



Article

A Central Edge Selection Based Overlapping Community Detection Algorithm for the Detection of Overlapping Structures in Protein–Protein Interaction Networks

Fang Zhang ^{1,†} , Anjun Ma ^{2,3,†}, Zhao Wang ¹, Qin Ma ^{2,3} , Bingqiang Liu ⁴, Lan Huang ^{1,*} and Yan Wang ^{1,*}

¹ Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education, College of Computer Science and Technology, Jilin University, Changchun 130012, China; jlu_zhangfang@163.com (F.Z.); wzl314521@163.com (Z.W.)

² Bioinformatics and Mathematical Biosciences Lab, Department of Agronomy, Horticulture and Plant Science, South Dakota State University, Brookings, SD 57007, USA; Anjun.Ma@sdstate.edu (A.M.); qin.ma@sdstate.edu (Q.M.)

³ Department of Mathematics and Statistics, South Dakota State University, Brookings, SD 57007, USA

⁴ School of Mathematics, Shandong University, Jinan 250100, China; bingqiang@sdu.edu.cn

* Correspondence: huanglan@jlu.edu.cn (L.H.); wy6868@jlu.edu.cn (Y.W.)

† These authors contributed equally to this work.

Received: 9 September 2018; Accepted: 9 October 2018; Published: 13 October 2018



Abstract: Overlapping structures of protein–protein interaction networks are very prevalent in different biological processes, which reflect the sharing mechanism to common functional components. The overlapping community detection (OCD) algorithm based on central node selection (CNS) is a traditional and acceptable algorithm for OCD in networks. The main content of CNS is the central node selection and the clustering procedure. However, the original CNS does not consider the influence among the nodes and the importance of the division of the edges in networks. In this paper, an OCD algorithm based on a central edge selection (CES) algorithm for detection of overlapping communities of protein–protein interaction (PPI) networks is proposed. Different from the traditional CNS algorithms for OCD, the proposed algorithm uses community magnetic interference (CMI) to obtain more reasonable central edges in the process of CES, and employs a new distance between the non-central edge and the set of the central edges to divide the non-central edge into the correct cluster during the clustering procedure. In addition, the proposed CES improves the strategy of overlapping nodes pruning (ONP) to make the division more precisely. The experimental results on three benchmark networks and three biological PPI networks of *Mus. musculus*, *Escherichia coli*, and *Cerevisiae* show that the CES algorithm performs well.

Keywords: protein–protein interaction network; overlapping community detection; central edge selection; overlapping node pruning

1. Introduction

The majority of the biological processes are constituted by a group of proteins which are connected densely [1]. The protein–protein interaction (PPI) network contains the communications among the protein groups that communicate with each other closely [2], which can be used to predict the complexity or function of normal proteins. The structures of the PPI networks can reflect some principles of the cellular organization [3]. Recently, the graph theory has been widely used to detect potential biological significance in PPI networks by regarding the proteins as nodes and the interactions

between proteins as links [4–6]. Therefore, the procedure of finding new protein families can be converted to the procedure of detecting the sub-graph in PPI networks [7].

In the early 1930s, researchers began to detect community structure in sociology. Moreover, computational identification of special protein molecules is a key issue in understanding protein function [8]. To some extent, the community structure [9] can reflect the topological relations of real communities directly. The real communities, such as PPI networks in biology and World Wide Web (WWW) networks in sociology, tend to have the heavy-tailed power law, that is, only a small part of the nodes' degrees is extremely higher than the rest. Therefore, network detection can be applied to the various research fields, such as biology and sociology [10–14].

Compared with non-overlapping communities, the overlapping community's existence is more widespread in real communities, and comes up with many overlapping community detection (OCD) algorithms, such as CMC [15], MCL-Caw [16], and COACH [17]. In addition, many OCD algorithms which are based on the node calculation in regulatory social networks have had many achievements. In 2005, Palla et al. proposed the clique percolation method (CPM) based on the theory of mass infiltration to analyze the overlapping community structure of networks [18]. In 2011, Kim Y et al. proposed link clustering (LC) based on hierarchical clustering [19]. In 2014, Rodriguez et al. proposed a semi-supervised learning algorithm called density peaks clustering (DPC) to recognize cluster centers based on local density maxima [20]. Later in 2017, Qi Jinshan et al. proposed the OCD algorithm based on central node selection (CNS) to select the more reasonable central nodes [21]. Moreover, algorithms of OCD based on the edge information in regulatory social networks have also arisen. Evans et al. first utilized edge information to detect the overlapping network [22]. In 2010, Ahn Y-Y et al. proposed an OCD algorithm based on edge information to reveal the overlap and hierarchical structures in regulatory social networks [23]. Later in 2013, Lan Huang et al. proposed an extended link similarity (ELC), considering the similarity among the non-neighbor links [24]. Recently, an OCD algorithm was proposed based on links by combining a novel link similarity measure with the classic Markov cluster [25], and Deng X et al. proposed another algorithm combining the edge information with the Markov chain to detect overlapping communities [26]. The algorithms in clustering procedures and overlapping nodes pruning (ONP), such as the nearest neighbor algorithm (NN) [27] and community detection with an adjustable extent of overlapping [28], were developed dramatically. Beyond that, the evaluation algorithms for OCD, such as extended modularity (EQ) [29] and normalized mutual information (NMI) [30,31], have also arisen as needed.

In this paper, we introduce the CNS algorithm and propose an OCD algorithm based on central edge selection (CES), which could combine the advantages of edge selection with node selection. The algorithm consists of three major parts, the CES, the clustering procedure, and the overlapping nodes pruning. Firstly, the theory of community magnetic interference (CMI) was proposed in the process of CNS, considering the influence among the nodes. Then, we proposed the strategy of distance calculation between the non-central edge and the set of the central edge in the clustering procedure. Finally, we improved the algorithm of ONP to obtain more reasonable network divisions. The performance of the proposed algorithm was validated by comparing CES with CNS, CPM, and LC in three benchmark networks and three real PPI networks. The experimental results showed better evaluation values, such as EQ, NMI, and cover rate (CR), of CES in most situations according to the network division. More meaningful divisions can be achieved from CES than the original CNS, the traditional CPM algorithm, and traditional LC algorithm. In Section 2, the relevant algorithms including CNS, CES, the time complexity analysis of the CES, and evaluation methods are introduced. The experimental validation is presented in Section 3, and the conclusion is presented in Section 4.

2. Materials and Methods

2.1. Data Source

In order to assess the viability of CES and compare its performance with other algorithms, five real networks were selected, including three benchmark networks—Zachary’s Karate Club Network [32], Dolphins Social Network [33], and American College Football Network [9]—and two protein interaction networks—*E. coli* Network, *M. musculus* Network, and *Cerevisiae* Network (Table 1).

Table 1. Six real networks.

Dataset	Nodes	Edges	BCN
<i>Karate</i>	34	78	2
<i>Dolphin</i>	62	158	2
<i>Football</i>	115	612	12
<i>E. coli</i>	1396	2092	-
<i>M. musculus</i>	1883	2597	-
<i>Cerevisiae</i>	2172	5124	-

Benchmark networks categories number (BCN) refers to the number of categories on benchmark networks that are recorded in each publication.

The first three benchmark networks describe community networks related to social communications or animal groups. (1) The *Karate network* dataset describes the interaction between every two members affected by two coaches in a karate club at a university in the United States. The nodes and edges refer to students and the communications among them, respectively. The resulting network includes 34 nodes and 78 edges. (2) The *Dolphin network* describes the relationship between two groups of bottlenose dolphins. After seven years of observation by Lusseau et al., a community including 158 edges and 62 nodes was obtained. Each edge represents the intersection between two dolphins, and the relationship in the community is relatively stable. According to the real situation, these dolphins can be divided into two categories. (3) The *Football network*, with 115 nodes and 612 edges, describes the rugby matches in 2000 between 12 different clubs and 115 teams. The nodes, edges, and categories represent different teams, the matches between every two teams, and the 12 clubs, respectively.

The other three datasets are as follows. (1) *E. coli*: This dataset describes the interaction between the proteins in *E. coli*. Each node in the network represents a protein, and an edge between the two nodes represents a relationship between the two proteins. The final network has 1396 nodes and 2092 edges. After removing these networks, the network with 344 nodes and 513 edges can be constructed. This dataset is a core protein interactive of the *E. coli* species, and the dataset name is Ecoli20170205. (2) *M. musculus*: This dataset describes the interaction between the proteins in *M. musculus*. Each node in the network represents a protein, and an edge between the two nodes represents a relationship between the two proteins. The final network has 1883 nodes and 2597 edges. After removing these networks, the network with 941 nodes and 1149 edges can be built. This dataset is a core protein interactive of the *M. musculus* species, and the dataset name is Mmusc20170205. (3) *Cerevisiae*: This dataset describes the interaction between the proteins in *Cerevisiae*. Each node in the network represents a protein, and an edge between the two nodes represents a relationship between the two proteins. The final network has 2172 nodes and 5124 edges. After removing these networks, the network with 2110 nodes and 4936 edges can be built. This dataset is a core protein interactive of the *Cerevisiae* species, and the dataset name is Scere20170205.

2.2. OCD Algorithm Based on Central Node Selection (CNS)

2.2.1. Procedure of the CNS

In 2017, Qi Jinshan and Liang Xun proposed CNS to detect overlapping communities [21], which includes two main steps, the central node selection and the clustering procedure.

(1) In the first step, the exact central nodes can be achieved by evaluating the influence of a node. Suppose that a network $G = (V, E)$ is given, where the $V(G)$ and $E(G)$ represent the set of nodes and edges in the graph G , respectively.

The definition of neighboring nodes of node v is set as the following formula:

$$N(v) = \{v' | (v', v) \in E, v' \in V\} \quad (1)$$

The definition $IB(v_1, v_2)$ of the influence between the node v_1 and the node v_2 is set as the following formula:

$$IB(v_1, v_2) = \frac{D(v_1) \times D(v_2)}{d(v_1, v_2)^2} = \frac{D(v_1) \times D(v_2)}{(1 - SIM(v_1, v_2))^2} \quad (2)$$

where $D(v)$ represents the degree of node v , and $SIM(v_1, v_2) = \frac{N(v_1) \cap N(v_2)}{N(v_1) \cup N(v_2)}$ represents the Jaccard distance between node v_1 and node v_2 .

The definition of all influence of node v is set as the following formula:

$$ALL(v) = \sum_{\substack{v_n \in N(v) \\ v_n \neq v}} IB(v, v_n) = \sum_{\substack{v_n \in N(v) \\ v_n \neq v}} \frac{D(v) \times D(v_n)}{(1 - SIM(v, v_n))^2} \quad (3)$$

The strategy of the central node selection is that if all influence of the node v is more significant than its neighbors, then it is selected as a central node in the community.

(2) In the second step, the non-central nodes can be clustered into the correct categories. Such a clustering procedure extends the communities, which are initialized from each central node. The relationship between a community and nodes is defined as the following formula:

$$attract(EC_i, u) = \frac{\sum_{v \in EC_i \wedge v \in N(u)} IB(u, v)}{ALL(u)} \quad (4)$$

where EC_i represents a community needing to be extended, u represents the neighbor nodes of EC_i , and v represents both the neighboring nodes of u and the nodes in EC_i . The neighboring nodes of EC_i can be enriched by adding nodes with an *attract* value higher than the threshold $\varepsilon = 0.4$ [21]. As a result, the new community can be achieved by iterating the search-and-add of neighboring nodes.

2.2.2. Limitation of CNS

Although the OCD algorithms based on central node selection have many advantages in detecting overlapping communities, such as combining the local information and global information of the regulatory social networks, the accuracy of the central node selection and the overlapping degree of the networks still hold the potential to be expanded. Specifically, considering the fact that the process of central node selection only focuses on the node itself and ignores the influence among the nodes, it may lead to CNS being incorrect. Many constraints should be considered for the formation of the overlapping nodes in each community they belong to, which leads to difficulty in using CNS to achieve overlapping nodes. Therefore, the degree of the overlapping node is insufficient in CNS. In either case, the result of the community detection can hardly match the real network well. For instance, in a small benchmark demo network containing 8 nodes and 12 edges (Figure 1a), the $ALL(v)$ values (Figure 1b)

can be calculated by the CNS algorithm, and node 3 will be regarded as the only central node. While in the benchmark network, two central nodes, node 3 and node 6, will be considered as central nodes.

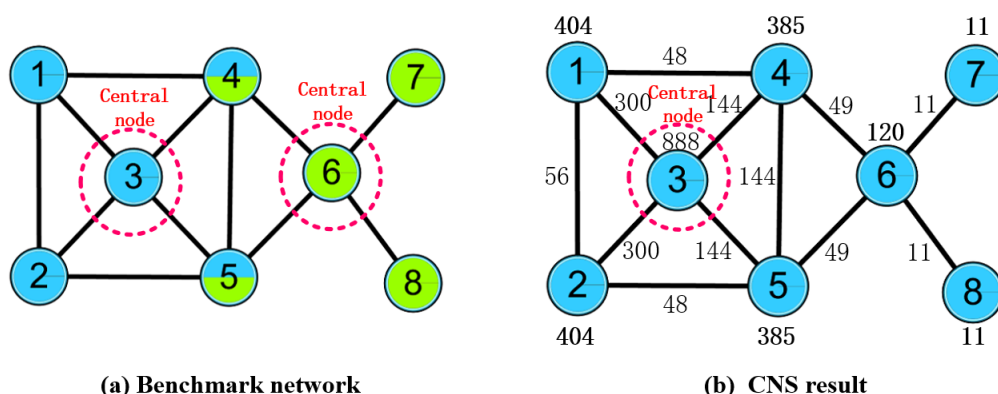


Figure 1. Example of the limitation of the central node selection (CNS) algorithm.

2.3. OCD Algorithm Based on Central Edge Selection (CES)

To avoid the shortcomings of CNS, we proposed CES using the information of edges to detect the overlapping communities. The workflow of the CES algorithm shown in Figure 2 contains three major parts, including a procedure of central edge selection, a clustering procedure, and an ONP step. The theory of CMI, introduced in Section 2.3.1, takes into consideration the influence among nodes to make the target central node more reliable. The network can be divided by edges to reduce the difficulty of getting overlapping nodes, and then optimized by ONP.

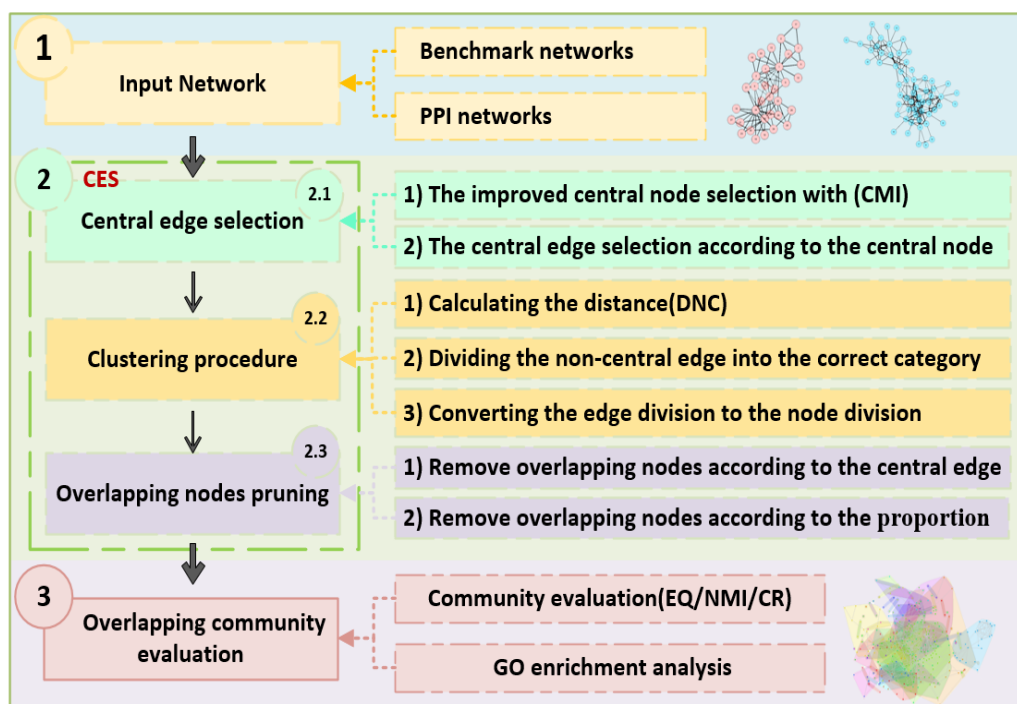


Figure 2. Workflow of the CES algorithm.

2.3.1. Central Edge Selection

The process of central edge selection is composed of two parts: An improved central node selection integrated with CMI, and the central edge selection.

(1) In the first part, the CMI theory is used to improve the process of the central node selection, which alters the central nodes to affect their neighboring nodes. Here, a formula used to revise the *ALL* value of nodes is shown as follows:

$$ALL(v) = GF \times \sum_{u \in N(v)} IB(v, u) \quad (5)$$

where v and u refer to the confirmed central node and its neighboring nodes, respectively. GF is a coefficient used to revise the *ALL* value of nodes according to CMI.

The influence between nodes in the network is calculated using Formula (2), and updates the *ALL* value by Formula (5), after determining one central node using the strategy of CNS in the CNS algorithm.

The pseudo-code of the improved central node selection can be described as following Algorithm 1:

Algorithm 1. Improved central node selection.

```

1 Calculate the all influence of nodes
2 For each  $v \in V$  do
3    $ALL(v) = \sum_{\substack{v_n \in N(v) \\ v_n \neq v}} IB(v, v_n) = \sum_{\substack{v_n \in N(v) \\ v_n \neq v}} \frac{D(v) \times D(v_n)}{(1 - SIM(v, v_n))^2}$ 
4 End for
5 Central node selection with CMI
6 For each  $v \in V$  do
7   Central node selection
8   If  $\forall v' \in N(v) \& \forall v' \notin N(CN)$  and  $ALL(v') \leq ALL(v)$ 
9     Then
10     $CN = CN \cup \{v\}$ 
11     $ALL(v) = 0$ 
12    Revise according to the CMI
13    For each  $v_v \in N(v)$  do
14       $ALL(v_v) = GF \times \sum_{u \in N(v_v)} IB(v_v, u)$ 
15    End for
16 End for

```

where CN refers to the set of central nodes. The $N(CN)$ represent all the neighboring nodes of the confirmed central nodes, which reduces the possibility of two adjacent nodes becoming the central nodes together; as a result, the case where two adjacent nodes are central nodes together cannot occur in the real network.

(2) In the second part, after selecting the central nodes, the procedure of the central edge selection is to classify all the edges connected with the central node as the central edges, and the remaining edges are classified as the non-central edges.

For each central node, the central edges category (CEC) is determined by $CEC(CE_i) = i$, where $CE_i = \{e(v_1, v_2) | v_1 = v \text{ or } v_2 = v\}$ represents the set of the Central Edges linked to a central node with i index, and $e(v_1, v_2)$ represents the edge between node v_1 and node v_2 . Edges other than the central edges are classified as non-central edges.

Considering the same demo network constructed in Section 2.2.2, more reasonable results can fortunately be achieved after recalculating the benchmark network (Figure 1a) with the CES algorithm. In the first circle, we calculate the $ALL(v)$ value (Figure 3a), which is the same as the CNS results (Figure 1b), and regard node 3 as the first central node. Then the values of node 3's neighboring nodes are revised (Figure 3b) according to the theory of CMI, which is introduced in the following

Section 2.3.1. Hence, the other central node, node 6, can be selected, as a result of which the values of node 6’s neighboring nodes are smaller than node 6, and node 1, node 2, node 4, and node 5 are not taken into account. Then, two overlapping nodes can be selected—node 4 and node 5. The result is the same as the benchmark network division (Figure 1a).

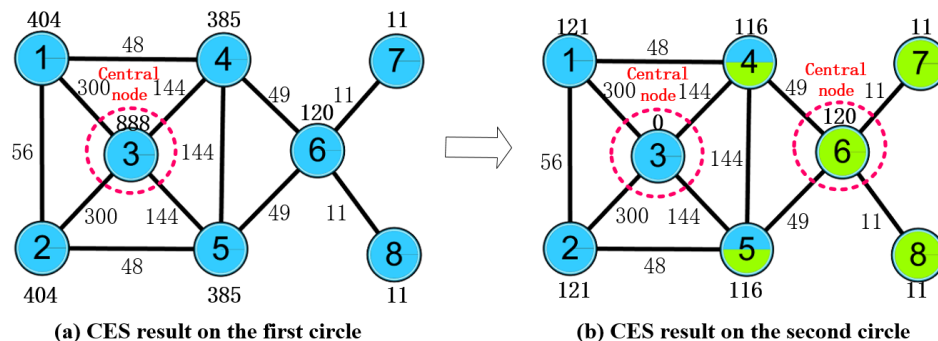


Figure 3. An example of a CES result.

2.3.2. Clustering Procedure

The clustering procedure intakes the result from the procedure of central edge selection to categorize the non-central edges by three steps: Calculating the distance between the non-central edge and the central edges, allocating the non-central edge into the correct category, and converting the edge division into the node division.

(1) In the first part, a novel edge similarity measure $ELC(e_k, e_j)$ [24] is defined as follows to calculate the distance between the edges with edge information.

$$ELC(e_k, e_j) = ELC(e(a, b), e(c, d)) = \frac{|N(a) \cap N(c) + N(a) \cap N(d) + N(b) \cap N(c) + N(b) \cap N(d)|}{|N(a) \cup N(c) + N(a) \cup N(d) + N(b) \cup N(c) + N(b) \cup N(d)|} \quad (6)$$

where $e(a, b)$ represents the edge $e(a, b)$ which has two nodes, node a and node b ; and $N(a)$ represents the neighboring nodes of node a . Therefore, the distance between the non-central edge e_k and the set of the central edges in CE_i can be defined as $DNC(e_k, CE_i)$:

$$DNC(e_k, CE_i) = \sum_{e_j \in CE_i} \frac{ELC(e_k, e_j) \times (\sum_{e_m \in CE_i} ELC(e_k, e_m) - ELC(e_k, e_j))}{\sum_{e_m \in CE_i} ELC(e_k, e_m)} \quad (7)$$

where the e_m and e_j represent the central edges belonging to the categories i .

(2) After calculating all the distances $DNC(e_k, CE_i)$ of e_k , the minimum value of $DNC(e_k, CE_i)$ can be found, and the non-central edge e_k belongs to the corresponding category i based on the NN algorithm [27].

(3) Finally, the remaining edge divisions are converted to the node division. The category of each edge and the corresponding two nodes in the network are the same. In this way, the node division of the network can be achieved as the final result.

2.3.3. ONP Procedure

In this paper, we have improved the ONP algorithm [28] by mixing two strategies. The two strategies are related to each other, and the first strategy is the special case of the second strategy, which can eliminate some steps of pruning and save running time of the CES algorithm.

(1) In the first strategy, overlapping nodes, whose connections are central edges completely in some categories, can be removed from some categories; that is, $con(v_i, C_j) \in CE_i$, where C_j represents the edges in the category j , and $con(v_i, C_j)$ represents the connections between the central node v_i and C_j . It is not necessary to calculate the number of non-central edges between central nodes and categories.

For the example in Figure 4, suggest that node 1 and node 2 are central nodes and node 3 is the overlapping node. According to the first strategy, node 3 can be changed to the left category only; that is, the connection between node 3 and the right category is the edge 2 to 3, which is completely the central edge.

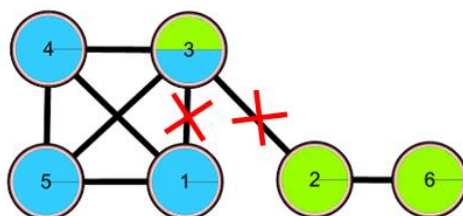


Figure 4. An example of the first pruning strategy.

(2) In the second strategy, the connections of each overlapping node in different categories have a different proportion, and overlapping nodes whose proportion is less than $prop$ can be removed; that is, $\frac{con(v_i, C_j)}{\sum_{k \in clus(v_i)} con(v_i, C_k)} < prop$, where $clus(v_i)$ represents the categories of the node v_i , and the empirical value $prop$ represents the threshold during the ONP.

A simple network is shown in Figure 5 in which node 2 and node 7 are central nodes and node 3 is the overlapping node. The connection between node 3 and the right category has only one non-central edge, while the connection between node 3 and the left category has many non-central edges. Therefore, node 3 will be included in the left category only.

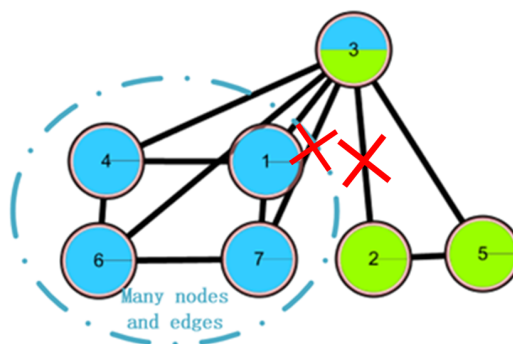


Figure 5. An example of the second pruning strategy.

2.3.4. Time Complexity Analysis

If the network is scale-free, such as the PPI network, then the network obeys the power-law distribution [34]. Suppose n represents the number of nodes, m represents the number of edges, the $seed$ represents the number of central nodes, and $adj(i)$ represents the number of node i 's neighboring nodes. In the procedure of the central edge selection, time is mainly spent in calculating the *all* values of all nodes, which is $O(n^2)$ based on Formula (3), improving central nodes selection based on CMI, which is $O(n \times adj(i))$ according to the improved CNS pseudo-code, and the selection of central edges based on the central node, which is $O(n)$. In the clustering procedure, time is mainly spent in dividing the non-central edges into appropriate categories, which is $O(seed \times m^2)$ according to Formulas (6) and (7). In the ONP procedure, time is mainly spent in finding connections of overlapping nodes in different categories, which is $O(n \times m)$. In the power-law distribution, the degree of each node is the probability of a natural number k where $P(\text{degree} = k) \propto \frac{1}{k^\gamma}$; that is, if a node's degree is k , then the probability is $\frac{1}{k^\gamma}$. In 2001, Béla Bollobás et al. found the $\gamma = 3$ in a big network [35]. The degree of the network is $DN = 1 \times \frac{1}{1^3} + 2 \times \frac{1}{2^3} + \dots + n \times \frac{1}{n^3} \leq \frac{6}{\pi^2} \times n$, and the number of the edges is $m = \frac{DN}{2} \leq \frac{3}{\pi^2} \times n$. So, the final time complexity is $O(n^2 + seed \times m^2 + n + n \times adj(i) + n \times m)$, that is $O(n^2)$. From Table 2, the comparison of the algorithms' running times can be seen clearly.

Table 2. The comparison of the algorithms' running times.

Methods Datasets	RT(s)	Karate	Dolphin	Football	E. coli	M. musculus	Cerevisiae
		CES	0.02	0.105	0.413	1.742	58.746
CNS	0.124	0.609	2.809	67.487	1395.1	15,780	
CPM	0.01	0.3	0.8	1	5	7	
LC	0.636	1.841	7.331	20.988	187	1682.22	

The runtime in seconds (RT(s)) in the table represent the runtime and the bold numbers represent the best RT among all algorithms.

2.4. CPM Algorithm

In 2005, Palla et al. proposed CPM based on the theory of mass infiltration to analyze the overlapping community structure of networks [18]. The result of CPM is based on the conception of K-cliques, which represents the K nodes connected with each other, and the two K-cliques are adjacent if they have (K – 1) common nodes. If K is given, CPM can search all adjacent K-cliques in the networks starting from any K-cliques, and these adjacent K-cliques are divided into the same cluster. Then CPM starts from any K-cliques which are not divided, and starts iteration by searching all adjacent K-cliques. The CFinders package [18] (version 2.0.6, Eötvös University, Budapest, Hungary) is supposed to get the process of the CPM.

2.5. LC Algorithm

In 2011, Kim Y et al. proposed LC based on hierarchical clustering [19]. The advantage of LC is that the node community scheme and link community scheme can be compared quantitatively by measuring the unknown information left in the networks besides the community structure. It can be used to determine quantitatively whether link community schemes should be used rather than node community schemes. However, LC easily achieves the local minimum and tends to divide the communities into small clusters.

2.6. Evaluation

To evaluate the performance of our CES based algorithm we used three evaluation standards, EQ, NMI, and CR, to compare with the performance of CNS and CPM. Specifically, for the PPI network, an additional Gene ontology (GO) enrichment analysis was introduced to evaluate the biological meaning of the network constructed by the four algorithms.

2.6.1. EQ Algorithm

In 2004, Newman et al. proposed an evaluation algorithm module Q, which can be used to evaluate the result of non-overlapping community detections, though it is not suitable to detect overlapping communities. In order to amend the algorithm, in 2009, Shen et al. proposed a novel evaluation EQ algorithm [29].

For a given network $G = (V, E)$, the community $C = \{C_1, C_2, \dots, C_i\}$ $i = 1, 2, \dots, m$ contains m categories, and EQ can be defined as below.

$$EQ = \frac{1}{2 \times link_num} \sum_i \sum_{v \in C_i, w \in C_i} \frac{1}{CN_v CN_w} \left(EB_{vw} - \frac{D(v) \times D(w)}{2 \times link_num} \right) \quad (8)$$

In the formula, m refers to the number of edges in each community, and CN_v and CN_w refer to the number of categories that node v and node w belong to, respectively. EB_{vw} is a logical value that represents the existence status of the edge between node v and node w ; 1 for existent and 0 for missing. $D(v)$ and $D(w)$ represent the degree of node v and node w , respectively. The EQ value ranges from 0 to

1, and a higher value indicates closer structure to the standard division. In the exceptional case, when the result of the community structure is identical to the original standard division, the EQ value is 1.

2.6.2. NMI Algorithm

In 2009, Lancichinetti et al. proposed a novel evaluation algorithm called NMI [30,31], which evaluates the accuracy between the CES result and the standard division. The NMI score ranges from 0 as completely different, to 1 as identical. The following equation defines NMI:

$$NMI(X|Y) = 1 - \frac{1}{2}[H(X|Y)_{norm} + H(Y|X)_{norm}] \quad (9)$$

where X refers to the standard division of the community and Y refers to the CES constructed community division. $H(X|Y)_{norm}$ and $H(Y|X)_{norm}$ are the normalized condition entropy of X with respect to Y , with $H(X|Y)_{norm} = \frac{1}{|NC|} \sum_k \frac{H(X_k|Y)}{H(X_k)}$, where NC represents the number of categories in the network, and X_k represents the network of category k ; and $H(Y|X)_{norm}$ is likewise.

2.6.3. CR Algorithm

The CR is used to describe the coverage of nodes in the community compared to those in the original community. It can be defined as $CR = 100 \times \frac{n'}{n}$, where n' refers to the number of nodes in the produced community division, and n refers to the number of nodes in the original.

2.6.4. GO Enrichment Analysis

In biological network study, GO is a common method used to compare the proteins (or genes) in a predicted network to the known universal functional groups with annotations, and evaluates how close the connections are. Three major aspects are involved in the GO analysis: (1) Biological process (BP) compares the functions or final outcomes of proteins from specific gene sets that carry the same function; (2) molecular function (MF) describes the biochemical activity of the given protein's sets; and (3) cellular component (CC) emphasizes the relative proteins location in a cell and cellular anatomy. For each of the GO enrichment analyses, the p -value is calculated to evaluate the probability predicted protein modules match the protein list annotated to the particular terms. Significant p -values indicate strong association of the proteins with a group. In this paper, we adopt the p -value provided by the R-package ClusterProfiler [36] to analyze the PPI network division.

3. Results and Discussion

3.1. Benchmark Network

The four OCD algorithms (CES, CNS, CPM, and LC) were tested using the three benchmark networks (*Karate*, *Dolphin*, and *Football*), and computational networks were evaluated by three criteria (EQ, NMI, and CR). The evaluation results of four OCD algorithms on three benchmark networks can be seen from Table 3.

Table 3. The evaluation results of four overlapping community detection (OCD) algorithms (CES, CNS, clique percolation method (CPM), and link clustering (LC)) on three benchmark networks (*Karate Network*, *Dolphin Network*, and *Football Network*).

Dataset	Karate					Dolphin					Football				
	EQ	NMI	CR	BCN	ECN	EQ	NMI	CR	BCN	ECN	EQ	NMI	CR	BCN	ECN
CES	0.37	0.92	100%	2	2	0.38	0.76	100%	2	2	0.40	0.52	99%	12	12
CNS	0.35	0.69	100%	2	2	0.46	0.41	100%	2	3	0.28	0.62	44%	12	5
CPM	0.19	0.18	94%	2	3	0.36	0.32	74%	2	4	0.19	0.26	100%	12	4
LC	0.17	0.06	97%	2	12	0.18	1×10^{-16}	87%	2	22	0.16	5.5×10^{-17}	100%	2	46

During the procedure of central edge selection, $GF = 4.2 \times \text{node}_{\text{num}} / \text{edge}_{\text{num}}$ and $prop = \text{node}_{\text{num}} / \text{edge}_{\text{num}}$ during the overlapping nodes pruning, where node_{num} represents the number of nodes in the network and edge_{num} represents the number of edges in the network. Figure 6 represents the selection of GF , which is based on the value of EQ on the three networks. GF is finally selected as $4.2 \times \text{node}_{\text{num}} / \text{edge}_{\text{num}}$. BCN refers to the number of categories on benchmark networks that are recorded in each publication, and evaluation category number (ECN) represents the number of the category which is produced from the algorithms. The bold numbers are the best values among all algorithms.

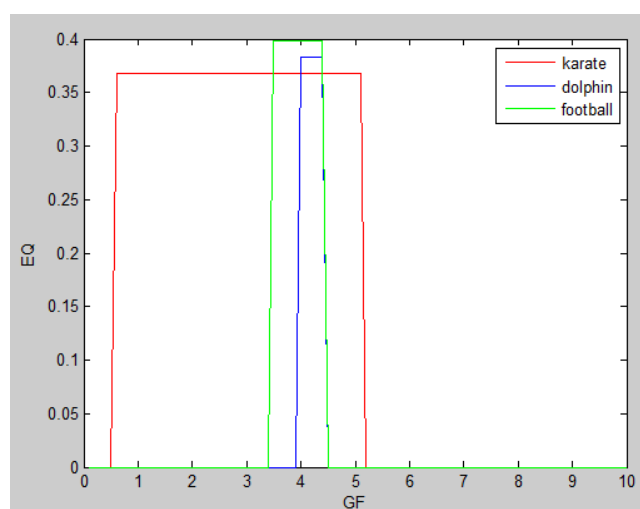
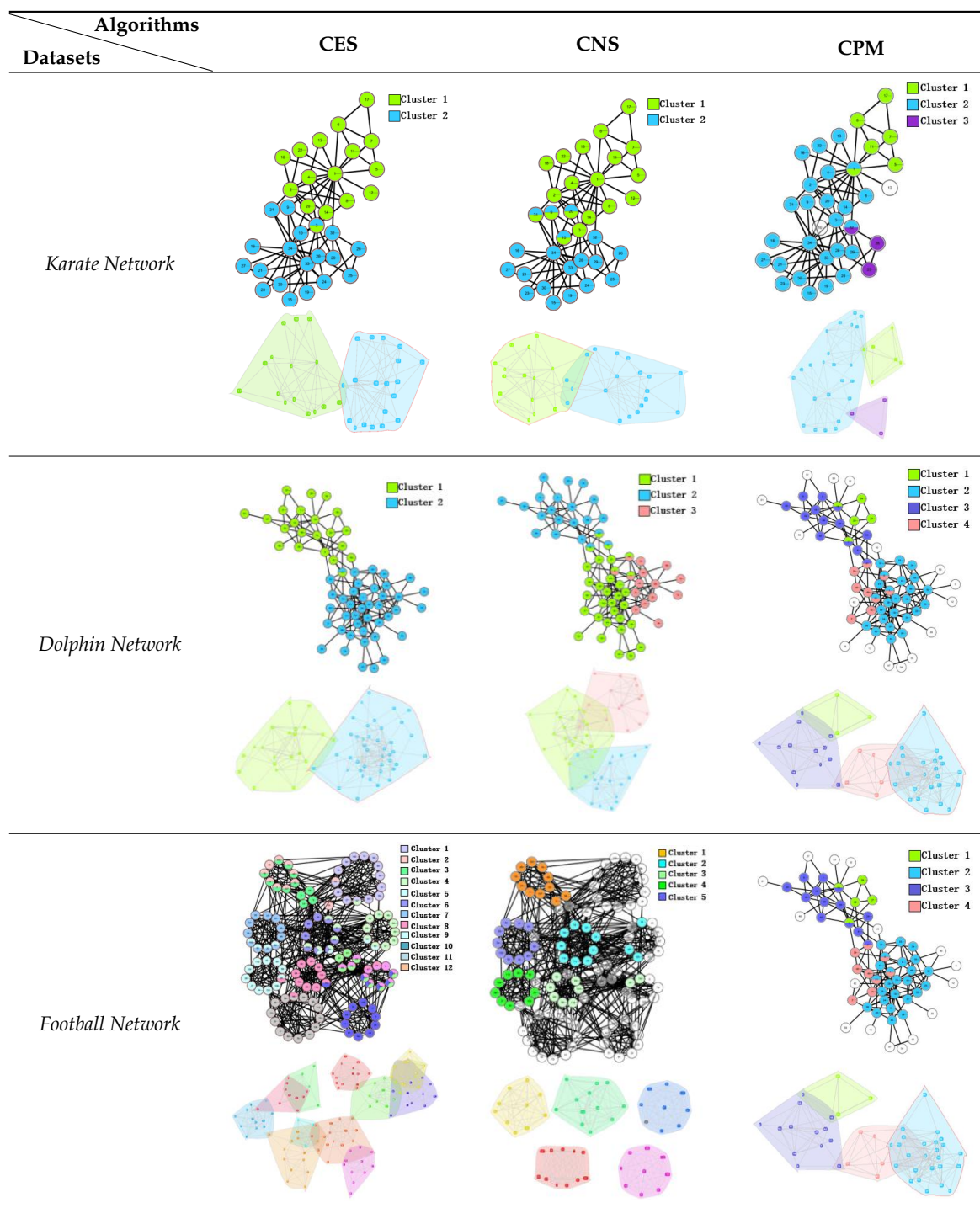


Figure 6. The selection of GF .

For the three datasets, the CES method achieved high scores for all three evaluations, and most of them surpassed the CNS, CPM, and LC methods. Furthermore, the ECN described by CES were identical to the known BCN.

In the *Karate Network*, CES has a better result than CNS, CPM, and LC for all three evaluation methods. The EQ value is 0.37 and the NMI value is 0.92. In addition, CES has a total cover rate. Additionally, the division of the CES has two categories, which is the same as the standard category. In the *Dolphin Network*, CES has a better result than CNS, CPM, and LC in NMI, with a value of 0.76. CES's EQ value is 0.38, which is slightly lower than CNS, as a result of which the number of the category CES has is the same as the number of the standard category, while CNS is inconsistent. Therefore, CNS is inaccurate in getting the correct number of categories, and the high EQ value of CNS has no significance. In addition, CES has a total cover rate. In the *Football Network*, CES has a better result than CNS, CPM, and LC in EQ, with a value of 0.4. CES's NMI value is 0.52 which is slightly lower than CNS, as a result of which the number of the category CES has is the same as the number of the standard category, while CNS is inconsistent. Hence, CNS is inaccurate on getting the correct number of categories, and the high NMI value of CNS has no significance. In addition, CES has a 99% cover rate and is almost completely covered. The visualization of the four algorithms' (CES, CNS, CPM, and LC) results on the three benchmark networks (*Karate Network*, *Dolphin Network*, and *Football Network*) is shown in Table 4, and Cytoscape [37] is used to visualize the network division. In addition, the results of LC on the three benchmark networks have big differences in the number of categories from the benchmark, so the results are meaningless and we do not show the results of LC.

Table 4. The visualization of three algorithms' (CES, CNS, and CPM) results on three benchmark networks (*Karate Network*, *Dolphin Network*, and *Football Network*).



3.2. PPI Network

Three PPI networks, from *M. musculus*, *E. coli*, and *Cerevisiae*, were used to test and compare the performance of the four OCD algorithms (CES, CNS, CPM, and LC). The *GF* values used for each dataset were chosen as 0.9, 0.8, and 0.5, respectively, and 0.1 *prop* for among the datasets. In each dataset, the CES method showed higher EQ and CR than CNS, CPM, and LC (Table 5). The categories found by CES in the three datasets covered all nodes (proteins) in the population, while CNS only covered 65%, 72%, and 55%, respectively, LC only covered 78%, 60%, and 92%, respectively, and the

CPM covered less. Table 5 displays all categories found by the four algorithms. The LC results show much more overlap among categories in each dataset, which induced higher network redundancy, and thus, is far away from the actual protein network structures. The visualization of the predicted PPI network using four algorithms can be seen from Figure 7.

Table 5. The results of four algorithms (CES, CNS, CPM, and LC) on three Protein–Protein Interaction (PPI) networks (*M. musculus* Network, *E. coli* Network, and *Cerevisiae*).

Dataset	<i>M. musculus</i>			<i>E. coli</i>			<i>Cerevisiae</i>		
Evaluation	EQ	CR	ECN	EQ	CR	ECN	EQ	CR	ECN
CES	0.719	100%	85	0.519	100%	77	0.562	100%	105
CNS	0.534	65%	43	0.49	72%	18	0.438	55%	46
CPM	0.191	18%	41	0.226	23%	19	0.467	53%	161
LC	0.19	78%	149	0.10	60%	47	0.06	92%	580

The bold numbers represent the best result among all algorithms.

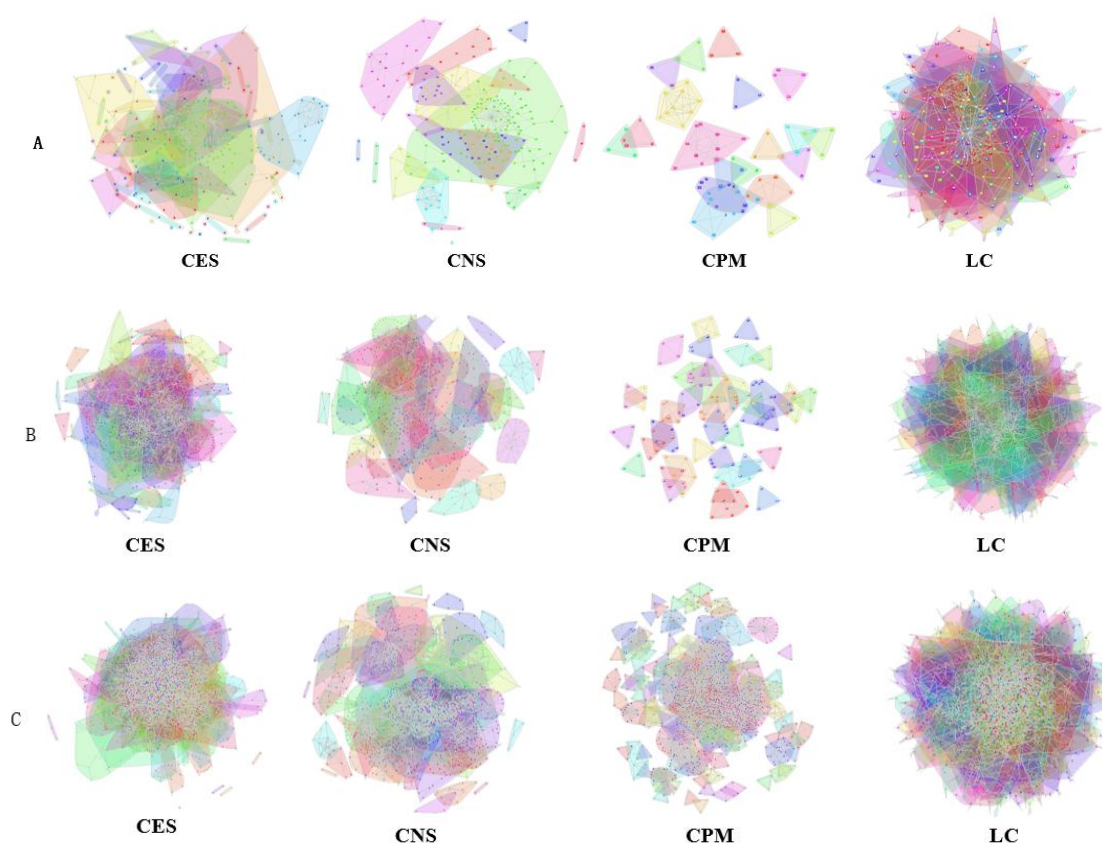


Figure 7. Visualization of the predicted PPI network using four algorithms. (A) *M. musculus* dataset. (B) *E. coli* dataset. (C) *Cerevisiae* dataset.

By performing GO enrichment analysis, the p -values of BP, ME, and CC were calculated to evaluate the connections between the predicted categories and biological functional protein groups (see details in Supplementary Table S1). Considering the overall performance among algorithms, we considered categories (protein modules) with a p -value < 0.001 as significant, and the total number of significant categories are summarized in Table 6. For most cases, the number of significant categories predicted by CES was more than those from CNS, CPM, and LC; the CPM showed a higher rate of significant categories while only presenting a relatively local relationship due to the low CR results, and the LC algorithm excessively categorized the nodes that lead to higher numbers of the total

and significant categories with higher biases. Nevertheless, combining with the overall CR, the CES algorithm still showed the best results for community categories prediction. The individual p -values were log-normalized and are distributed in Supplementary Figures S1–S3 in order to showcase the overall comparison among algorithms and datasets.

Table 6. Total number of significant categories with p -value ≤ 0.001 predicted by each algorithm.

Datasets	CES	CNS	CPM	LC
<i>M. musculus</i>	66/85	33/43	40/41	118/149
<i>E. coli</i>	44/77	17/18	15/19	10/47
<i>Cerevisiae</i>	79/105	44/46	159/161	344/580

Two categories predicted by the CES algorithm, No. 3 in *M. musculus* and No. 1 in the *E. coli* dataset, were selected to showcase the investigation of the relationships among categories and overlapped nodes. For the No.1 significant category in *E. coli*, six proteins, *iscA*, ECs3391, ECs3395, HSCB, *hscA*, and ISCU, were included. Protein *hscA*, responsible for the transfer of iron-sulfur clusters, was considered as the central node and contributed to the enriched category function. The No. 1 category was found to overlap with the 10th and 13th categories, and shared a common overlapping protein, ISCU, which assembles the Fe-S clusters. The 1st and 10th categories overlapped at two more protein positions, ECs3391 and ECs3395, other than ISCU. ECs3391 is an iron-sulfur protein that helps the assembly of Fe-S clusters, and ECs3395 is a scaffold protein that works with ISCU in the formation of Fe-S clusters. The overall relationships of the three categories are shown in Figure 8. The individual protein functions can be found in Supplementary Table S2, along with the overlapping investigation in *M. musculus*.

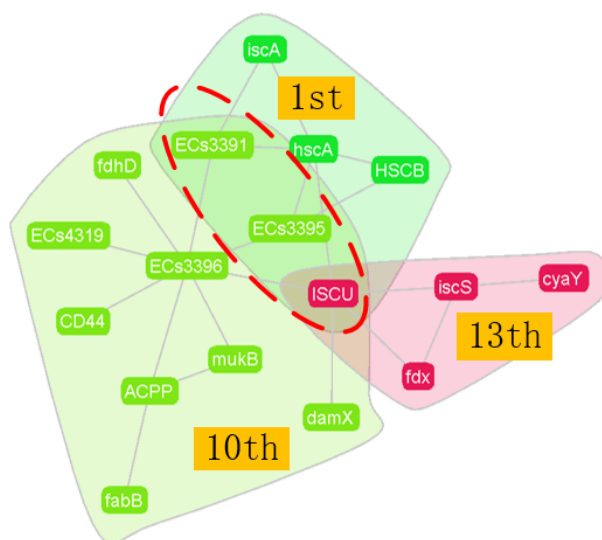


Figure 8. Overlapping structure of the No.1 category in *E. coli*.

4. Conclusions

In this study, a CES based OCD algorithm was introduced to construct community networks. The improved CES method applies the CMI algorithm in the traditional central node selection step, and combines with central edge selection to use both nodes and edge information for the main community construction. Then, the clustering procedure calculates the distance between the non-central edge and central edge to allocate the non-central edges into the right categories. Finally, an improved ONP algorithm is applied to assign the overlapping nodes into an appropriate community to complete the network construction. To evaluate the performance of network construction, the proposed CES method was used to test three benchmark networks and two protein–protein interaction networks, and compared with the CNS, CPM, and LC methods. The results indicated excellent performance of

the CES algorithm in the community with moderate complexities. As a result, we believe our CES algorithm has the potential to achieve more accurate and sufficient networks for community studies, especially in sociology and the systematic biology area. Our future work will focus on improving the efficiency and accuracy of the CES algorithm, and adapting it to dynamic network analyses.

Supplementary Materials: Figure S1: Comparison of three levels on *M. musculus* Network, Figure S2: Comparison of three levels on *E. coli* Network, Figure S3: Comparison of three levels on *Cerevisiae* Network.

Author Contributions: Conception and design of experiments: L.H., Y.W., and Q.M.; Conduction of experiments: F.Z., Y.W., A.M., and Z.W.; Data analyses: F.Z., A.M., and Y.W.; Manuscript draft: Y.W., L.H., F.Z., A.M., Z.W., B.L., and Q.M.

Funding: This work is supported by the National Natural Science Foundation of China (Nos. 61472159, 61572227, 61772313, and 61432010), the Development Project of Jilin Province of China (Nos. 20160204022GX, 2017C033-3, and 20180414012GH) and the Jilin Provincial Key Laboratory of Big Data Intelligent Computing (20180622002JC), and this work is supported in part by Premier-Discipline Enhancement Scheme supported by Zhuhai Government and Premier Key-Discipline Enhancement Scheme supported Guangdong Government Funds. Support for this project was also provided by an RO1 award #1R01GM131399-01 from the National Institute of General Medical Sciences of the National Institutes of Health. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by the National Science Foundation grant number ACI-1548562.

Acknowledgments: The authors are grateful to all the individuals who participated in this study. In particular, we thank Guangchuang Yu for providing “ClusterProfiler” source code of R packages and technical assistance. We also thank Ille’s Farkas and the CFinder’s team for providing the CFinder package and support.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cui, G.; Shrestha, R.; Han, K. ModuleSearch: Finding functional modules in a protein–Protein interaction network. *Comput. Methods Biomech. Biomed. Eng.* **2012**, *15*, 691. [[CrossRef](#)] [[PubMed](#)]
2. Sriganesh, S.; Chern, H.Y.; Limsoon, W. *Computational Prediction of Protein Complexes from Protein Interaction Networks*; ACM Books and Morgan and Claypool: New York, NY, USA, 2017.
3. Li, M.; Wang, J.; Chen, J. A Graph-Theoretic Method for Mining Overlapping Functional Modules in Protein Interaction Networks. *Bioinf. Res. Appl.* **2008**, *4983*, 208–219.
4. Diez, D.; Hutchins, A.P.; Miranda-Saavedra, D. Systematic identification of transcriptional regulatory modules from protein–protein interaction networks. *Nucleic Acids Res.* **2014**, *42*, e6. [[CrossRef](#)] [[PubMed](#)]
5. Wang, Y.; Wang, G.; Meng, D.; Huang, L.; Cui, J.; Blanzieri, E. A Markov Clustering Based Link Clustering Method for Overlapping Module Identification in Yeast Protein–Protein Interaction Networks. *Bioinform. Res. Appl.* **2014**, *8492*, 28–30.
6. Vinayagam, A.; Zirin, J.; Roesel, C.; Hu, Y.; Yilmazel, B.; Samsonova, A.A. Integrating protein–protein interaction networks with phenotypes reveals signs of interactions. *Nat. Methods* **2013**, *11*, 94–99. [[CrossRef](#)] [[PubMed](#)]
7. Jonsson, P.F.; Cavanna, T.; Zicha, D.; Bates, P.A. Cluster analysis of networks generated through homology: Automatic identification of important protein communities involved in cancer metastasis. *BMC Bioinform.* **2006**, *7*, 2. [[CrossRef](#)] [[PubMed](#)]
8. Zou, Q.; He, W. Special Protein Molecules Computational Identification. *Int. J. Mol. Sci.* **2018**, *19*, 536.
9. Girvan, M.; Newman, M.E.J. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826. [[CrossRef](#)] [[PubMed](#)]
10. Lancichinetti, A.; Fortunato, S. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E* **2009**, *80*, 016118. [[CrossRef](#)] [[PubMed](#)]
11. Lancichinetti, A.; Fortunato, S. Community detection algorithms: A comparative analysis. *Phys. Rev. E* **2009**, *80*, 056117. [[CrossRef](#)] [[PubMed](#)]
12. Fortunato, S. Community detection in graphs. *Phys. Rep.* **2010**, *486*, 75–174. [[CrossRef](#)]
13. Raghavan, U.N.; Albert, R.; Kumara, S. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* **2007**, *76*, 036106. [[CrossRef](#)] [[PubMed](#)]
14. Xie, J.; Kelley, S.; Szymanski, B.K. Overlapping Community Detection in Networks: The State-of-the-Art and Comparative Study. *ACM Comput. Surv.* **2013**, *45*, 43. [[CrossRef](#)]
15. Enright, A.J.; Van, D.S.; Ouzounis, C.A. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.* **2002**, *30*, 1575–1584. [[CrossRef](#)] [[PubMed](#)]

16. Srihari, S.; Kang, N.; Hon, L. MCL-CAw: A refinement of MCL for detecting yeast complexes from weighted PPI networks by incorporating core-attachment structure. *BMC Bioinform.* **2010**, *11*, 504. [[CrossRef](#)] [[PubMed](#)]
17. Wu, M.; Li, X.; Kwok, C.K.; Ng, S.K. A core-attachment based method to detect protein complexes in PPI networks. *BMC Bioinform.* **2009**, *10*, 1–16. [[CrossRef](#)] [[PubMed](#)]
18. Palla, G.; Derenyi, I.; Farkas, I.; Vicsek, T. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **2005**, *435*, 814–818. [[CrossRef](#)] [[PubMed](#)]
19. Kim, Y.; Jeong, H. Map equation for link communities. *Phys. Rev. E* **2011**, *84*, 026110. [[CrossRef](#)] [[PubMed](#)]
20. Rodriguez, A.; Laio, A. Clustering by fast search and find of density peaks. *Science* **2014**, *344*, 1492–1496. [[CrossRef](#)] [[PubMed](#)]
21. Qi, J.; Liang, X.; Yi, W. Overlapping community detection algorithm based on selection of seed nodes. *Appl. Res. Comput.* **2017**, *34*, 3534–3537.
22. Evans, T.S.; Lambiotte, R. Line graphs, link partitions, and overlapping communities. *Phys. Rev. E* **2009**, *80*, 016105. [[CrossRef](#)] [[PubMed](#)]
23. Ahn, Y.-Y.; Bagrow, J.P.; Lehmann, S. Link communities reveal multiscale complexity in networks. *Nature* **2010**, *466*, 761. [[CrossRef](#)] [[PubMed](#)]
24. Huang, L.; Wang, G.; Wang, Y.; Blanzieri, E.; Su, C. Link Clustering with Extended Link Similarity and EQ Evaluation Division. *PLoS ONE* **2013**, *8*, e66005. [[CrossRef](#)] [[PubMed](#)]
25. Wang, G.; Huang, L.; Wang, Y.; Pang, W.; Ma, Q. Link community detection based on line graphs with a novel link similarity measure. *Int. J. Modern Phys. B* **2016**, *30*, 1650023. [[CrossRef](#)]
26. Deng, X.; Li, G.; Dong, M.; Ota, K. Finding overlapping communities based on Markov chain and link clustering. *Peer-to-Peer Netw. Appl.* **2017**, *10*, 411–420. [[CrossRef](#)]
27. Angiulli, F. Fast condensed nearest neighbor rule. In Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 7–11 August 2005; pp. 25–32.
28. Wu, Z.; Lin, Y.; Wan, H.; Tian, S. A fast and reasonable method for community detection with adjustable extent of overlapping. In Proceedings of the International Conference on Intelligent Systems and Knowledge Engineering, Hangzhou, China, 15–16 November 2010; pp. 376–379.
29. Khorasgani, R.R.; Chen, J.; Zaiane, O.R. Top leaders community detection approach in information networks. In Proceedings of the SNA-KDD, Washington, DC, USA, 10 July 2010.
30. Danon, L.; Diaz-Guilera, A.; Duch, J.; Arenas, A. Comparing community structure identification. *J. Stat. Mech. Theory Exp.* **2005**, *2005*, 09008. [[CrossRef](#)]
31. Lancichinetti, A.; Fortunato, S.; Kertész, J. Detecting the overlapping and hierarchical community structure in complex networks. *New J. Phys.* **2009**, *11*, 033015. [[CrossRef](#)]
32. Zachary, W.W. An information flow model for conflict and fission in small groups. *J. Anthropol. Res.* **1977**, *33*, 452–473. [[CrossRef](#)]
33. Lusseau, D.; Schneider, K.; Boisseau, O.J.; Haase, P.; Slooten, E.; Dawson, S.M. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behav. Ecol. Sociobiol.* **2003**, *54*, 396–405. [[CrossRef](#)]
34. Guerriero, V. Power law distribution: Method of multi-scale inferential statistics. *J. Modern Math. Front.* **2012**, *1*, 21–28.
35. Bollobás Be Riordan, O.; Spencer, J.; Tusnády, G. The degree sequence of a scale-free random graph process. *Random Struct. Algorithms* **2001**, *18*, 279–290. [[CrossRef](#)]
36. Yu, G.; Wang, L.-G.; Han, Y.; He, Q.-Y. clusterProfiler: An R Package for Comparing Biological Themes Among Gene Clusters. *OMICS J. Integrat. Biol.* **2012**, *16*, 284–287. [[CrossRef](#)] [[PubMed](#)]
37. Shannon, P.; Markiel, A.; Ozier, O.; Baliga, N.S.; Wang, J.T.; Ramage, D. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Res.* **2003**, *13*, 2498–2504. [[CrossRef](#)] [[PubMed](#)]

Sample Availability: Samples of the compounds are not available from the authors.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).