'$ where $g_\alpha t_\gamma$ indicates term(annotation) $t_\gamma$ of gene alpha. The set of GO terms that are common to gene $g_i$ is represented as $T(g_i) = g_i t_1, g_i t_2, \ldots, g_i t_n$.

$$S(p) = \frac{\sum_{\forall\delta,\gamma|0<\delta<\gamma<|p|} R'(p(\delta), p(\gamma))}{\sum_{\forall\delta,\gamma|0<\delta<\gamma<|v|} E\left(g_\delta, g_\gamma\right)} \tag{4}$$

Let $V = \{g_1, g_2, \ldots, g_n\}$ be a set of genes and $P(V) = T(g_1) \times T(g_2) \times \cdots \times T(g_n)$ be the set of all possible combinations of GO terms where each $p = g_\alpha t_\gamma \in P$ represents a possible solution and $p\ (\delta)$ denotes the GO terms allocated to gene$g_\delta$. If $E$ denotes the set of network edges and $E[g_\propto, g_\beta]$ does the weight of the edge associated with the genes $g_\propto$ and$g_\beta$, then the value of dissimilarity of the two gene annotations $(R')$ is calculated using Relation 5 [26].

$$R'\left(g_\alpha t_\gamma, g_\beta t_\delta\right) = \begin{cases} 0\ if\ E\left[g_\propto, g_\beta\right] = 0; \\ R\left(g_\alpha t_\gamma, g_\beta t_\delta\right)\ if\ E\left[g_\alpha, g_\beta\right] \neq 0; \end{cases} \tag{5}$$

where, $g_\propto$ and $g_\beta$ represent two genes or proteins, $R$ is the distance between the two annotations (terms of gene ontology) $g_\propto t_\gamma$ and $g_\beta t_\delta$ and $t_\delta$ and $t_\gamma$ are two terms. Value of $R$ is obtained using Relation (6) [26].

$$R\left(t_\alpha, t_\beta\right) = \frac{distance\left(t_\alpha, t_\beta\right)}{2 * depth\left(LCA\left(t_\propto, t_\beta\right)\right) + distance\left(t_\alpha, t_\beta\right)} \tag{6}$$

where:

*Distance* $(t_a, t_b)$ denotes the minimum number of vertices separating two vertices $a$ and $b$,

*Depth* represents the maximum number of vertices between a vertex and the root of the DAG,

*LCA* represents the lowest common ancestor between the two vertices $a$ and $b$,

R has a value between 0 and 1. Based on available knowledge in GO, the values near zero denote more similarity and the values near 1 do more dissimilarity a value near zero.

### 2.1.2 Step 1–2

We used the extended Adamic-Adar relation in the topology similarity. The Adamic-Adar similarity criterion is one of the best methods to measure the reliability of the connection between two selected vertices. This method has been originally proposed to measure the similarity between two web pages. Relations 7 and 8 show the basic [27] and our extended Adamic-Adar relations.

$$AA(u, v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{\log(|\Gamma(z)|)} \tag{7}$$

where $|\Gamma(z)|$ represents the number of common neighbors of two nodes u and v. In the Adamic Adar method, the common neighbors with lower degrees had higher weights. In other words, the small number of neighbors denotes the great intensity of the relation. However, the following condition may occur in the Adamic-Adar method. Suppose two vertices $u$ and $v$ in Fig. 1 have the common vertex z; then some neighboring vertices z (i.e. $w_1 \ldots w_4$) may be neighbors of vertices $u$ and $v$. In this case, the number of these neighbors should be considered as a positive value for the interaction

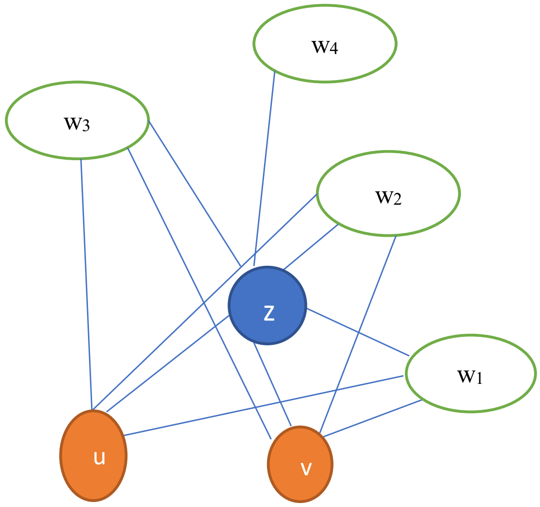


**Fig. 1** A sample of immediate and intermediate common neighbors of two vertices $u$ and $v$ in the Adamic-Adar method

between $u$ and $v$, which is not considered in the original Adamic-Adar relation; we considered such situation in our extended relation (Relation 8).

In our extended relation, we considered a local criterion of similarities of two vertices. Reliability of the interaction can be measured by the number of direct common neighbor/neighbors of two vertices. The extended relation for (1): edge allocation between nodes when there is no edges between them and (2) weight allocation to these edges. These types of nodes may have direct common neighbor/neighbors of z (Relation 8).

$$ST(u,v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{|\Gamma(u) \cap \Gamma(z)| + |\Gamma(v) \cap \Gamma(z)|}{Log|\Gamma(z)|} \tag{8}$$

where $\Gamma(z)$ denotes the set of direct common neighbors of nodes $u$ and $v$ and $|\Gamma(z)|$ represent the number of the neighbors. Considering $\Gamma(z)$, $\Gamma(u) \cap \Gamma(z)$ represents the number of the nodes that are just direct common neighbors of $z$ and $u$ and $\Gamma(v) \cap \Gamma(z)$ does the number of the nodes that are just direct common neighbors $z$ and $v$.

### 2.1.3 Step 1–3

Finally, we use a combination of biological (semantic) and structural information to weight the edge between two vertices $u$ and $v$ (Relation 9).

$$W(u,v) = \frac{(1 - Q(u,v)) + ST(u,v)}{2} \tag{9}$$

$$\text{where } Q(u,v) = \frac{\sum_{\forall u,v|0<u<v|p|} R'(p(u),p(v))}{\sum_{\forall u,v|0<u<v|v|} E[g_u, g_v]} \tag{10}$$

In Relation 9, $W(u, v)$ represents the weight of the edge between $u$ and $v$. This weight is obtained using: (1) the rate of topological similarity of $u$ and $v$ indicated by ($ST(u,v)$ (Relation 9) and (2) the dissimilarity of a set of annotations of $p$, indicated by $Q(u,v)$ (Relation 10). Indeed, $Q(u,v)$ represents the average dissimilarity between the pair terms for the proteins connected to the input network.

Based on the interaction confidence value obtained from Relation 9, the interactions values ($W(u,v)$) with less than threshold $\eta = 0.3$ are not considered. Our measurement to find out a suitable value for this threshold was conducted based on the experiments done through running the ClusterONE algorithm, as one of the best algorithms on network Collins. We applied ClusterONE to network DIP through different thresholds and measured the F-measure values (see Sect. 2 for F-measure). The highest F-measure value was obtained for $\eta = 0.3$ (Fig. 2). Therefore, a new edge is inserted between the pair of nodes with the weight similarity greater than 0.3. Through such similarity, we eliminate false positive edges and insert false negative ones. The graph with a new edge is shown as $G=(v, Enew)$ where $Enew$ denotes set of new edges. Relation 11 shows the number of edges after eliminating false positive edges and inserting new edges in case of false negative edges where $n$, $n_1$, and $n_2$ represent the number of edges of the initial graph, the number of false negative edges and the number of false positive ones, respectively. Algorithm 1 shows the process of constructing the new graph (network) by reconstructing the old one. In this algorithm, $G_{complete}$ is a complete graph of the input network with $\frac{n(n-1)}{n}$ edges where $n$ denotes the number of vertices of the graph.

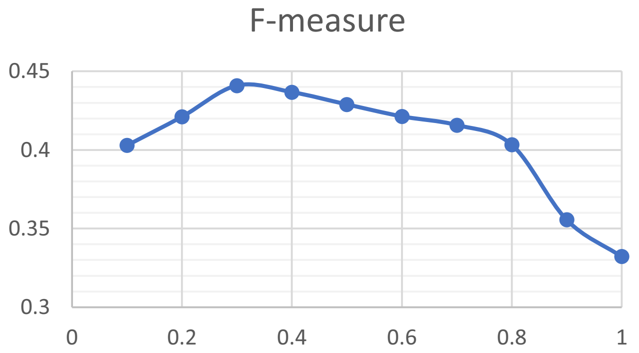
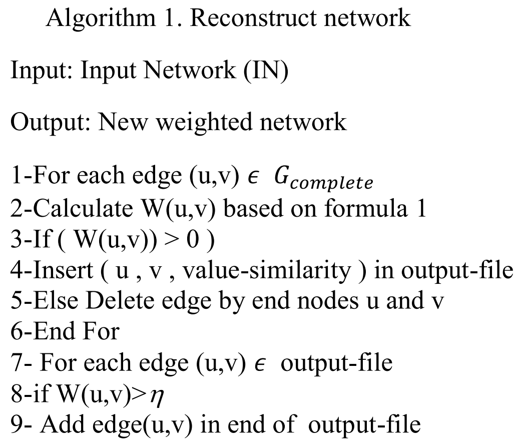


**Fig. 2** The F-measure values (in horizontal axis) obtained in different threshold ($\eta$) values (in vertical axis) by applying ClusterOne to network Collins

$$|Enew| = n + n_1 - n_2 \tag{11}$$



```
Algorithm 1. Reconstruct network

Input: Input Network (IN)

Output: New weighted network

1-For each edge (u,v) $\epsilon$ $G_{complete}$
2-Calculate W(u,v) based on formula 1
3-If ( W(u,v)) > 0 )
4-Insert ( u , v , value-similarity ) in output-file
5-Else Delete edge by end nodes u and v
6-End For
7- For each edge (u,v) $\epsilon$ output-file
8-if W(u,v)>$\eta$
9- Add edge(u,v) in end of output-file
```

## 2.2 Step 2

Now, we have a weighted network that was obtained in step 1. In order to find and eliminate them by considering the similarity measure on the edges of the false-positive interactions. We eliminate the edge and interaction between two vertexes when there is a low similarity between them.

Figure 3 shows the prediction accuracy of our method (CEBCOA) using: (1) different methods of the semantic similarity of the gene ontology, (2) our extended Adamic-Adar Relation, and (3) the combination of (1) and (2). Metrics "P", "R", and "F" indicate precision, recall, and F-measure, respectively (see Sect. 2 for these metrics). As Fig. 3 shows, using the combination method for weighting produces better F-measure and better performance than cases (1) and (2). Figure 3 represents the F-measure value when weighting is carried out by means of: (1) CEBCOA-Go_weight with semantic similarity in the GFD-Net tool environment, (2) CEBCOA-newAA with topological semantics of Adamic-Adar, and (3) CEBCOA-hybrid_weight with the new hybrid formula. As Fig. 3 shows, value of metrics P (precision), R (recall), and F (F-measure), which were obtained by applying CEBCOA-hybrid-weight (indicated by the grey color) to Collins are more than those of the two other methods (see Sect. 2 for description of these metrics).

GFD-Net is an application from Cytoscape designed to visualize and analyze the function dissimilarity of a protein network, which can calculate a protein network based on the gene ontology (GO) and a quantity of its function difference. Figure 4 shows a part of the running GFD-Net plug-in to weigh the DIP network. In the GFD-Net plug-in, by clicking on 'See more organisms' more selections appear in which we select 'Saccharomyces cerevisiae S288c' organism and finally we click the 'Run GFD-Net' button.

In the field of graph theory and network analysis, there are several centrality measures for determining the importance of vertices whose definitions are stated in the following.

### 2.2.1 Definition 1: Weighted Degree Centrality

The sum weights of the edges that connect node *i* to its neighbors; this measure is obtained from the (Relation 12) [28].

$$\mathrm{DCW(i)} = \sum_{\mathrm{j\in N_i}} \mathrm{W_{i,j}} \tag{12}$$

So that *DCW (i)* is the weighted degree centrality and $N_i$ is a set of all of the neighbors of node *i*.

### 2.2.2 Definition 2: Weighted Betweenness Centrality

The average value of the shortest paths that go through node *i* (Relation 13) [28].

$$BCW(i) = \sum_{\mathrm{s}} \sum_{\mathrm{t}} \mathrm{\sigma_{st}(i)/\sigma_{st},\, s \neq t \neq i} \tag{13}$$

where *BCW (i)* is weighted betweenness centrality; $\sigma_{st}$ is the total number of shortest paths between nodes *s*, *t* and $\sigma_{st}(i)$ is the number of shortest paths from node *s* to node *t* that passing through node *i*.

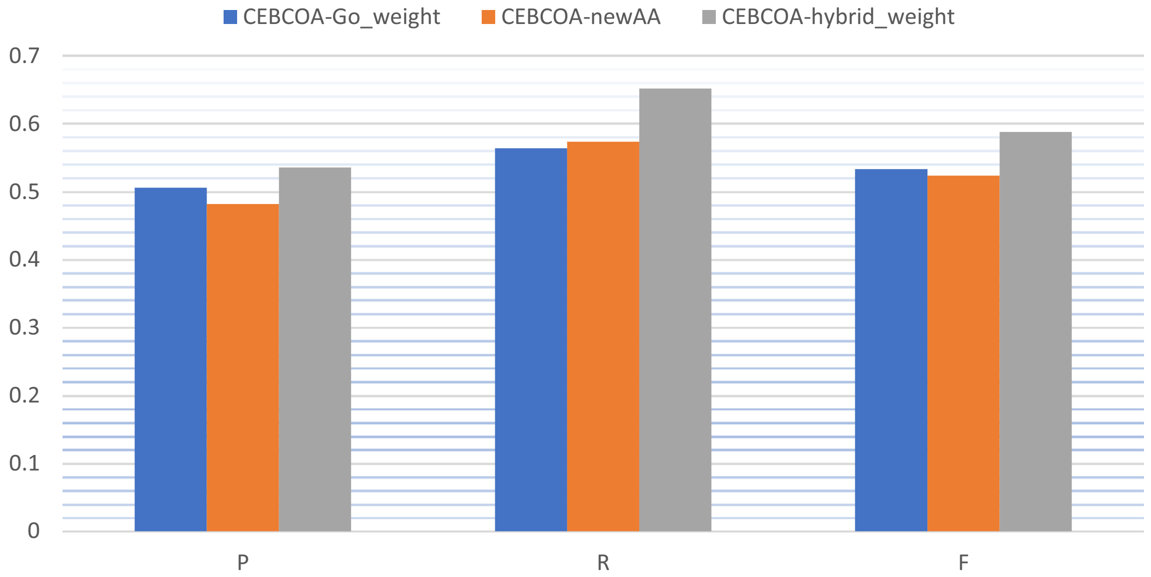
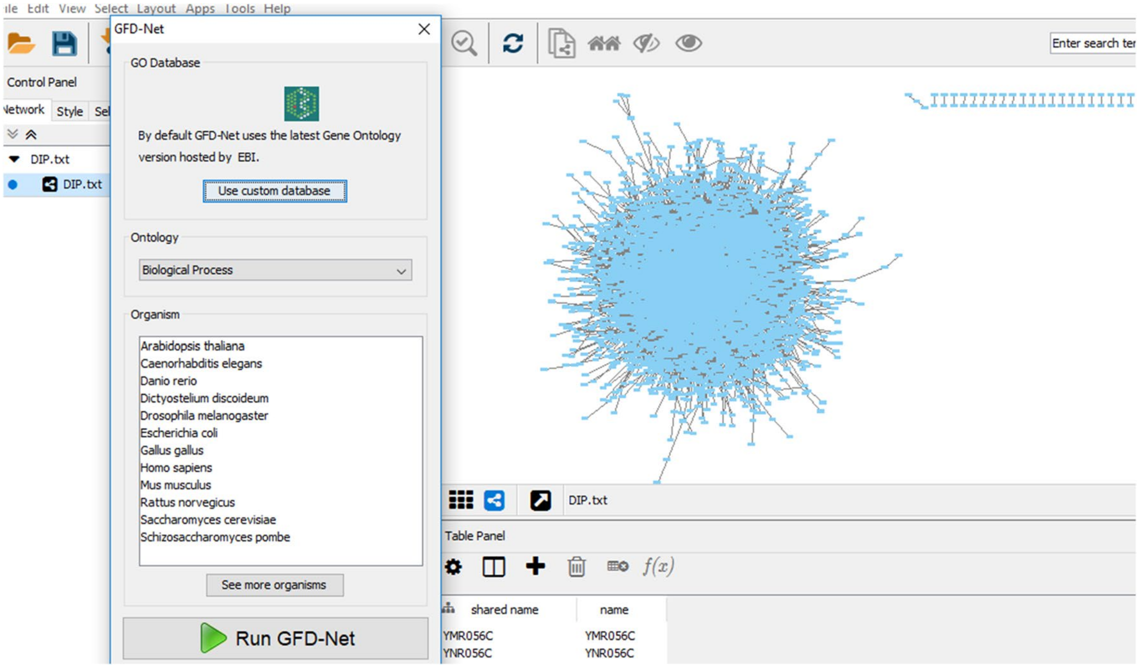


**Fig. 3** Accuracy (Precision, Recall, F-measure) of different weighting methods in CEBCOA by applying to network Collins based on benchmark CYC2008 dataset



GO Database

By default GFD-Net uses the latest Gene Ontology version hosted by EBI.

Use custom database

Ontology

Biological Process

Organism

Arabidopsis thaliana
Caenorhabditis elegans
Danio rerio
Dictyostelium discoideum
Drosophila melanogaster
Escherichia coli
Gallus gallus
Homo sapiens
Mus musculus
Rattus norvegicus
Saccharomyces cerevisiae
Schizosaccharomyces pombe

See more organisms

Run GFD-Net

| shared name | name |
|---|---|
| YMR056C | YMR056C |
| YNR056C | YNR056C |

**Fig. 4** A part of the running GFD-Net plug-in to weigh the networks for selecting ‘Saccharomyces cerevisiae S288c’ organism

### 2.2.3 Definition 3: Weighted Local Average Connectivity-Based Centrality

The local average connectivity-based centrality *LACW (u)* of node *u* is defined as (Relation 14) [29].

$$\mathrm{LACW}(u)=\sum_{s\in N_u}\sum_{t\in N_s\cap N_u} w(s,t)/\left|N_u\right| \tag{14}$$

where $N_u$ denotes the set of its neighbors and $\left|N_u\right|$ does the number of its neighbors. *W(s, t)* is the weight of the edge connecting node *s* and node *t*. Notation *s* is one of the neighbors of node *u*, and *t* is the neighbor of *s* in $N_u$.

### 2.2.4 Definition 4: Semantic Similarity. The Semantic Similarity is the Amount of Information that Two Nodes Share

To indicate the importance of each vertex, the graph topology properties are used. One of the most well-known measurements for detecting essential proteins (important nodes) is the vertex degree. The centrality of degree is used for local measures, but this criterion shows poor network connectivity on the network, while, in most articles, it is reported that the centrality measure alone cannot properly predict all the essential proteins, in particular, the low-connectivity essential protein. Thus, using a combination of the biological property and centrality measures, we increase the precision of the detection of the essential proteins using Relation 15 [14].

$$\begin{aligned}\mathrm{RANKESS}(x)&=a*\mathrm{DWNN}(x)+b*\mathrm{MCLUS}(x)\\&\quad+c*\mathrm{ESS}(x)+d*\mathrm{LACW}(x)\end{aligned} \tag{15}$$

where *a, b, c,* and *d* represent coefficients to show the importance of each component in the calculation for the identification of essential proteins.

To obtain the appropriate values of the parameters a, b, c and d, tests were conducted by a logistic regression classifier. The *DWNN* functions show the impact of the effective neighboring vertex, and the *MCLUS* function measures the biological property of the accumulation of essential proteins in a module, as well as better identifies low-connectivity essential proteins. Meanwhile, the *ESS* function measures the biological property of the binding of essential proteins to other important proteins and better identifies the low-connectivity essential protein. *LACW* is also a centrality measure and, when used with a combination of local and global centrality measures, better detects the essential proteins. When the two vertices have the same power and number of interactions, but different topology properties, the number, and power of interactions cannot provide a good comparison in the identification of essential proteins. For this reason, we investigated the neighborhood at higher levels to distinguish vertex topological differences (Relations 16 and 17).

$$\mathrm{DW(x)} = \sum_{\mathrm{y\in adj(x)}} \mathrm{CWD(y)} \tag{16}$$

$$\mathrm{DWNN(y)} = \sum_{\mathrm{x\in adj(y)}} \mathrm{W_{y,x}} \times \mathrm{DW(x)} \tag{17}$$

The *adj(x)* represents the set of close neighbors near *x*, and *CWD(y)* represents the total centrality of weight degree in the betweenness centrality of the y vertex and its neighbors at two subsequent levels. The coefficient $W_{y,x}$ also shows the effect of interaction power on the importance of the essential protein, which is set by the weight of the edges. There is a number of functional modules in the PPI network, which plays a key role in the biological function, and essential proteins tend to be in these functional modules. Using the *MCODE* plug-in in the Cytoscape application, the clustering operation was performed on the data set, after which the rank of each protein was calculated relative to the available complexes. If $CP=cp_1, cp_2, \dots, cp_n$, this means that there is a set of protein complexes and protein x in the complexes. In turn, the total weight degree of that protein is obtained relative to the proteins within the complexes, which is the measure of protein *x* (*MCLUS(x)*) (Relation 18) in this section.

$$MCLUS(x) = \sum W_{x,y}\, |x, y \in \left\{cp_1, cp_2, \dots, cp_n\right\}\, and\, y \in neighbor(x) \tag{18}$$

The other biological properties of the essential proteins can involve their relationship with other important proteins. To measure this biological property, we present it as *ESS(x)* (Relation 19).

$$ESS(x) = \sum W_{x,y}\, |y \in neighbor(x)\, and\, y \in \{Importantprotein\} \tag{19}$$

The important protein is calculated using Algorithm 2.

Using a logistic regression method, we tested the combinations of centrality measures with three neighborhood criteria in the second and third levels *(DWNN)*, while *MCLUS*, *ESS,* and *LACW* were selected with a good level of significance compared to other centrality measures. Then, with the ranking of proteins, their essentiality as judged. Inevitably, before the ranking, all values of the measures were normalized by the maximal-minimum normalization method. In applying this method, all values are converted to values between 0.0 and 1.0. If max ($X_{ij}$) and min ($X_{ij}$) are respectively the maximum and minimum value of the property of *j* and $X_{ij}$ is the measure of protein *i* in the centrality and the property *j*, the normal value of the measures is obtained according to Relation 20.

$$\mathrm{NM}\left(\mathrm{X_{ij}}\right) = \mathrm{X_{ij}} - \min\left(\mathrm{X_{ij}}\right) / \max\left(\mathrm{X_{ij}}\right) - \min\left(\mathrm{X_{ij}}\right) \tag{20}$$

## 2.3 Step 3

In this step, we obtain a probability value of essentiality of each protein (Fig. 5, column 2). Figure 5 shows the result of Relation 15 which has been obtained by the software we presented in [14].

The first and second columns in Fig. 5 denote the systematic protein name and the probability value. The probability value is used to select the essential protein. Next, we arrange proteins based on importance as descending. Proteins with a *RANKESS* value greater than $\alpha$ are considered to be primary centers (*seed*) (Algorithm (3)). By setting $\alpha$, we can control the number of predicted complexes. The higher values for $\alpha$ result in fewer protein complexes. Tests show that the best value for $\alpha$ should be greater than the average for the upper 50% of the *RANKESS* values. Further, to discover the complex core, the weighted network *G=(V, E, W)* is considered as an input, and the set *C* is formed as Relation (21).

```
Algorithm 2. Important protein algorithm
Input: G (V, E, W)
Output: Set of Important proteins (SIP)
1.  Initialize the dataset of the important protein, SIP={∅}.
2.  Calculate the weighted degree of each protein in the dataset.
3.  Find the protein with the largest weighted degree (p-max) in the dataset and add it to the SIP.
4.  Remove p-max and its neighboring proteins from the dataset.
5.  Return to step three and repeat until the dataset is empty.
6.  Output the SIP.
```

| Protein Name | Probability value |
|---|---|
| YML094W | 0.633646425 |
| YLR200W | 0.671355636 |
| YGR078C | 0.76116095 |
| YNL153C | 0.541727692 |
| YEL003W | 0.548460752 |
| YER016W | 0.525051973 |
| ... | ... |

**Fig. 5** Essential probability of each protein

$$C = \{(Gc_v, S_v)\ |v\ \in \text{Seed and } S_v = \emptyset\} \tag{21}$$

where $Gc_v$ is the core graph of vertex $v$ and is obtained using Relations 22–24.

```
Algorithm 3. Seed generation algorithm
Input: G (V, E, W), DCW, BCW, LACW
Output: the set of candidate proteins for core centers
    Seeds ← an empty set.
    For each protein∈ V(G) do
    Compute its DWNN, MCLUS and Ess function.
    Compute its RANKESS function as Relation 15
    End For
    For each protein P ∈ V(G) do
    If RANKESS(P)>∝ then Seeds← Seeds ∪ P
    End For
```

$$V^{Gc_v} = V_w(G_v) \tag{22}$$

$$E^{Gc_v} = \{(u, u') \in E^{Gc_v}\ |\ u, u' \in V^{Gc_v}\} \tag{23}$$

$$\forall(u, u') \in E^{Gc_v}, W^{Gc_v}\ (u, u') = W^{G_v}\ (u,u') \tag{24}$$

where $V_w\ (G_v)$ the set of the core vertices in a weighted graph (Relation 25), $G_v$ is the neighborhood graph of the vertex $v \in seed$, and $S_v$ is a set of vertices that are initially null.

$$V_w(G_v) = \left\{u \in V^{Gc_v} | W(u)\ >\ W_m(G)\right\} \tag{25}$$

where $W_m(G)$ is the weighted average of the network edges.

## 2.4 Step 4

Considering the seed obtained in *step 3*, we use Algorithm 3 to predict preliminary cores where the initial core set is formed in the input network.

```
Algorithm 3: predict preliminary cores
Input: set C, threshold $t$
Output: the set of preliminary cores P
    P = ∅
    ∀ $(Gc_v, S_v) \in C$
    if ( $D_w(Gc_v) \geq t$)
    ∀ vertex of $S_v$ is added to $Gc_v$ and $Gc_v$ is appended to C
    else
    sub-graph $Gc_v$ is decomposed by removing $V_w(G_C)$ vertices from the set $V^{Gc_v}$. Each connected component
      of this decomposition and $S_v$ are considered as a pair and is added to C.
```

where $D_w$ is the weighted density (Relation 26) [30] and $t$ is a threshold for controlling the density of the discovered cores.

$$D_w(G) = \frac{2 * \sum_{(u,v)\in E} W(u,v)}{|V|(|V| - 1)} \tag{26}$$

## 2.5 Step 5

After the discovery of the primary cores and the removal of the repetitive cores, depending on their overlapping, we obtain the set of final cores using Algorithm (4).

The measure of the overlapping of the two graphs is calculated from the neighborhood affinity equation (Relation 27) [31].

$$NA(G,G') = |V \cap V'|^2/|V| * |V'| \tag{27}$$

where $t$ is the threshold for controlling the amount of overlap between predicted cores.

## 2.6 Step 6

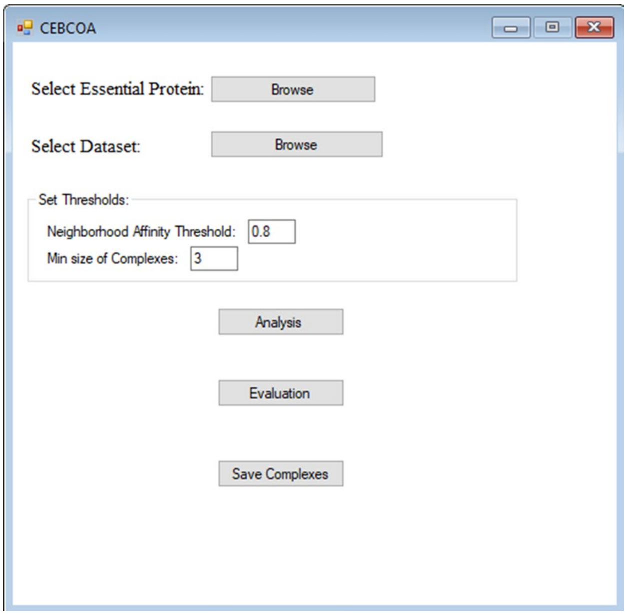
In this step, the neighboring nodes of each core ($V_N(\mathrm{Gc_v})$) (Relation 28) are assumed as attachment nodes. Each of the neighbors is added to the core. If the new density ($D_{NEW}$) (Relation 29) is greater than the density of the core ($D_w(\mathrm{Gc_v})$), then the neighboring node is considered as the attachment node.

```
Algorithm 4. Redundancy Filtering
Input: the set of Preliminary cores P and threshold $t$
Output: F: The set of final cores
F=∅.
For each core graph $G_C$∈ P.
$G_{c'}$=argmax $NA$ ($G_{c'}$,$G_c$) | core graph $G_{c'}$ is the most similarity to core graph $G_c$.
If $NA(G_{c'}, G_c)<t$
    Insert $G_c$ into F.
else
    if den( $G_c$) * $|V^{G_c}|$ ≥ den ($G_{c'}$) *$|V^{G_{c'}}|$
        Replace $G_{c'}$ with $G_c$.
```

$$V_N(\mathrm{Gc_v}) = \{\mathrm{u} \mid (\mathrm{u},u')\in E\ ,u \in V(G) and\ \ u' \in V_{\mathrm{Gc_v}}\} \tag{28}$$

$$D_{NEW} = \left(D_w\left(\mathrm{Gc_v} \cup V_{N_j}(\mathrm{Gc_v})\right)\right) \tag{29}$$

Ultimately, complexes are formed after finding attachment proteins and combining them with cores. We then remove complexes containing duplicate proteins. In this way, the proteins remaining in them are again assigned to the complexes according to the algorithms as mentioned earlier. Our proposed method (CEBCOA including six steps) was implemented through the C# language where Fig. 6 shows a part of running this method. First, we click the 'Browse' buttons to select the dataset and essential proteins as a seed of complex core. Next, we set the thresholds and then click the 'Analysis' for construction of complexes.



Neighborhood Affinity Threshold: 0.8
Min size of Complexes: 3

**Fig. 6** A part of running of the CEBCOA method. "Select dataset and essential proteins" for seed of complex core and "Analysis" for construction of complexes

# 3 Results and Discussion

In this section, we first discuss the parameter of the proposed method. Then, based on the evaluation criteria, comparisons between the proposed method and the traditional methods are made and the results analyzed. Finally, comparisons are made in terms of the statistical significance of predicted complexes in different methods. We applied *CEBCOA* and other methods to networks *DIP* [18], *COLLINS* [19], *Krogan* [20] and *Human PPI* [32].

## 3.1 Evaluation Metrics

Three evaluation metrics were explained in Sect. 2. In this section, three other evaluation metrics, which we used to performance evaluation of methods, are explained.

**Table 1** Properties of the used PPI networks

| Data Set | #proteins | #interaction | Benchmark |
|---|---|---|---|
| DIP | 4930 | 17,201 | CYC2008,MIPS |
| Krogan | 2675 | 7084 | CYC2008,MIPS |
| Collins | 1622 | 9074 | CYC2008,MIPS |
| Human PPI | 5664 | 37,437 | CORUM |

### 3.1.1 Coverage Rate

This metric is used to evaluate the number of proteins of a real complex that can be covered by a predicted complex. It is defined as Relation 30 [33] where $n$ and $m$ are the number of real complexes and the number of predicted complexes, respectively, $T_{ij}(i \le n, j \le m)$ is the number of common proteins between the $i^{th}$ real complex and $j^{th}$ predicted complex, and $N_i$ is the number of proteins in the $i^{th}$ real complex.

$$GR = \frac{\sum_i^m max\{T_{ij}\}_j}{\sum_i^n N_i} \tag{30}$$

### 3.1.2 $S_n$, PPV, Acc

Sensitivity ($S_n$) and Positive Predictive Value (PPV) are two other metrics, which are used to evaluate the accuracy of predictions (Relation 31, [33] where $n$ and $m$ are the number of benchmark complexes and the number of predicted complexes, respectively.

$$S_n = \frac{\sum_{i=1}^n \max\{T_{ij}\}_j}{\sum_{i=1}^n N_i}, \ PPV = \frac{\sum_{j=1}^m \max\{T_{ij}\}_i}{\sum_{j=1}^m T_j} \tag{31}$$

where $N_i$ is the number of proteins of the $i^{th}$ benchmark complex and $T_j = \sum T_{ij}$ is the number of common proteins

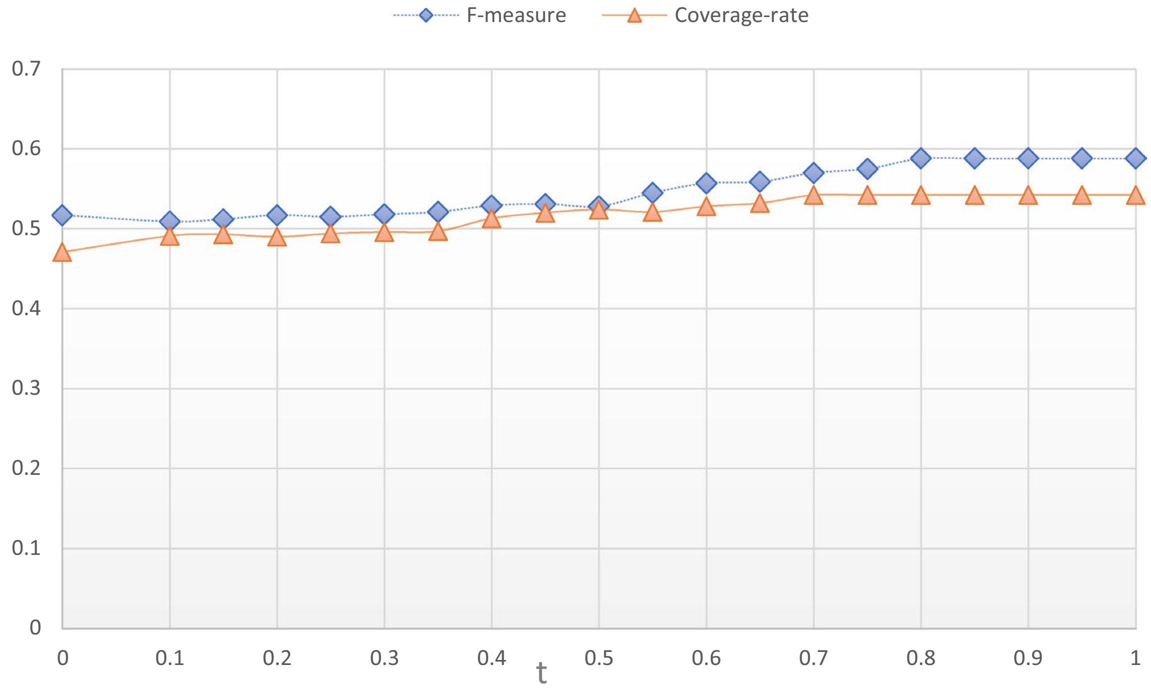


**Fig. 7** The F-measure and coverage rate values obtained by applying *CEBCOA* to network Krogan when threshold value (t) of neighborhood affinity varies

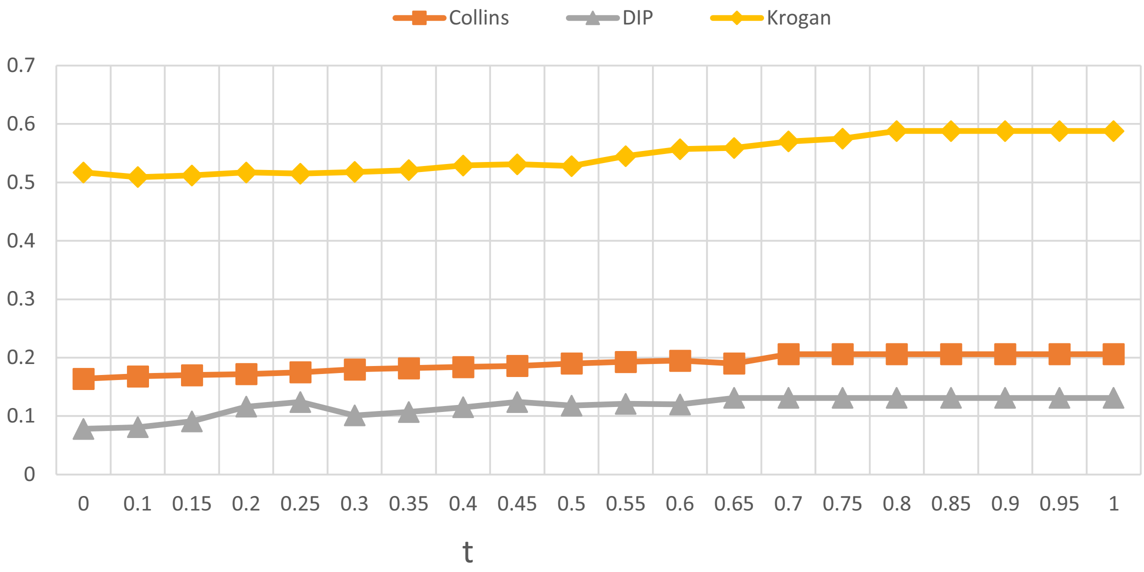


**Fig. 8** Increase of F-measure values by increasing neighbourhood thresholds in applying *CEBCOA* to three networks

between $i^{th}$ benchmark complex and the predicted $j^{th}$ complex. Usually a high value of $S_n$ indicates that there is a good prediction of proteins in the real complex and a high PPV does the predicted complex has an accurate result with a high probability. Based on $S_n$ and PPV, the geometric accuracy (Acc) (Geometric mean, [33] is obtained (Relation 32).

$$\mathrm{Acc} = \sqrt{\mathrm{S_n} \times PPV} \tag{32}$$

### 3.1.3 P-Value

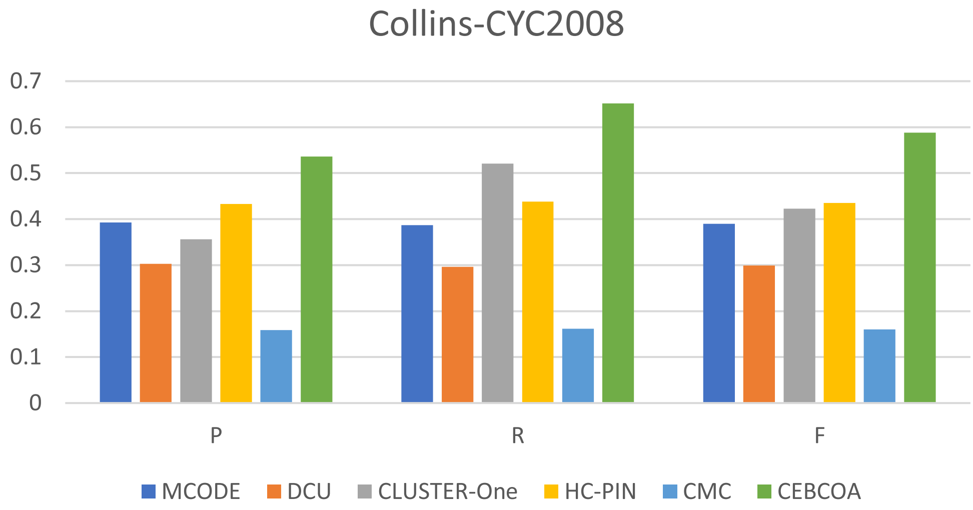
P-value is a metric that is used to evaluate the biological significance of the predicted complexes. If the $k$ protein in the predicted complex $P = (V^p, E^p)$ is assumed to have the same function and to be in a functional group with $m$ members, then the *p-value* is obtained for the predicted complex using Relation 33 [34].

$$p - value = 1 - \sum_{i=0}^{k-1} \binom{m}{i} \binom{N-m}{|V^p| - 1} \Big/ \binom{N}{|V^p|} \tag{33}$$

where $N$ denotes the total number of proteins in the PPI network.

A low P-value indicates the low probability of randomly gathered proteins in the predicted complex [25]. If the *P-value* of the predicted complexes is less than 0.01, then they will be significant [25]. Using the *GoTermFinder* tool [35], we can obtain the *P-value* from the gene ontology structure. When comparing the proposed method with

**Fig. 9** Comparison of CEBCOA with five methods for network Collins in terms of (P)recision, (R)ecall and (F)-measure of based on benchmark CYC2008 as a gold standard



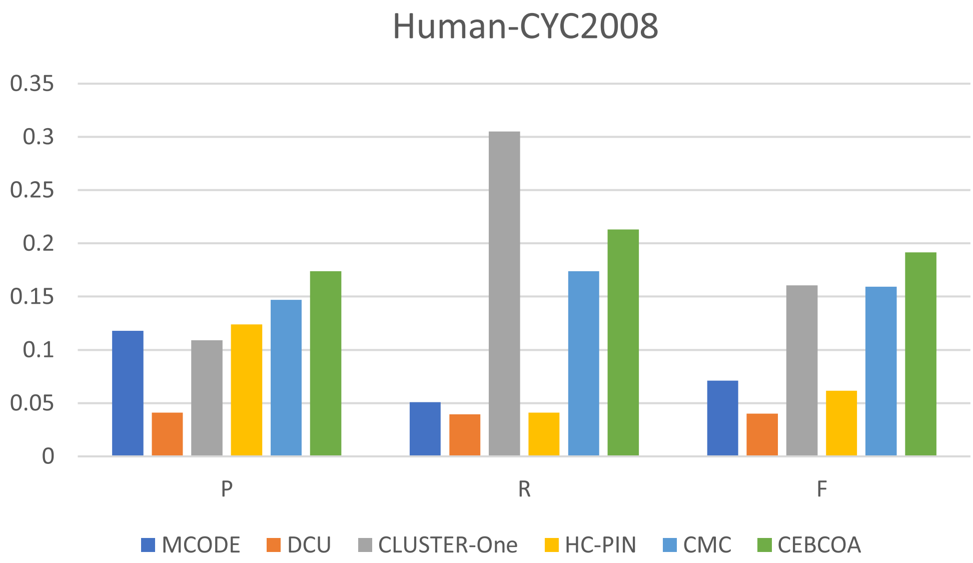


**Fig. 10** Comparison of CEBCOA with five methods for network Human in terms of (P)recision, (R)ecall and (F)-measure based on benchmark CYC2008 as a gold standard dataset

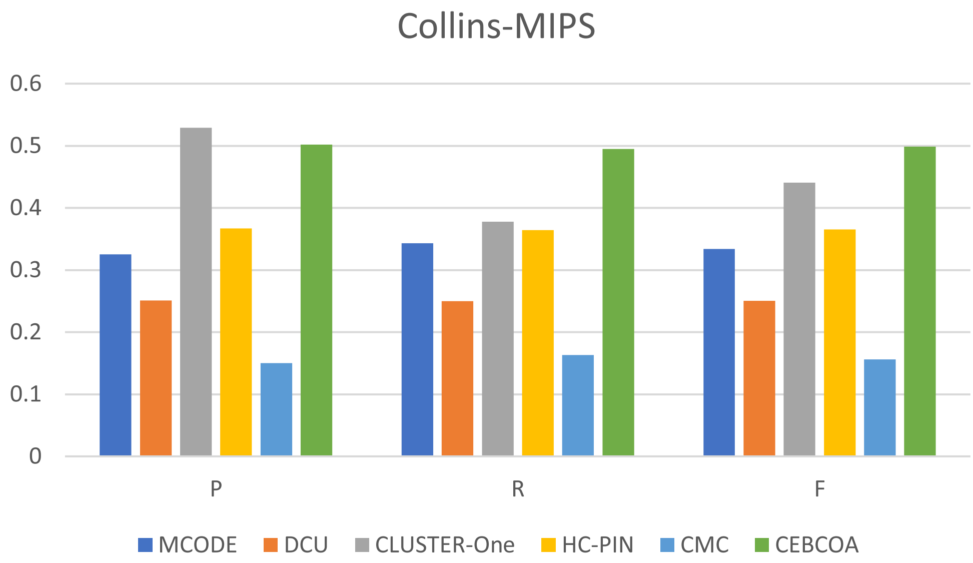


**Fig. 11** Comparison of CEBCOA with five methods for network Collins in terms of (P)recision, (R)ecall and (F)-measure based on benchmark MIPS as a gold standard dataset

other methods, it can be seen that a method that predicts the number of significant complexes is better.

## 3.2 Practical Considerations

### 3.2.1 Datasets

To evaluate the CEBCOA method, three yeast protein interaction networks, i.e., *DIP* [18], *COLLINS* [19] and *Krogan* [20] and one human protein interaction network, i.e., *Human PPI* datasets [32] were used. Table 1 shows the number of proteins and interactions in each of these networks.

Nowadays, researchers use Homo sapiens dataset to evaluate their methods. Human PPI data has many challenges, such as high noisy data, a large number of small complexes, and overlapping functions of proteins in several complexes, which may create naming problems in mapping the various *UniProt IDs* into identical proteins [36]. In this paper, we used the *Human PPI* dataset [32] with 37,437 interactions and the *CORUM* dataset [37] as the golden standard of 1843 human complexes.

### 3.2.2 The Effect of Parameters

The threshold value of the neighborhood affinity $t$ was applied to the redundancy-filtering algorithm in order to control the amount of overlap between the discovered cores. The *CEBCOA* method was executed with different values of $t$ in the networks. For example, Fig. 7 shows the value of F-measure and Coverage rate of the proposed method for

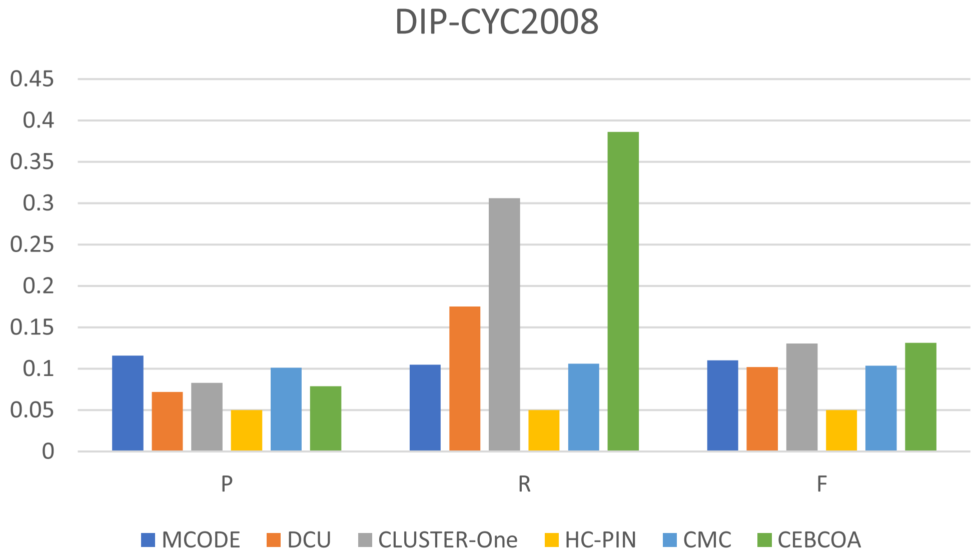


**Fig. 12** Comparison of CEBCOA with five methods for network DIP in terms of (P)recision, (R)ecall and (F)-measure based on benchmark CYC2008 as a gold standard dataset

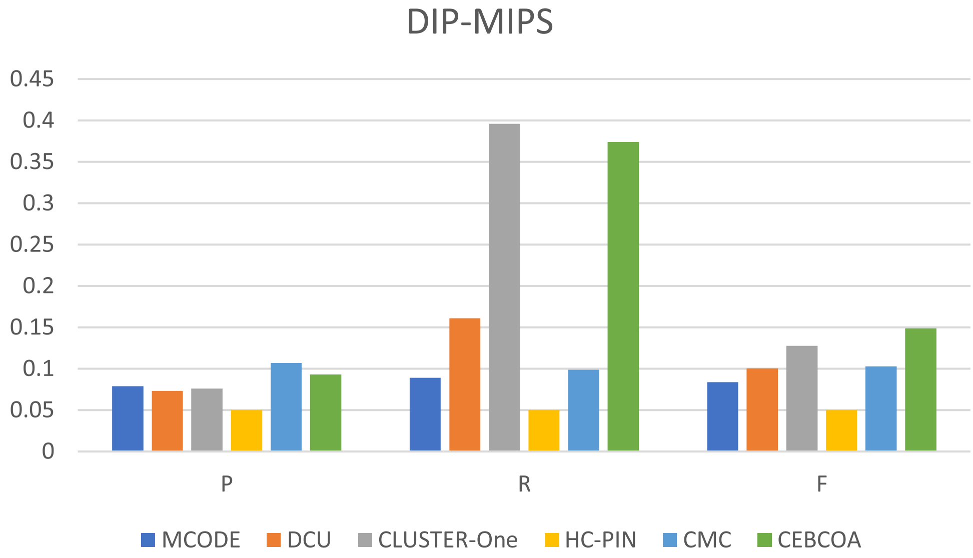


**Fig. 13** Comparison of CEBCOA with five methods for network DIP in terms of (P)recision, (R)ecall and (F)-measure based on benchmark MIPS as a gold standard dataset

different values of t on the Krogan network. By increasing the value of t, overlap among the cores of protein complexes increases, thus the number of predicted complexes, more real complexes were covered by predicted complexes.

As Fig. 7 shows, in our experiment with increasing the value of $t$ in case of $t = 0.8$, there is a good balance between F-measure and coverage rate. Moreover, As Fig. 8 shows, by increasing the value of $t$ that of *F_measure* also increases. When $t > 0.65$, *F_measure* in DIP data set remains stable. In dataset *Collins*, when $t > 0.7$, *F_measure* remains stable, and in dataset *Krogan* when $t > 0.8$, *F_measure* has a tendency to remain stable. Thus, when the neighbourhood threshold is adjusted to the 0.8, all of the datasets have the highest *F-measure* and remain stable.

The threshold of density value $d$ was used to discover the primary cores and control the weighted density of the predicted cores. To obtain a value for $d$, we first obtain the average weight of the network edges ($W_{ave}$), then according to the choice of *Seed* and attachment vertices based on multiple classifier, we can calculate the density threshold for each core graph using Relation 34.

$$D_{ave}\left(Gc_v\right) = 2 * \left|E^{Gc_v}\right| * W_{ave} \Big/ \left|V^{Gc_v}\right| \left(\left|V^{Gc_v}\right| - 1\right) \quad (34)$$

### 3.2.3 Method Comparison

We compared the CEBCOA method to methods *MCODE* [7], *DCU* [15], *ClusterOne* [16], *HC-PIN* [17] and *CMC*

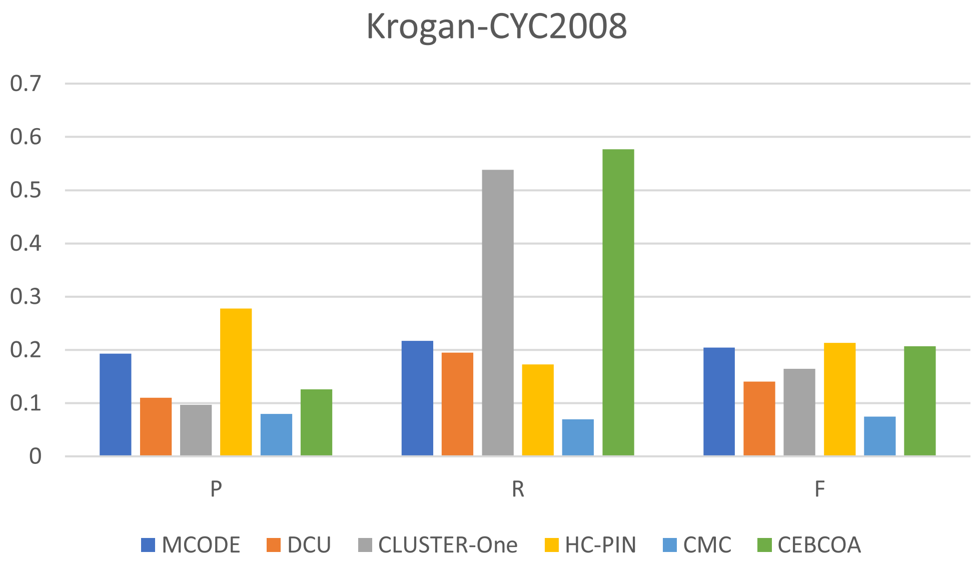


**Fig. 14** Comparison of CEBCOA with five methods for network Krogan in terms of (P)recision, (R)ecall and (F)-measure based on benchmark CYC2008 as a gold standard dataset

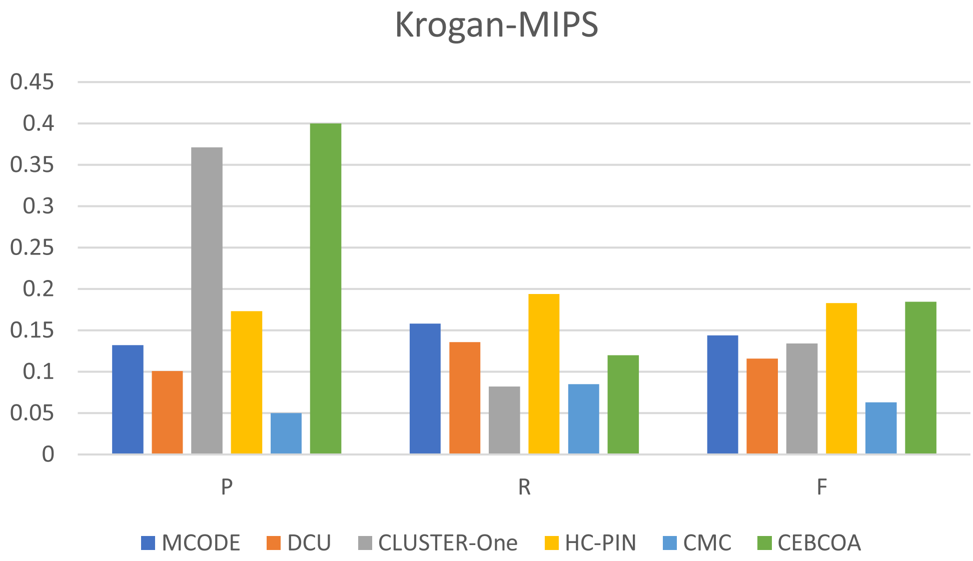


**Fig. 15** Comparison of CEBCOA with five methods for network Krogan in terms of (P)recision, (R)ecall and (F)-measure based on benchmark MIPS as a gold standard dataset

[5]. MCODE, *ClusterOne*, HC-PIN and CMC have a plug-in with the same name in the Cytoscape software. Figures 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 and 21 show the results of these methods by applying to networks *Collins, DIP* and, *Krogan*.

**3.2.3.1 Comparison Based on Precision, Recall, F-Measure** By applying methods to the *Collins* network and based on CYC2008 as gold standard, Fig. 9 shows *CEBCOA* outperforms other methods in terms of *precision*, *recall,* and *F-measure* where *precision*, *recall,* and *F-measure* values of CEBCOA are 0.536, 0.652, and 0.588, respectively, while the *F-measure* value *for MCODE, DCU, ClusterOne, HC-PIN and CMC* methods is 0.389, 0.299, 0.422, 0.435 and 0.160, respectively.

For the *Human* network, *CEBCOA* gained 0.174, 0.213 and 0.191 for *Precision*, *Recall*, and *F-measure*, respectively (Fig. 10) while the *F-measure* of the *MCODE, DCU, ClusterOne, HC-PIN and CMC* methods are 0.071, 0.04, 0.161, 0.061 and 0.159, respectively. According to these values, the *CEBCOA* method performed better than the five other methods.

Based on the MIPS data as gold standard (Fig. 11), *CEBCOA* has the best performance in term of F-measure value with 0.495, while this value for MCODE, DCU, ClusterOne, HC-PIN and CMC methods is 0.333, 0.250, 0.440, 0.365 and 0.156, respectively.

As shown in Fig. 12, results of applying *CEBCOA* to the *DIP* network are very good based on the CYC2008 gold standard as a benchmark where the *CEBCOA recall*

**Fig. 16** Comparison of CEBCOA with five methods for network Collins in terms of Sn, PPV, Acc, and MMR based on benchmark CYC2008 as a gold standard dataset

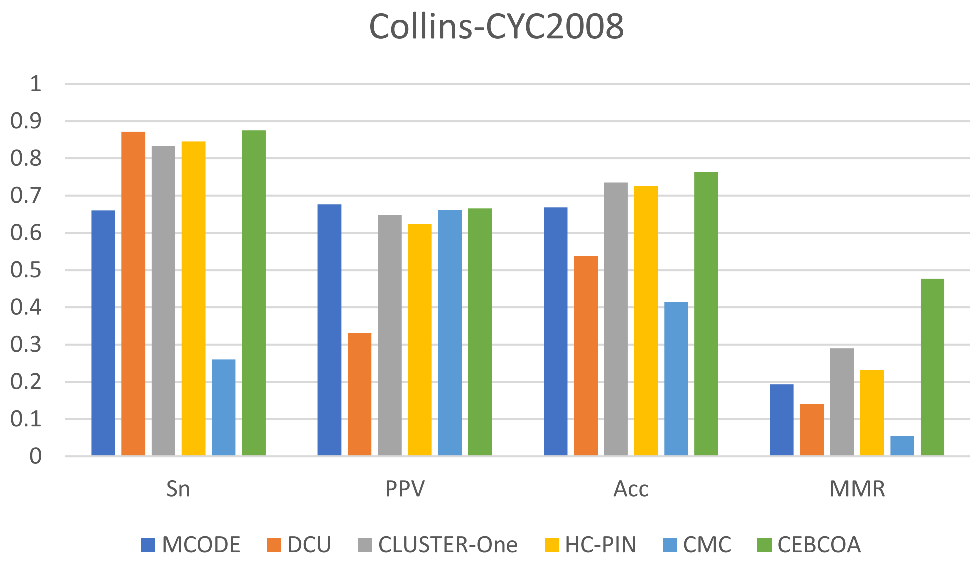


**Fig. 17** Comparison of CEBCOA with five methods for network Collins in terms of Sn, PPV, Acc, and MMR based on benchmark MIPS as a gold standard dataset

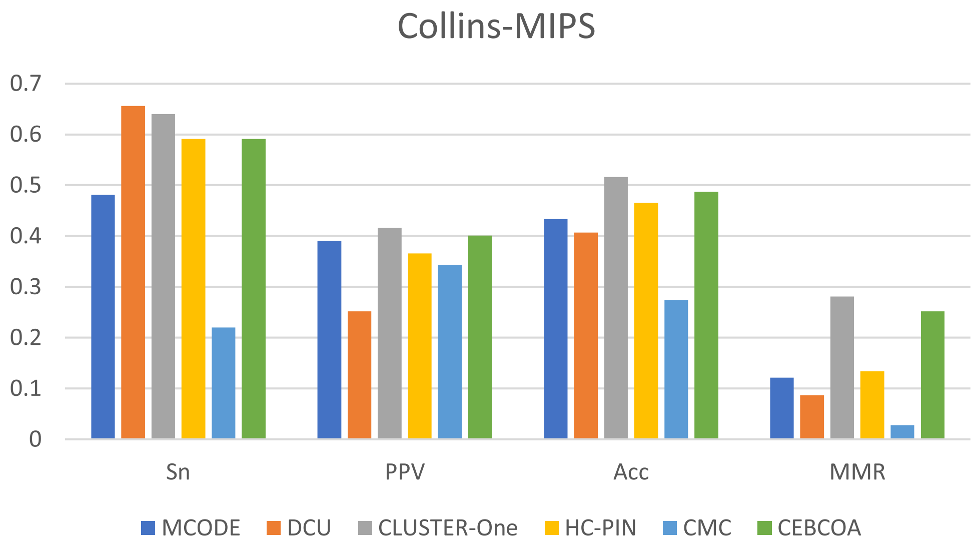


**Fig. 18** Comparison of CEBCOA with five methods for network DIP in terms of Sn, PPV, Acc, and MMR based on benchmark CYC2008 as a gold standard dataset

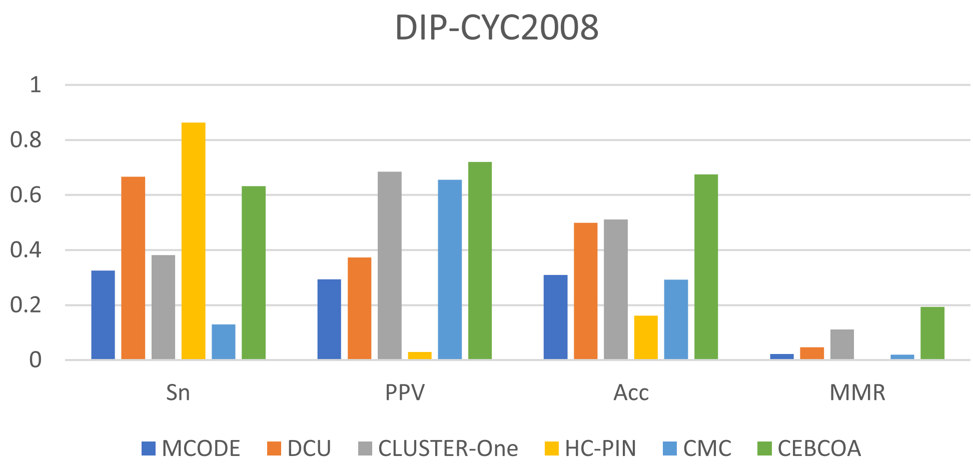


**Fig. 19** Comparison of CEBCOA with five methods for network DIP in terms of Sn, PPV, Acc, and MMR based on benchmark MIPS as a gold standard dataset

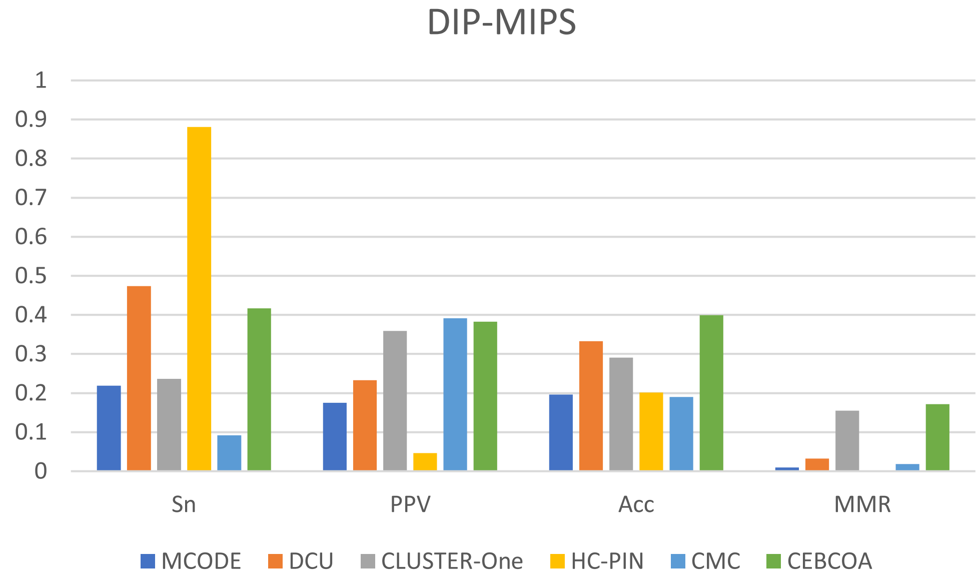
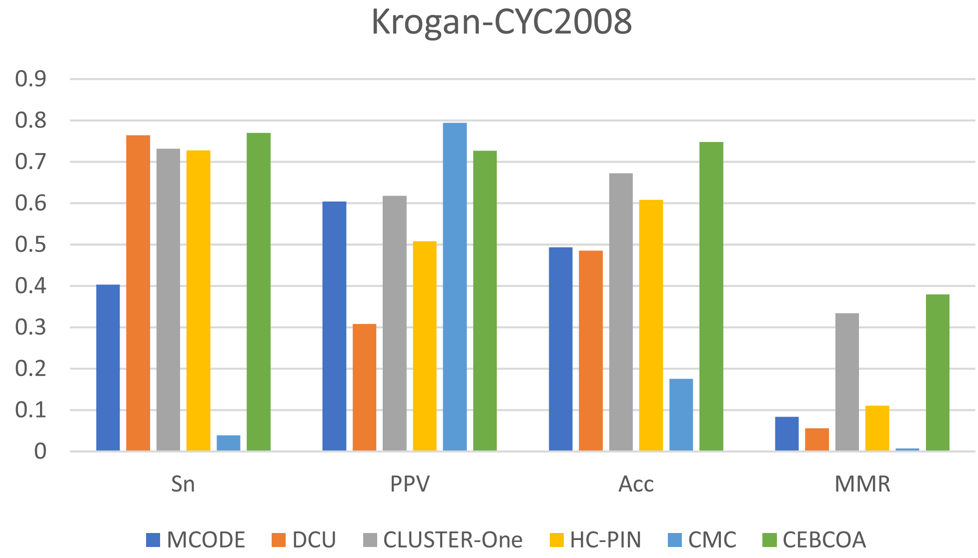


**Fig. 20** Comparison of CEBCOA with five methods for network Krogan in terms of Sn, PPV, Acc, and MMR based on benchmark CYC2008 as a gold standard dataset



and *F-measure* values outperform those of other methods. The F-measure value of *CEBCOA* is 0.131 while that of MCODE, DCU, ClusterOne, HC-PIN and CMC is 0.083, 0.100, 0.127, 0.05 and 0.103, respectively. When we used the MIPS data set as benchmark gold standard (Fig. 13), the F-measure value of *CEBCOA* was 0.148 while that of MCODE, DCU, ClusterOne, HC-PIN and CMC was 0.110, 0.102, 0.130, 0.05 and 0.102, respectively. We applied methods to the Krogan network and used the CYC2008 as a benchmark dataset to performance evaluation (Fig. 14). The CEBCOA method has the highest: (1) precision value after the HC-PIN and MCODE methods, (2) Recall value and (3) highest F-measure harmonic mean value after the HC-PIN method. The F-measure value of CEBCOA is 0.007 less than that of HC-PIN.

By applying methods to the *Krogan* network and based on the MIPS benchmark complexes, Fig. 15 shows that the *precision* and *F-measure* values of the *CEBCOA* method are higher than those of other methods. These values for *CEBCOA* are respectively 0.4 and 0.184, while the *precision* and *F-measure* values for the *MCODE, DCU, ClusterOne, HC-PIN and CMC* methods are (0.132, 0.143), (0.101, 0.115), (0.371, 0.134), (0.173, 0.182) and (0.05, 0.06) respectively. However, the *recall* value for *CEBCOA*, *MCODE, DCU, ClusterOne, HC-PIN and CMC* is 0.12, 0.158, 0.136, 0.082, 0.194 and 0.085, respectively.

Since the F-measure value is the harmonic mean of the two values Precision and Recall, it shows a better criterion in the comparisons. As Figs. 9, 10, 11, 12, 13 and 15 show,

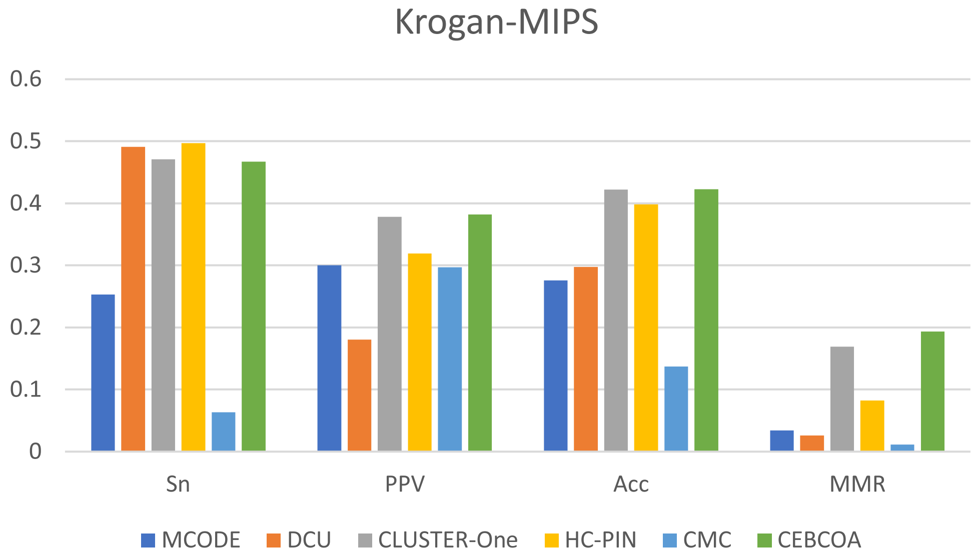
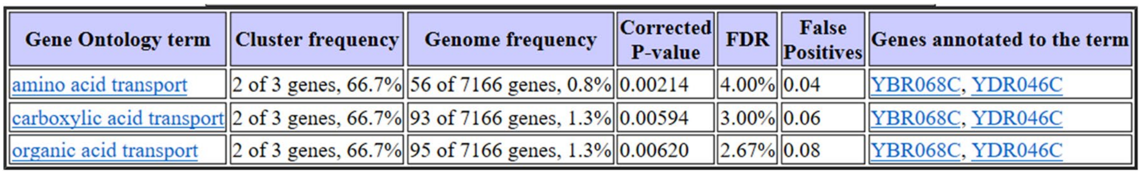


**Fig. 21** Comparison of CEBCOA with five methods for network Krogan in terms of Sn, PPV, Acc, and MMR based on benchmark MIPS as a gold standard dataset



| Gene Ontology term | Cluster frequency | Genome frequency | Corrected P-value | FDR | False Positives | Genes annotated to the term |
|---|---|---|---|---|---|---|
| amino acid transport | 2 of 3 genes, 66.7% | 56 of 7166 genes, 0.8% | 0.00214 | 4.00% | 0.04 | YBR068C, YDR046C |
| carboxylic acid transport | 2 of 3 genes, 66.7% | 93 of 7166 genes, 1.3% | 0.00594 | 3.00% | 0.06 | YBR068C, YDR046C |
| organic acid transport | 2 of 3 genes, 66.7% | 95 of 7166 genes, 1.3% | 0.00620 | 2.67% | 0.08 | YBR068C, YDR046C |

**Fig. 22** The number of anticipated proteins by CEBCOA using the GOTermFinder tool and their p-value for three samples of Gene Ontology term

the F-measure value of the proposed method is higher than that of all methods.

#### 3.2.3.2 Comparison based on Sensitivity, Positive Prediction Value, Geometry Accuracy

Moreover, we used mertics $S_n$ (Sensitivity), PPV (Positive Prediction Value) and Acc (Geometry Accuracy) to evaluate our proposed method (*CEBCOA*) in detection of protein complexes. Based on benchmarks the MIPS and CYC2008 dataset, values of these criteria for methods were obtained by applying to networks: (1) Collins in Figs. 16 and 17, (2) DIP in Figs. 18 and 19, and (3) Krogan in Figs. 20 and 21.

As Fig. 17 shows, methods DCU, ClusterOne, and CEBCOA have the highest $S_n$ values, respectively. However, we know that if a method produces a large complex among the predicted clusters, its $S_n$ value is not fully reliable. The PPV criterion is a solution to this defect so that the significant difference between $S_n$ and PPV for the DCU method is a reason for this claim. Therefore, Acc, which is a geometric mean of values $S_n$ and PPV is used to compare the performance of the methods. Figure 16 shows that the CEBCOA method has a higher Acc value than the other methods. According to the results shown in Fig. 17, the ClusterOne and CEBCOA methods are the first and the second methods in terms of Acc, respectively. Figures 18 and 19 show that the CEBCOA method has a higher Acc value than the other methods.

#### 3.2.3.3 Comparison Based on MMR

The next metric we used to compare the efficiency of the methods is MMR (Maximum Matching Ratio). It shows how much a method can distinguish real complexes based on quantity and quality. As Fig. 17 shows, CEBCOA has the best MMR value after ClusterOne. According to Figs. 18, 19, 20 and 21, CEBCOA and ClusterOne are considered to be the first and second methods in term of the MMR benchmarks. Therefore, these methods have higher accuracy for identifying predicted clusters for real complexes. Finally, Figs. 16, 17, 18, 19, 20 and 21 show applying the six methods to different networks where CEBCOA produces more efficient results in terms of metrics PPV, Acc, and MMR; however, among the methods, HC-PIN benefits the highest sensitivity (Sn) for some networks. Furthermore, as Figs. 16 and 17 show, CEBCOA has the lower efficiency than ClusterOne for the Collins network based on benchmark MIPS.

#### 3.2.3.4 Comparison Based on Biological Significance

Since the set of reference complexes is incomplete, to verify the effectiveness of our method, we considered the biological

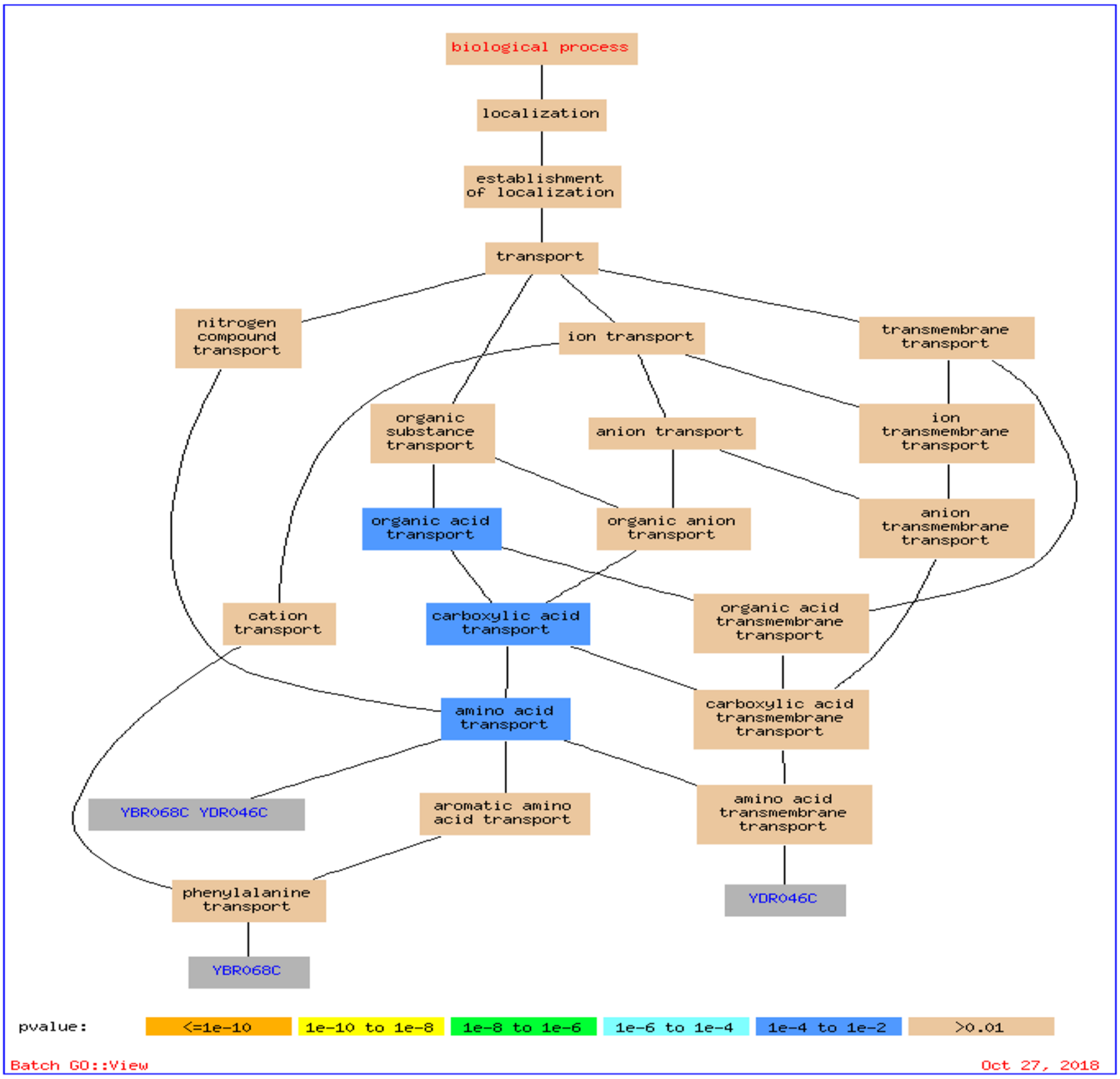


**Fig. 23** Search result in GO (gene ontology) view tree image format

significance of the predicted protein complexes. If we count just the proteins containing annotation, it will be highly misleading because the distribution of the genes found among the various margins is not uniform [34]. Therefore, we use *p-value* to calculate the statistical and biological significance of a complex of proteins. Basically, p-value shows the probability of observing the proteins of a complex which are located in a functional group by random chance. If a complex with a *p-value* is larger than a cutoff value (0.01), it is counted as a non-significant value. Since the *p-value* of a complex cannot be statistically displayed, a score clustering function for all complexes is defined as Relation 35 [34].

$$Clusteringscore = 1 - \frac{\sum_{i=1}^{n_s} min\left(p_i\right) + \left(n_I * cutoff\right)}{\left(n_s + n_I\right) * cutoff} \quad (35)$$

where $n_s$ and $n_I$ represent the number of significant and non-significant complexes, respectively and min $(p_i)$ represents the smallest significant for complex $i$. Note that for each of the predicted complexes, the smallest *p-value* is used to measure the homogeneity of its function.

A small p-value shows that the predicted complex has high biological significance. In our study, a predicted complex benefits a biological significance provided that its *p-value* is less than 0.01. The P-value from the methods is derived by the GOTermFinder tool from the gene ontology structure. First, the predicted complexes are converted

**Table 2** Biological information of 10 predicted complexes with the lowest p-value

| GOID | GOTerm | Cluster frequency | P_value | Genes annotated to the term |
|---|---|---|---|---|
| GO:0006378 | mRNA polyadenylation | 16 of 19 genes, 84.2% | 2.9e–39 | YJR093C, YLR115W, YKL059C, YLR277C, YDR195W, YNL222W, YAL043C, YDR228C, YGR156W, YOR250C, YGL044C, YMR061W, YKR002W, YPR107C, YDR301W, YNL317W |
| GO:0071051 | Polyadenylation-dependent snorna 3′-end processing | 12 of 13 genes, 92.3% | 2.2e–32 | YOL021C, YDR280W, YGR095C, YOR001W, YHR081W, YOL142W, YGR158C, YCR035C, YHR069C, YGR195W, YDL111C, YNL232W |
| GO:0042273 | Ribosomal large subunit biogenesis | 18 of 20 genes, 90.0% | 9.2e–29 | YMR290C, YGL111W, YHR088W, YBR142W, YNL002C, YPR016C, YHR052W, YOR272W, YKR081C, YDR060W, YNL110C, YGR103W, YOR206W, YPL043W, YKL014C, YMR049C, YHR066W, YNL061W |
| GO:0006511 | Ubiquitin-dependent protein catabolic process | 21 of 23 genes, 91.3% | 6.4e–28 | YGL004C, YDR363W-A, YDR394W, YER012W, YLR421C, YER021W, YEL037C, YDL147W, YOR261C, YKL145W, YPR108W, YDR427W, YDL097C, YFR010W, YDL007W, YGL048C, YFR052W, YOR259C, YOR117W, YFR004W, YHR200W |
| GO:0010499 | Proteasomal ubiquitin-independent protein catabolic process | 11 of 13 genes, 84.6% | 9.9e–28 | YER012W, YPR103W, YOL038W, YOR362C, YML092C, YFR050C, YGR253C, YER094C, YMR314W, YGL011C, YGR135W |
| GO:0000398 | mRNA splicing, via spliceosome | 16 of 17 genes, 94.1% | 3.2e–26 | YDL087C, YKL012W, YFL017W-A, YGR013W, YIL061C, YMR125W, YBR119W, YPL178W, YHR086W, YDR240C, YLR275W, YDR235W, YER029C, YML046W, YLR298C, YGR074W |
| GO:0031145 | Anaphase-promoting complex-dependent catabolic process | 10 of 11 genes, 90.9% | 3.6e–26 | YGL240W, YHR166C, YDL008W, YFR036W, YBL084C, YKL022C, YOR249C, YLR127C, YDR118W, YNL172W |
| GO:0042273 | Ribosomal large subunit biogenesis | 15 of 15 genes, 100.0% | 6.5e–26 | YHR052W, YOR272W, YPL093W, YDR060W, YKR081C, YOL041C, YPL043W, YOR206W, YBR142W, YKL014C, YMR049C, YNL002C, YLR276C, YHR066W, YNL061W |
| GO:0006511 | Ubiquitin-dependent protein catabolic process | 19 of 21 genes, 90.5% | 6.6e–25 | YGL004C, YGR184C, YDR394W, YER012W, YLR421C, YER021W, YEL037C, YDL147W, YOR261C, YKL145W, YDR427W, YDL097C, YFR010W, YDL007W, YGL048C, YFR052W, YFR004W, YOR259C, YHR200W |
| GO:0007035 | Vacuolar acidification | 10 of 10 genes, 100.0% | 6.5e–24 | YMR054W, YLR447C, YHR039C-A, YOR332W, YPR036W, YKL080W, YEL051W, YOR270C, YGR020C, YBR127C |

**Table 3** The MCODE and CEBCOA p-values and the number of matching annotations for the best GOTerm

| GOID | GOTerm | Method | Number of annotations | p-value | Genes annotated to the term |
|---|---|---|---|---|---|
| GO:0006378 | mRNA polyadenylation | CEBCOA | 16 (of 22) | 2.9e–39 | YJR093C, YLR115W, YKL059C, YLR277C, YDR195W, YNL222W, YAL043C, YDR228C, YGR156W, YOR250C, YGL044C, YMR061W, YKR002W, YPR107C, YDR301W, YNL317W |
| GO:0006378 | mRNA polyadenylation | MCODE | 12 (of 22) | 3.4e–29 | YJR093C, YLR115W, YKL059C, YLR277C, YDR195W, YAL043C, YGR156W, YDR301W, YMR061W, YKR002W, YPR107C, YNL317W |

into the format of this tool which, this file is zipped and uploaded. Then, 'choose annotation' is selected in the tool leading to open the organism list where we select 'Yeast' or 'Human' based on the input data. A sample of typical results for three Gene Ontology term has been shown in Fig. 22. The second and third columns in Fig. 22 denote the number of proteins (genes) we could anticipate regarding the real data and the genome, respectively. Value "2 of 3 genes" in the second column, for instance, states that of three proteins, two ones were anticipated using our method and value "56 of 7166" in the third column, for instance, states that of 7166 proteins in genome, we could anticipate 56 ones. To calculate the FDR (False Discovery Rate) value, GOTermFinder runs 50 simulations with random genes and reports the average number of times a p-value is as good as or better than the p-value generated from the real data.

Figure 23 shows the tree image for the two Gene annotations (last column in Fig. 22) generated by the GOTermFinder tool where each colored non-leave rectangle node shows a different gene ontology and the leave nodes are genes (proteins). Each colored p-values at the bottom of the figure corresponds a colored gene ontology (non-leave node). In Fig. 23, two colored p-values have corresponding gene ontology and the other four colored p-values have gene ontology in other trees, which not shown here. The less amount a p-value has, the more gene similarity there exist.

Table 2 shows the *p-value* obtained by applying *CEBCOA* to predict 10 complexes (arranged based on its biological significance), which is the best discovered complexes in the network DIP. As stated in the description of Fig. 22, in Table 2, "Cluster frequency" denotes the number of anticipated proteins regarding the real data.

Table 3 shows the *p-values* obtained by applying CEBCOA and MCODE for the best Gene Ontology Term. According to Table 3, the *p-value* for *CEBCOA* is much less than the *p-value* for *MCODE* where GOterm (Gene Ontology term) is *mRNA polyadenylation*. Regarding this GO term, the *p-value* of 12 out of 22 annotations is 3.43e-29 using *MCODE* while it is 2.92e-39 using *CEBCOA*. As stated in the explanation of Relation 35, the less *p-value* a method has, the more significant biological meaning it has.

## 4 Conclusion

Regarding the role of protein complexes in biological processes, identifying protein complexes is an important issue in computational biology. In this paper, a new core-attachment structure has been proposed for detecting protein complexes in weighted networks. The semantic similarity between a pair of proteins has been used to estimate the reliability of protein interactions based on gene ontology. Using combined centrality measures and biological properties, the importance of each protein has been determined. Seed central proteins have in turn been detected by sorting down the measures, after which, by filtering the noisy proteins from the essential proteins, the precision of the selection of central proteins in the core centers has been increased. Further, the primary core of high-weighted density complexes has been determined, and attachment proteins have been subsequently added to the core. Experimental results show that the detection accuracy of protein complexes in the proposed method is significantly improved, compared to other methods. Furthermore, the predicted protein complexes have a high biological significance. In the future, we will integrate additional information, such as domain information and gene expression data, for a more precise prediction of protein complexes.

## References

1. Srihari S et al (2013) A survey of computational methods for protein complex prediction from protein interaction networks. J Bioinform Comput Biol 11:1230002
2. Tu S et al (2010) A binary matrix factorization algorithm for protein complex prediction. In: Proceedings of the BIBM 2010 International Workshop on Computational Proteomics, Hong Kong.
3. Enright AJ et al (2002) An efficient algorithm for large-scale detection of protein families. Nucleic Acids Res 30:1575–1584
4. Adamcsek B et al (2006) CFinder: locating cliques and overlapping modules in biological networks. Bioinformatics 22:1021–1023
5. Liu G et al (2009) Complex discovery from weighted PPI networks. Bioinformatics 25:1891–1897
6. Junker BH et al (2008) Analysis of biological networks. Wiley, New York
7. Bader GD et al (2003) An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinform 4:2
8. Dezső Z et al (2003) Bioinformatics analysis of experimentally determined protein complexes in the yeast Saccharomyces cerevisiae. Genome Res 13:2450–2454
9. Srihari S et al (2010) MCL-CAw: a refinement of MCL for detecting yeast complexes from weighted PPI networks by incorporating core-attachment structure. BMC Bioinform 11:504
10. Peng W et al (2015) Identification of protein complexes using weighted PageRank-Nibble algorithm and core-attachment structure. IEEE/ACM Trans Comput Biol Bioinform (TCBB) 12:179–192
11. Srihari S et al (2015) Methods for protein complex prediction and their contributions towards understanding the organisation, function and dynamics of complexes. FEBS Lett 589:2590–2602
12. Price T et al (2013) Survey: enhancing protein complex prediction in PPI networks with GO similarity weighting. Interdiscip Sci 5:196–210
13. Zaki N et al (2013) Protein complex detection using interaction reliability assessment and weighted clustering coefficient. BMC Bioinform 14:163
14. Elahi AB et al (2018) Identification essential proteins based on a new combination of topological and biological features in weighted protein-protein interaction networks. IET Syst Biol 12:247–257

15. Zhao et al (2014) Detecting protein complexes based on uncertain graph model. IEEE/ACM Trans Comput Biol Bioinform
16. Nepusz T et al (2012) Detecting overlapping protein complexes in protein-protein interaction networks. Nat Methods 9:471
17. Wang J et al (2011) A fast hierarchical clustering algorithm for functional modules discovery in proteininteraction networks. IEEE/ACM Trans Comput Biol Bioinform 8:607–20
18. Kerrien S et al (2006) IntAct—open source resource for molecular interaction data. Nucleic Acids Res 35:D561–565
19. Collins SR et al (2007) Toward a comprehensive atlas of the physical interactome of Saccharomyces cerevisiae. Mol Cell Proteomics 6:439–450
20. Krogan NJ et al (2006) Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. Nature 440:637
21. Pu S et al (2008) Up-to-date catalogues of yeast protein complexes. Nucleic Acids Res 37:825–831
22. Mewes HW et al (2005) MIPS: analysis and annotation of proteins from whole genomes in 2005. Nucleic Acids Res 34:D169–D172
23. Ahn J et al (2013) Improved method for protein complex detection using bottleneck proteins. BMC Med Inform Decis Mak 13:S5
24. Habibi M et al (2010) Protein complex prediction based on k-connected subgraphs in protein interaction network. BMC Syst Biol 4:129
25. Ma X et al (2012) Predicting protein complexes in protein interaction networks using a core-attachment algorithm based on graph communicability. Inf Sci 189:233–254
26. Díaz-Montaña JJ et al (2017) Gfd-net: A novel semantic similarity methodologyfor the analysis of gene networks. J Biomed Inform 1(68):71–82
27. Adamic LA et al (2001) Search in power-law networks. Phys Rev E 64:046135
28. Li M et al (2010) Essential proteins discovery from weighted protein interaction networks. In: International Symposium on Bioinformatics Research and Applications. Springer, Berlin
29. Tang Y et al (2015) CytoNCA: a cytoscape plugin for centrality analysis and evaluation of protein interaction networks. Biosystems 127:67–72
30. Yu Y et al (2012) Detecting protein complexes based on sequence information in the weighted protein-protein interaction network. J Comput Theor Nanosci 9:1565–1570
31. Wu M et al (2009) A core-attachment based method to detect protein complexes in PPI networks. BMC Bioinform 10:169
32. Keretsu S et al (2016) Weighted edge based clustering to identify protein complexes in protein–protein interaction networks incorporating gene expression profile. Comput Biol Chem 65:69–79
33. Brohee S et al (2006) Evaluation of clustering algorithms for protein-protein interaction networks. BMC Bioinform 7:488
34. Asur S et al (2007) An ensemble framework for clustering protein–protein interaction networks. Bioinformatics 23:i29–i40
35. Boyle EI et al (2004) GO: TermFinder—open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. Bioinformatics 20:3710–3715
36. Liu Q et al (2016) Using contrast patterns between true complexes and random subgraphs in PPI networks to predict unknown protein complexes. Sci Rep 6:21223
37. Ruepp A et al (2009) CORUM: the comprehensive resource of mammalian protein complexes—2009. Nucleic Acids Res 38:D497–501