electronics

MDPI

Article

A Network Clustering Algorithm for Protein Complex Detection Fused with Power-Law Distribution Characteristic

Jie Wang [1,*], Ying Jia [1], Arun Kumar Sangaiah [2,3,*] and Yunsheng Song [4]

[1] School of Information, Shanxi University of Finance and Economics, Taiyuan 030006, China; 13007526396@163.com
[2] International Graduate Institute of Artificial Intelligence, National Yunlin University of Science and Technology, Douliou 64002, Taiwan
[3] Department of Electrical and Computer Engineering, Lebanese American University, Byblos 1102-2801, Lebanon
[4] School of Information Science and Engineering, Shandong Agricultural University, Taian 271018, China; songys@sdau.edu.cn
* Correspondence: 20191031@sxufe.edu.cn (J.W.); aksangaiah@ieee.org (A.K.S.); Tel.: +86-351-7666-126 (J.W.)

Abstract: Network clustering for mining protein complexes from protein–protein interaction (PPI) networks has emerged as a prominent research area in data mining and bioinformatics. Accurately identifying complexes plays a crucial role in comprehending cellular organization and functionality. Network characteristics are often useful in enhancing the performance of protein complex detection methods. Many protein complex detection algorithms have been proposed, primarily focusing on local micro-topological structure metrics while overlooking the potential power-law distribution characteristic of community sizes at the macro global level. The effective use of this distribution characteristic information may be beneficial for mining protein complexes. This paper proposes a network clustering algorithm for protein complex detection fused with power-law distribution characteristic. The clustering algorithm constructs a cluster generation model based on scale-free power-law distribution to generate a cluster with a dense center and relatively sparse periphery. Following the cluster generation model, a candidate cluster is obtained. From a global perspective, the number distribution of clusters of varying sizes is taken into account. If the candidate cluster aligns with the constraints defined by the power-law distribution function of community sizes, it is designated as the final cluster; otherwise, it is discarded. To assess the prediction performance of the proposed algorithm, the gold standard complex sets CYC2008 and MIPS are employed as benchmarks. The algorithm is compared to DPClus, IPCA, SEGC, Core, SR-MCL, and ELF-DPC in terms of F-measure and Accuracy on several widely used protein–protein interaction networks. The experimental results show that the algorithm can effectively detect protein complexes and is superior to other comparative algorithms. This study further enriches the connection between analyzing complex network topology features and mining network function modules, thereby significantly contributing to the improvement of protein complex detection performance.

Keywords: data mining; network clustering; protein complex detection; power-law distribution; topological characteristics

Citation: Wang, J.; Jia, Y.; Sangaiah, A.K.; Song, Y. A Network Clustering Algorithm for Protein Complex Detection Fused with Power-Law Distribution Characteristic. *Electronics* **2023**, *12*, 3007. https://doi.org/10.3390/electronics12143007

Academic Editor: Ping-Feng Pai

Received: 17 June 2023
Revised: 6 July 2023
Accepted: 6 July 2023
Published: 8 July 2023

1. Introduction

Cells rely on the interaction of multiple proteins for life activities. A protein complex, formed through interactions, consists of molecules with similar functions. Detecting protein complexes in protein–protein interaction (PPI) networks facilitates the exploration of the relationships between network structures and function modules. Moreover, it plays a crucial role in annotating the proteins with unknown functions and gaining insights into the organization and functionality of cells [1].

Researchers have proposed many experimental methods to identify the interactions between proteins, including yeast two-hybrid (Y2H) [2,3] and tandem affinity purification (TAP) [4]. These methods have generated a vast amount of protein–protein interaction (PPI) data, which serve as valuable support for the application of data mining techniques in protein complex detection.

A PPI dataset is usually abstracted as an undirected network, wherein proteins are nodes and the interactions between proteins are edges. A PPI network contains different protein function modules [5]. Generally, a protein complex is a biological functional module [6] comprising two or more proteins that share the same function. Proteins in the same protein complex exhibit strong connections, whereas the proteins belonging to different complexes have weaker connections. Detecting protein complexes from PPI networks aims to discover sets of proteins with dense connections. This process can be viewed as a network clustering task, wherein clusters are determined based on topological features, where the connection strength within a cluster is greater than that between clusters [7,8]. This process yields disjoint or overlapping clusters as its outcome [9].

Various network clustering algorithms for identifying protein complexes have been developed. In general, these algorithms include graph partition algorithms, density-based local search algorithms, and algorithms based on graph embedding [10–12].

The clustering algorithm based on graph partition divides nodes into clusters according to an objective function, aiming to identify an optimal partitioned network. It maximizes the similarity between nodes within each cluster while minimizing the similarity between different clusters. One well-known algorithm in this category is the Markov algorithm (MCL) [13,14]. MCL begins by constructing the initial flow matrix based on a PPI network and then simulates random flow through the network using the concept of random walk to partition the entire network into sub-graphs with high connectivity probability. The collection of nodes within each sub-graph represents a protein complex. However, MCL does not handle overlapping clusters. To address this limitation, the soft regularized MCL (SR-MCL) algorithm was developed, which enables the identification of overlapping clusters.

The density-based local search clustering algorithm focuses on identifying dense sub-graphs based on the characteristic of connection density. Among the various network clustering methods, one approach aims to find *k*-closely connected sub-network modules, such as the Closely Connected Percolation Method (CPM) [15]. CPM initially identifies closely connected subnets within the network and subsequently identifies *k*-closely connected subnet modules based on these initial subnets. A few approaches are also known as the seed expansion method. They select a node as a seed and expand around the seed to a cluster according to certain rules. One example of the seed expansion method is the density peak clustering (DPClus) algorithm [16]. DPClus introduces the concept of "cluster periphery" in protein interaction networks. It assigns edge weights based on common neighbor counts between interacting proteins, while node weights are determined by the sum of their adjacent edges' weights. The peripheral value of a node within a cluster is determined as the ratio of its adjacent nodes to the total number of nodes in the cluster. The algorithm starts by selecting the highest-weighted node as the seed for the initial cluster. Edge weights are influenced by common neighbor counts, and node weights reflect the density of immediate neighbors. If nodes satisfy both the custom threshold for local density and the threshold for cluster peripheral value, DPClus iteratively adds the nodes to obtain the final cluster. To account for the minimum diameter and average node distance characteristics of protein complexes, the improved DPClus algorithm (IPCA) [17] enhances DPClus through the integration of sub-graph diameters and interaction probabilities, which provide insights into the density of the network. Other methods in this category include SEGC [18], Core [19], etc.

Network clustering algorithms based on graphs embed map network nodes onto a lower-dimensional vector space by encoding their properties [20]. This mapping preserves the topological characteristics of the nodes as much as possible. Subsequently, a network

clustering was performed in this transformed vector space [21,22]. One example of such an algorithm is the ensemble learning framework for density peak clustering (ELF-DPC) [23]. ELF-DPC first maps the PPI network to the vector space and constructs a weighted network to identify core edges. By integrating structural modularity and trained voting regression models, the algorithm creates an ensemble learning model. ELF-DPC then expands the core edges into clusters based on this learning model.

The PPI network, as a type of complex network, exhibits intricate network topology characteristics [24–26]. The fundamental features used to describe the network topology are primarily derived into three levels. Firstly, micro-topological structure metrics focus on individual nodes or edges, including measures such as node degree and centrality [27,28]. Secondly, meso-topological metrics analyze groups of nodes, such as community structure [29], modules, and motifs. Lastly, macro-topological metrics consider the entire network, encompassing aspects such as degree distribution and community size distribution. Developing a network clustering algorithm that incorporates these network features can enhance the accuracy of community detection [30]. At present, seed expansion methods can effectively utilize network features. However, existing algorithms mainly consider local micro-topological structure features [31] and ignore the potential distribution characteristics of community size at a macro-global level. The distribution of community sizes in the PPI network exhibits a certain correlation with power-law distribution [32].

In this paper, we present a novel network clustering approach that incorporates the characteristics of power-law distribution to identify protein complexes. Our proposed algorithm, named GCAPL, encompasses two main stages: cluster generation and cluster determination. During the cluster generation stage, the GCAPL algorithm incorporates node degree and clustering coefficient to assign weights to nodes. The unclustered nodes with the highest weight were selected as seeds. Following that, a cluster generation model leveraging the scale-free power-law distribution was given to discovery clusters with dense centers and sparse peripheries. Through an iterative process, candidate nodes were added to the seeds to form candidate clusters using the cluster generation model. In the cluster determination stage, we constructed a power-law distribution function about the distribution of cluster sizes and the cluster total number. The function acts as a criterion to regulate the presence of clusters of various sizes. By applying the power-law distribution function, we can assess whether a candidate cluster qualifies as a final cluster.

This paper makes several significant contributions: (1) Integrating multiple available basic micro-topological structural information into the k-order neighborhood of a node for seed selection; (2) Constructing a cluster generation model considering scale-free power-law distribution to obtain inherent organization information of functional modules; (3) Giving a cluster determination model based on macro-topological structure characteristic of the number distribution of clusters of different sizes to constrain final clusters; (4) Verifying the proposed network clustering algorithm fused with topological structural information could effectively mine functional modules by the experiment results on the real datasets.

The other sections of our paper are as follows. Section 2 introduces preliminary concepts and symbols. Section 3 presents a network clustering algorithm fused with power-law distribution characteristics. Section 4 reports the relevant experiments to verify the effectiveness of the network clustering algorithm. Section 5 provides conclusions.

2. Preliminary

A PPI network is represented by an undirected network G = (V, E), with V as the set of proteins (nodes) and E as the set of interactions (edges) between proteins. *Dia(G)* represents the diameter of the network G, which corresponds to the maximum value in the shortest path between any two nodes in the network G. The k-adjacent nodes set of a given node v_i is denoted as $NE_k(v_i)$, and it is defined by

$$NE_k(v_i) = \begin{cases} NE(v_i) & if\ k = 1 \\ NE_{k-1}(v_i) \cup \{v_j \in V | distance(v_i, v_j) = k\} & if\ k > 1 \end{cases} \quad (1)$$

where $distance(v_i, v_j)$ represents the length of the distance between nodes v_i and v_j.

The clustering coefficient of v_i [33] is

$$CCE(v_i) = \frac{2|ES(H(v_i))|}{|NE(v_i)|(|NE(v_i)| - 1)} \quad (2)$$

where $H(v_i)$ represents sub-graph created by the directly adjacent node set $NE(v_i)$, and $ES(H(v_i)) = \{(v_j, v_l)|v_j, v_l \in NE(v_i), (v_j, v_l) \in E\}$. A network's clustering coefficient $CCE(v_i)$ is calculated as the average value of the clustering coefficients of all nodes in the node set V, i.e., $\overline{CCE(G)} = \sum_{i=1}^{|V|} CCE(v_i)$. In order to facilitate readers' reading of this paper, some main symbols and their corresponding meanings are listed in Table 1.

Table 1. Main symbols and their corresponding meanings.

Symbols	Meaning
$G = (V, E)$	Network G is composed of a collection of nodes V and set of edges E.
v_i	Node i in a certain node set.
(v_i, v_j)	The edge between nodes i and j.
$distance(v_i, v_j)$	The shortest path distance between nodes i and j.
NE_k	Set of k-neighbors.
$ES(M)$	The set of edges within sub-graph M.
CCE	The clustering coefficient for a node
ND	The degree of a node in the network.
$w(.)$	The weight for a node or an edge
X_{size}	Set of cluster sizes
Y_{num}	Set of cluster numbers
$CT(u, M)$	The tightness measure of node u with respect to sub-graph M.
$CS(v)$	Node set generated by the selected seed v.
$Dia(G)$	The diameter of a network G
PC	Final cluster set
λ	Rate of change

3. Methods

GCAPL algorithm consists of two stages: cluster generation and cluster determination. In the first stage, the algorithm calculates the weights of nodes and edges by incorporating micro-topological structure metrics. A seed is the node that has the highest weight among the unclustered nodes. The seed is expanded by a cluster generation model considering a scale-free power-law distribution to a candidate cluster. In the second stage, we established the cluster determination model with a power-law distribution of the cluster numbers with different cluster sizes. This cluster decision model was utilized to determine the final clusters. Figure 1 shows the algorithm flow chat.

3.1. Cluster Generation

In the cluster generation stage, the GCAPL algorithm initially selects seeds based on node weights and subsequently expands these seeds using the cluster generation model to obtain candidate clusters.

To identify a suitable seed node, a node with a higher weighted degree may be a good seed node in network community mining. A node with a higher weighted degree may serve as a useful seed node in network community mining. The weighted degree of a node v_i is calculated based on its directly adjacent edges and the weights associated with these edges, and was defined as:

$$w(v_i) = \sum_{v_j \in NE(v_i)} w(v_i, v_j) \quad (3)$$

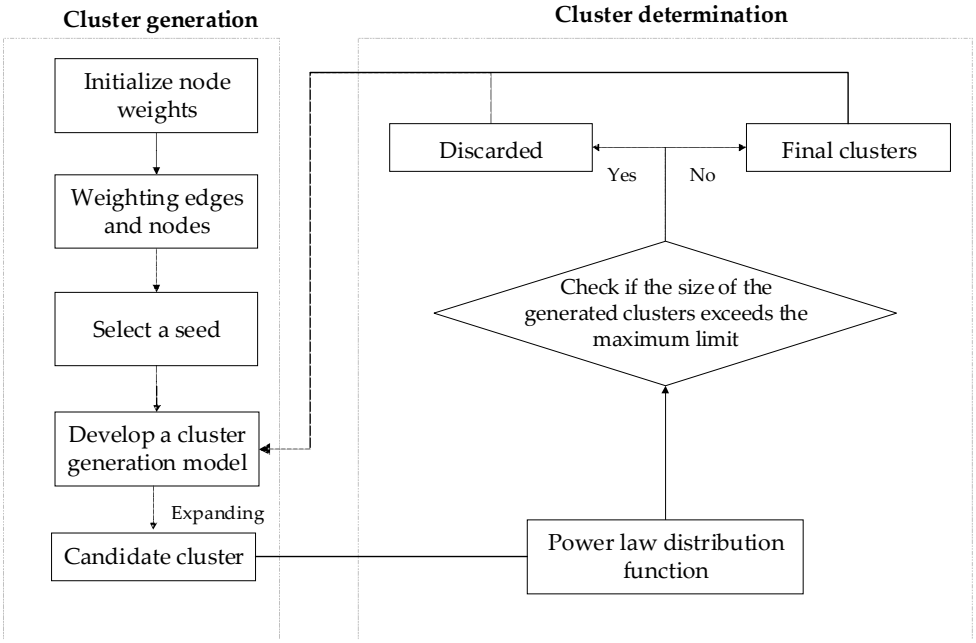


Figure 1. Algorithm flow chart.

For an edge (v_i, v_j), the endpoints of the edge and the common adjacent nodes between these endpoints are tightly surrounding this edge. We can obtain the edge weight of (v_i, v_j) according to the importance of these nodes in topological characteristics. The micro-topological structure metrics, such as clustering coefficient and node degree, are employed to capture the topological characteristics and assign weights to nodes. For a dense submodule in a network, nodes with high clustering coefficients and low node degrees may serve as important central nodes. The topological characteristics of a node v_i is expressed by the ratio of its clustering coefficient $CCE(v_i)$ to its node degree $ND(v_i)$, i.e., $CCE(v_i)/ND(v_i)$. More comprehensively, the global information of a network is introduced. A network G's clustering coefficient *FC(G)* is defined as the average value of the clustering coefficients of all nodes in the node set V, i.e., $FG(G) = \overline{CCE(G)}$. Similarly, the G's node degree *FD(G)* is the average of all node degrees in the network, i.e., $FD(G) = \overline{ND(G)}$. In the network G, the connection strength of a node v_i is related to $\frac{CCE(v_i)}{\overline{CCE(G)}} \times \frac{\overline{ND(G)}}{ND(v_i)}$. Therefore, the weight of the edge (v_i, v_j) can be defined as follows:

$$w(v_i, v_j) = \frac{CCE(v_i)}{CCE(G)} \times \frac{\overline{ND(G)}}{ND(v_i)} + \frac{CCE(v_j)}{CCE(G)} \times \frac{\overline{ND(G)}}{ND(v_j)} + \sum_{u \in NE(v_i) \cap NE(v_j)} \frac{CCE(v_u)}{\overline{CCE(G)}} \times \frac{\overline{ND(G)}}{ND(v_u)} \tag{4}$$

Furthermore, Equation (4) from the previous section only considers the information of the node's direct neighbors. To highlight the importance of an edge within a large network module, the edge weight in its t neighborhood can be defined as follows:

$$\begin{aligned} w^t(v_i, v_j) = \; & w^{t-1}(v_i) \times \frac{CCE(v_i)}{\overline{CCE(G)}} \times \frac{\overline{ND(G)}}{ND(v_i)} + \\ & w^{t-1}(v_j) \times \frac{CCE(v_j)}{\overline{CCE(G)}} \times \frac{\overline{ND(G)}}{ND(v_j)} + \\ & \sum_{u \in NE(v_i) \cap NE(v_j)} w^{t-1}(u) \times \frac{CCE(v_u)}{\overline{CCE(G)}} \times \frac{\overline{ND(G)}}{ND(v_u)} \end{aligned} \tag{5}$$

Here, t is a predefined parameter that determines the extent of the neighborhood. After the t-th iteration, the node weight can be defined as:

$$w^t(v_i) = \sum_{v_j \in NE(v_i)} w^t(v_i, v_j) \tag{6}$$

Initially, the node weights are set to $w^0(v_i) = 1$ for all nodes, indicating that the initial importance of all nodes is the same.

Once the node weight calculation is completed, the next step is to select a seed node v from the node set V whose node weight is highest. Following that, the seed node is used to establish the cluster generation model, which allows for the expansion of the seed into a candidate cluster.

The cluster generation model aims to expand seed nodes into candidate clusters based on connection strength. The obtained seed node v serves as the initial cluster *CS(v)*, and candidate nodes from the neighborhood *NE(C(v))* are considered for addition based on the compactness of *CS(v)* and the connection strength between *CS(v)* and a candidate node u to expand the initial cluster *CS(v)*. The compactness g of the cluster *CS(v)* quantifies the connection density within the cluster and is defined as $g(u, CS(v)) = |NE(u) \cap V(CS(v))|/|V(CS(v))|$, where $V(CS(v))$ represents a set of nodes that make up *C(v)*, and $|NE(u)|$ denotes the node u's direct neighbor nodes. The connection strength h of a candidate node u reflects the peripheral edges of the cluster and is defined as $h(u, CS(v)) = |NE(u) \cap V(CS(v))|/|NE(u)|$. The cluster generation model requires a variable function to combine the compactness of the cluster and the peripheral edges of the cluster, so that as the cluster size increases, the contribution of the cluster's compactness to the cluster generation gradually decreases while the contribution of the cluster's peripheral connections to the cluster generation gradually increases. A suitable choice for this function is the scale-free power-law distribution function, which is a monotonic function. It serves as a foundation for constructing the variable function that effectively fuses the above two kinds of connection information. A power-law distribution function is $y = cx^{-k}$ and let $c = 1/\lambda$, $k = ND(v)$, $x = V(CS(v)) - 1$, then we can define the variable function as:

$$\beta(CS(v)) = \frac{1}{\lambda \times \sqrt[ND(v)]{V(CS(v)) - 1} + 1} \tag{7}$$

where λ is a parameter to control the change of $\beta(CS(v))$. Then, define the cluster generation model as:

$$\begin{aligned} CT(u, CS(v)) = \beta(CS(v))g(u, CS(v)) + \\ (1 - \beta(CS(v)))h(u, CS(v)) \end{aligned} \tag{8}$$

When $\beta(CS(v))$ is set to 1, CT tends to prioritize the formation of dense clusters. On the other hand, when $\beta(CS(v))$ is set to 0, nodes with lower degrees are more likely to be added to *CS(v)*. The $\beta(CS(v))$ enables the cluster generation model to find both dense clusters and clusters with dense cores and sparse peripheries, providing flexibility in capturing different types of cluster structures. For each candidate node u and threshold $\mu \in [0, 1]$, if $CT(u, CS(v)) > \mu$ and $Dia([CS(v) \cup \{u\}]) \leq \delta$ (δ is a user-defined threshold), then the node u is added to the cluster *CS(v)*. This process is repeated for each node in *NE(CS(v))*, resulting in the initial formation of a candidate cluster *CS(v)*.

3.2. Cluster Determination

In complex networks, the distribution of community size exhibits heterogeneity. Smaller communities tend to be more abundant in number, while larger communities are relatively scarce. This inverse relationship between size and number also holds in PPI networks, where the sizes and numbers of protein complexes are inversely proportional. It is assumed that the number of complexes follows a power-law distribution that is defined as follows:

$$y = cx^{-k} \tag{9}$$

where x and y represent positive random variables.

Let the size of a protein complex be X_{size}. The corresponding number of the complexes under this size is given by a cluster determination model:

$$Y_{num} = cX_{size}{}^{-k} \tag{10}$$

where c and k are positive parameters.

The cluster determination model aims to effectively regulate the number of clusters, considering their varying sizes, from a global perspective. To accomplish this, we defined two sequences: $X_{size} = \{x^1_{size}, x^2_{size}, \ldots, x^n_{size}\}$ is a predefined sequence with uniform values representing the cluster sizes, and $Y_{num} = \{y^1_{num}, y^2_{num}, \ldots, y^n_{num}\}$ is a sequence obtained through the cluster generation model representing the corresponding cluster numbers.

Let the cluster *CS(v)*'s size be denoted as $|V(CS(v))|$ and $error_{size}$ be a parameter that refers to the allowable difference or deviation in the size of a cluster. Following that, we can find a value x^i_{size} with $|V(CS(v))| \in \left[x^i_{size} - error_{size}, x^i_{size} + error_{size}\right]$ in X_{size}, assuming that $y^{'}_{num}$ clusters of size x^i_{size} have been generated at the current stage. We calculated the maximum number of clusters y^i_{num} corresponding to a given cluster size x^i_{size} according to the power-law distribution function. If $y^{'}_{num} \leq y^i_{num}$, the candidate cluster *CS(v)* is considered a final cluster. Otherwise, *CS(v)* is discarded.

The two stages of cluster generation and cluster determination are repeated alternately until all nodes have been clustered.

3.3. Complexity Analysis

The GCAPL algorithm utilizes linked lists to construct a graph. First, it calculates the weights of all nodes using Formula (6). Following that, it selects the node with the highest weight as the seed and treats it as the initial cluster. Subsequently, following the cluster generation model, neighbor nodes of the initial cluster are incrementally added to create candidate clusters. Finally, the algorithm determines the final clusters by the cluster determination model. The specific process of the GCAPL algorithm is shown in Algorithm 1.

Algorithm 1: GCAPL Algorithm.

Input: Network $G = (V, E)$, Parameters *iter*, λ, μ for cluster generation, Parameters c, k, $error_{size}$ for Cluster determination
output: Set of final clusters *PC*
1: Initialize $PC = \varnothing$, and the unclustered nodes set, $UV = V$;
2: Compute edge and node weights by utilizing information within the t-neighborhood;
3: Determine the cluster size set $X_{size} = \{x^1_{size}, x^2_{size}, \ldots, x^n_{size}\}$;
4: Calculate the upper limit of the number of clusters $Y_{num} = \{y^1_{num}, y^2_{num}, \ldots, y^n_{num}\}$, corresponding to the cluster size X_{size} using Equation (9);
5: **while** $UV \neq \varnothing$, do
6: Select a node v with the largest weight in UV as a seed, and the initial cluster is *CS(v)*;
7: Iteratively select the node set *AN* among the neighbor nodes of *CS(v)*, such that each node u in *AN* satisfies $CT(u, H) > \mu$ and $Dia([CS(v) \cup \{u_i\}]) \leq \delta$;
8: $CS(v) = CS(v) \cup AN$;
9: Compute the cluster *CS(v)*'s size as $|V(CS(v))|$, and compute the number of generated clusters with size $|V(CS(v))|$ as $y^{'}_{num}$;
10: Find x^i_{size} in X_{size}, and $|V(CS(v))| \in \left[x^i_{size} - error_{size}, x^i_{size} + error_{size}\right]$
11: Compute the number of generated clusters of size $|V(CS(v))|$ as $y^{'}_{num}$
12: **if** $y^{'}_{num} \leq y^i_{num}$ **then**
13: $PC = PC \cup \{CS(v)\}$,$UV = UV - CS(v)$
14: return *PC*

The time cost of the GCAPL algorithm lies in two parts: cluster generation and cluster determination.

Assuming a network G has n nodes and m edges. In the cluster generation stage, the node weighting process revealed a time cost of $O(k \times \overline{ND} \times n) = O(k \times m)$. The time cost of seed selection based on node weights is $O(n \times \log n)$. The expansion of seeds into clusters also has a time cost of $O(n \times \log n)$. Therefore, $O(|PC| \times n \times \log n)$ is the total time complexity of the cluster generation phase.

In the cluster determination phase, the worst-case scenario is when each candidate cluster size needs to be compared with each element in the sequence X_{size}. As a result, this phase revealed a time cost of $O(n \times |X_{size}|)$. Therefore, algorithm GCAPL's overall time complexity is $O(|PC| \times n \times \log n)$, considering both the cluster generation and cluster determination phases.

4. Experiments and Results

4.1. Datasets

The protein interaction networks used in the experiments are presented in Table 2. These datasets were processed to remove self-intersections and duplicate interactions.

Table 2. Datasets of protein interaction networks.

	Gavin02 [34]	**Gavin06 [35]**	**K-Extend [36]**	**BioGRID [37]**
Proteins	1352	1430	3672	4187
Interactions	3210	6531	14,317	20,454

The gold standard complex datasets CYC2008 [38] and MIPS [39] were utilized for parameter analysis and evaluation of the clustering results.

4.2. Evaluation Metrics

The evaluation of the effectiveness of the GCAPL algorithm was performed using the F-measure and Accuracy metrics as evaluation criteria.

The F-measure [40] provides a balanced measure of precision and recall. It serves as a quantitative metric of the agreement between a predicted complex set and a benchmark complex set, capturing the level of similarity between them. Precision measures the agreement between the generated clusters and known complexes, while recall quantifies the agreement between the known complexes and the generated clusters.

Given the generated cluster as $PC = \{PC_1, PC_2, \ldots, PC_p\}$ and the gold standard complex as $TC = \{TC_1, TC_2, \ldots, TC_l\}$, the affinity score within the neighborhood $NA(PC_i, TC_j)$ is employed for quantifying the similarity between the generated cluster PC_i and the standard complex TC_j, and $NA(PC_i, TC_j) = |PC_i \cap TC_j|^2 / |PC_i| \times |TC_j|$, $i \in \{1, 2, \ldots, p\}$, $j \in \{1, 2, \ldots, l\}$. A higher $NA(PC_i, TC_j)$ value indicates a stronger resemblance between PC_i and TC_j. Assuming a threshold of $\mu = 0.2$ [40,41], if $NA(PC_i, TC_j) \geq \mu$, PC_i and TC_j can be considered as matched. Let M_C represent the set of correct predictions, where each generated cluster exhibits some correspondence with at least one known protein complex in the set TC, and $M_C = \{PC_i | PC_i \in PC \wedge \exists j(TC_j \in TC \wedge NA(PC_i, TC_j) \geq \mu)\}$. Additionally, let M_{CO} be the set of known complexes, where each complex matches at least one complex in the generated cluster set PC, and $M_{CO} = \{TC_j | TC_j \in TC \wedge \exists i(PC_i \in PC \wedge NA(PC_i, TC_j) \geq \mu)\}$.

Precision is quantitatively calculated as the ratio of the number of correctly predicted instances to the total number of predicted instances, i.e., $Precision = |M_C|/|PC|$. Recall is defined as $Recall = |M_{CO}|/|TC|$. F-measure is quantitatively calculated as

$$F - measure = 2 \times Precision \times Recall / (Precision + Recall) \quad (11)$$

Accuracy, as another evaluation metric, is computed as the geometric mean of the positive predictive value (PPV) and sensitivity (Sn). PPV represents the proportion of correctly identified positive instances among the predicted instances, while Sn measures the proportion of correctly identified positive instances among all actual positive instances.

Suppose T is a $p \times l$ matrix, in which the i-th row of T represents the i-th prediction cluster PC_i and the j-th column represents the j-th annotation complex TC_j. T_{ij} denotes the count of shared proteins between the predicted complex PC_i and the known complex TC_j and quantifies the degree of overlap or similarity between these two complexes. PPV is characterized by

$$PPV = \frac{\sum_{i=1}^{p}\sum_{j=1}^{l}\left(T_{ij} \times \max_{j=1}^{l}\left(\frac{T_{ij}}{\sum_{j=1}^{l} T_{ij}}\right)\right)}{\sum_{i=1}^{p}\sum_{j=1}^{l} T_{ij}} \tag{12}$$

Sn is defined as

$$Sn = \frac{\sum_{j=1}^{l}\left(\left|TC_j\right| \times max_{i=1}^{p}\left(\frac{T_{ij}}{\left|TC_j\right|}\right)\right)}{\sum_{j=1}^{l}\left|TC_j\right|} \tag{13}$$

Accuracy [39] is then calculated as

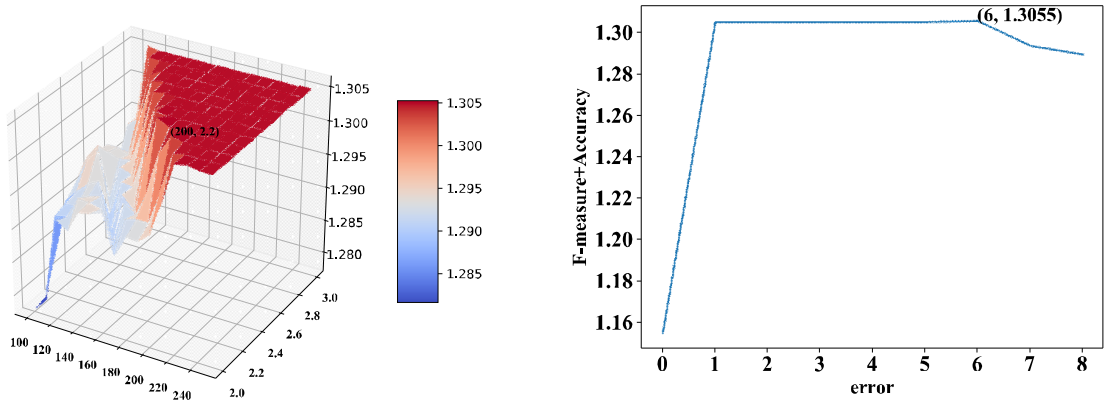
$$Accuracy = \sqrt{PPV \times Sn} \tag{14}$$

4.3. Parametric Analysis

GCAPL encompasses several predefined parameters, including c, $k \in [2,3]$, $error_{size}$, $iter$, $\lambda \in [0,1]$, and $\mu \in [0,1]$. The coefficients c and k correspond to the coefficients and exponents of the power-law distribution function, respectively. The $error_{size}$ is a cluster size error. The $iter$ refers to the count of repetitive steps. The λ stands for an adaptive parameter. The μ is defined as the compactness threshold. The BioGrid dataset serves as a standard protein interaction network dataset, wherein all interactions are derived from reliable and precise low-throughput theoretical interactions. Consequently, on this dataset, the parameter optimization aims to maximize the value of $F-measure + Accuracy$, prompting a thorough parameter analysis to identify the optimal parameter value.

The analysis of parameters c, k, and $error_{size}$ was performed to investigate the impact of these parameters on the algorithm. The coefficient c and the exponent k were utilized to generate the sequences X_{size} and Y_{num} based on the power-law distribution function. Meanwhile, the parameter $error_{size}$ was employed to regulate the error tolerance in cluster size. Initially, the analysis focuses on varying c and k while keeping the parameter $error_{size}$ constant. Subsequently, the investigation shifts to studying the influence of the parameter $error_{size}$ while maintaining c and k at constant values.

We first fixed $error_{size} = 6$, and experiments were conducted on the BioGrid PPI network to investigate the impact of the parameters c and k. The values of c ranged from 100 to 250, while k varied from 2.0 to 3.0. These experiments aimed to assess how the changes in c and k influenced the results and outcomes of the study. When the values of $c = 200$ and $k = 2.2$ are set, the $F-measure + Accuracy$ metric attains a higher value. Next, we first fixed $c = 200$, $k = 2.2$ and $F-measure + Accuracy$ is maximized at $error_{size} = 6$. We set $c = 200$, $k = 2.2$, $error_{size} = 6$. In Figure 2a, the impact of parameters c and k on $F-measure + Accuracy$ is illustrated, with $error_{size} = 6$. The relationship between the parameter $error_{size}$ and $F-measure + Accuracy$ are depicted in Figure 2b, with $c = 200$ and $k = 2.2$.



(**a**) (**b**)

Figure 2. Performance impact analysis of parameters on BioGRID dataset: (**a**) analyze c and k; (**b**) analyze $error_{size}$.

Next, we kept the values of $c = 200$, $k = 2.2$, and $error_{size} = 6$ fixed, and analyzed the real-valued discrete parameters: the number of iterations *iter*, the adjustment parameter $\lambda \in [0, 1]$ of the change rate, and the tightness threshold $\mu \in [0, 1]$. Considering the interdependence among these parameters, an orthogonal matrix was employed to identify the optimal parameter combination with a high likelihood. During the experimental design phase, each parameter variable was treated as an independent factor. Feasible values corresponding to these factors are assigned as distinct levels. The complete set of parameter combinations represents the experimental space. An orthogonal array L36 (63×37) is employed, which comprises 36 parameter combinations. There are parameters, $iter \in \{1, 2, 3, 4, 5, 6\}$, $\lambda \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$, and $\mu \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$, that we exclusively consider the initial three columns of the orthogonal array to facilitate the analysis. Among the 36 parameter combinations, the one with the highest $F - measure + Accuracy$ is selected as the optimal configuration. Through the experiments, the parameters are set to $iter = 2$, $\lambda = 0.1$, and $\mu = 0.4$.

4.4. Power-Law Distribution Analysis

This subsection examines the power-law distribution of network clustering results, taking the BioGRID dataset as an example. The clustering result of this dataset was utilized to explore the relationship between the cluster size and number.

Assume that the cluster size is represented by x and the corresponding number of clusters is denoted by y. According to Equation (9), we have $y = cx^{-k}$. By performing logarithm operations on both sides of the equation, it represents that

$$\ln y = \ln c - k \ln x \tag{15}$$

It was observed that $\ln y$ and $\ln x$ exhibit a linear relationship. Thus, the analysis of the power-law distribution of x and y was transformed into a linear relationship analysis of $\ln x$ and $\ln y$.

In the clustering result of the BioGRID dataset, we took the logarithm of the cluster size x and the corresponding cluster number y, resulting in transformed variables $x' = \ln x$ and $y' = \ln y$. To explore whether there is a linear relationship between x' and y', a linear fitting method was performed on x' and y'. The results of the linear fitting analysis conducted on x' and y' is shown in Figure 3, providing valuable insights into the nature of their relationship.

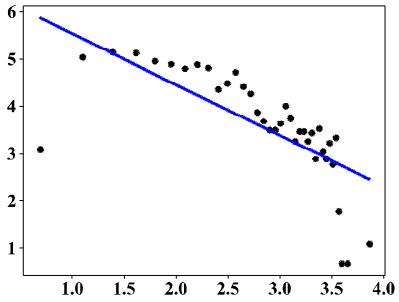


Figure 3. Fitting curve of x' and y'.

Table 3 presents the calculated p-value and R^2 for the linear fitting analysis conducted on x' and y'. A small p-value indicates a strong fit of the clustering result, demonstrating good fitting effectiveness. Similarly, a large value of R^2 suggests a favorable fit. In Table 3, the obtained p-value is 9.9×10^{-7}, and the value of R^2 is 0.5. Thus, the sizes of clusters generated by the proposed algorithm in the PPI network follow a power-law distribution, along with the corresponding numbers of these clusters.

Table 3. Fitting effect of x' and y'.

Criteria	Value
p-value	$9.90771462 \times 10^{-7}$
R^2	0.5001443526421876

4.5. Comparative Experiment

To assess the algorithm's performance, we compared the GCAPL algorithm with several other algorithms, namely DPClus, IPCA, SEGC, Core, SR-MCL, and ELF-DPC.

Figure 4a–d present the experimental results on the Gavin02, Gavin06, K-extend, and BioGRID datasets, using CYC2008 as the standard set. The results demonstrate that, compared to other algorithms, the GCAPL algorithm achieves comparable or higher F-measure and Accuracy values. The GCAPL algorithm performs well in terms of $F - measure + Accuracy$. Compared with other algorithms, the *F-measure* + *Accuracy* of GCAPL exhibits an average improvement of 13.12%, 6.97%, 14.43%, and 14.39% on Gavin02, Gavin06, K-extend, and BioGRID. In addition, the SEGC algorithm demonstrates lower F-measure and Accuracy performance compared to GCAPL on the Gavin02, K-extend, and BioGRID datasets. On the Gavin06 dataset, the DPClus algorithm performs better than other algorithms, except for the GCAPL algorithm. The GCAPL algorithm has a similar framework to the two algorithms mentioned above, and incorporating macro-topological information contributes to improving complex detection performance. By considering both the micro-topological structure of a network and the macro-topological structure feature of the power-law distribution, the GCAPL algorithm effectively detects protein complexes.

Figure 5a–d illustrate the evaluation results of the DPClus, IPCA, SEGC, Core, SR-MCL, ELF-DPC, and GCAPL algorithms on the Gavin02, Gavin06, K-extend, and BioGRID datasets, respectively, using MIPS as the standard set. The GCAPL algorithm consistently exhibits superior values of F-measure and Accuracy across the four different PPI datasets compared to compared algorithms. Compared with other algorithms, the *F-measure* + *Accuracy* of GCAPL exhibits an average increase of 9.90%, 7.01%, 14.34%, and 13.63% on Gavin02, Gavin06, K-extend, and BioGRID. This indicates that the GCAPL algorithm performs well in terms of its ability to detect protein complexes.

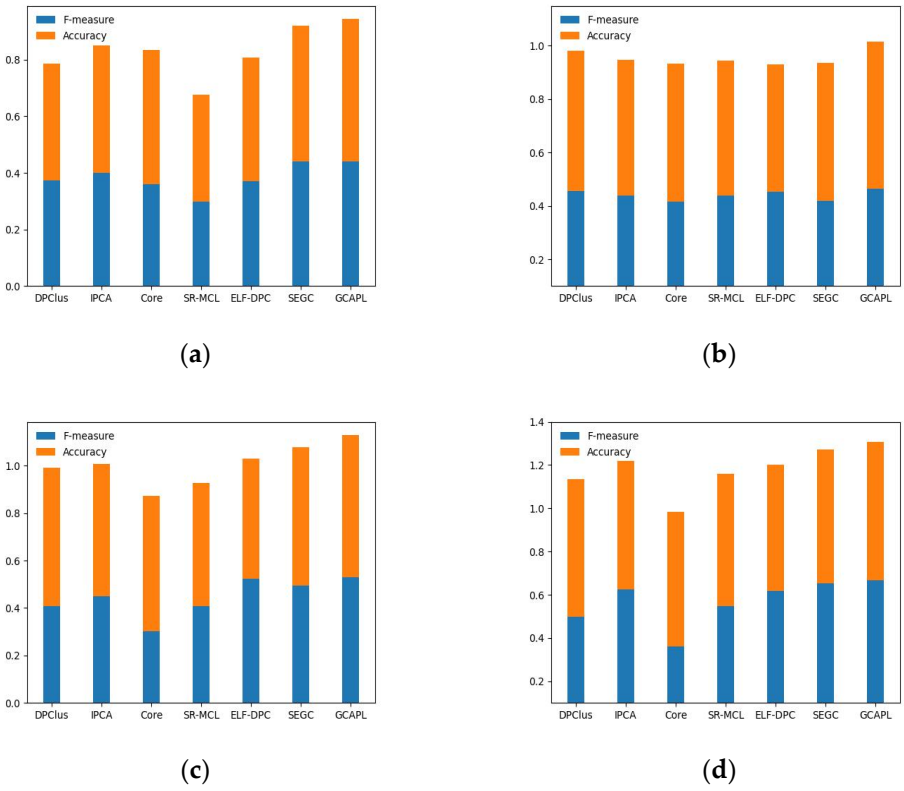


Figure 4. CYC2008 as benchmarks. Evaluation results by different algorithms on (**a**) Gavin02; (**b**) Gavin06; (**c**) K-extend; (**d**) BioGRID.

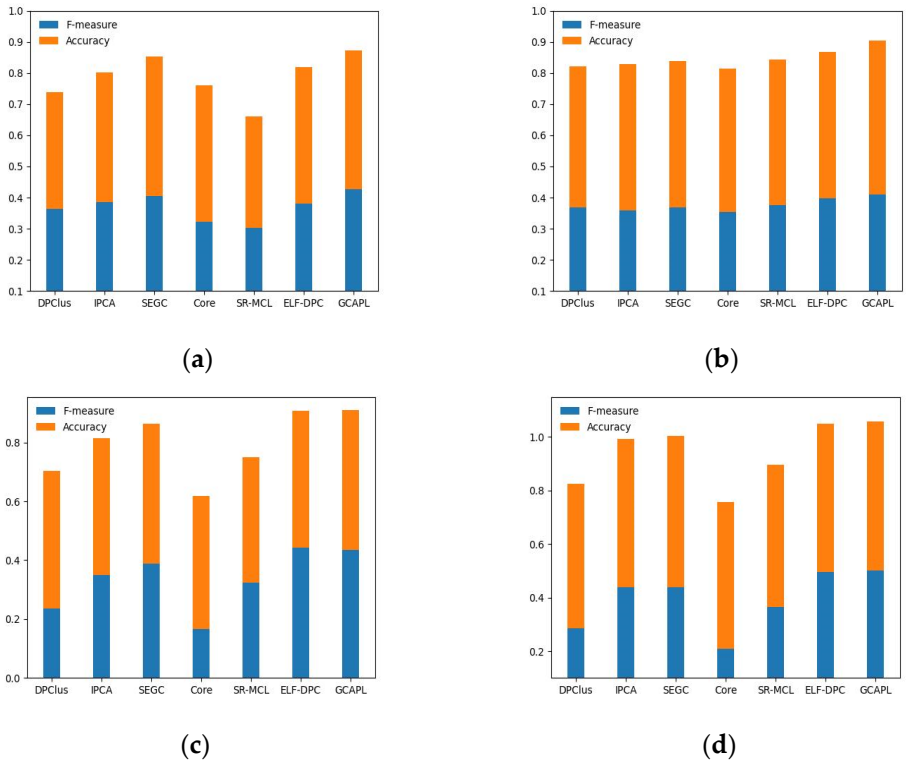


Figure 5. MIPS as benchmarks: Evaluation results by different algorithms on (**a**) Gavin02; (**b**) Gavin06; (**c**) K-extend; (**d**) BioGRID.

In summary, the GCAPL algorithm has good performance in detecting protein complexes. The GCAPL algorithm uses not only micro-topological structure metrics but also the macro-topological structure characteristic of the power-law distribution about clusters, and it can obtain better results in complex detection. The GCAPL algorithm further explores the relationship between network topological characteristics and functional modules in PPI networks, which is of great significance for improving the accuracy of protein complex detection.

4.6. Examples of Predicted Complexes

In this subsection, four predicted protein complexes with different sizes detected by the GCAPL algorithm are exhibited, and their corresponding network topology structures are shown in Figure 6. The predicted complex in Figure 6a is a fully interconnected network. Figure 6b shows a cluster that has a dense sub-graph with a relatively sparse periphery. Figure 6c,d show two clusters that are dense sub-graphs. Table 4 presents the Gene Ontology annotations of these predicted protein complexes in three aspects of biological processes, molecular functions, and cell components with corresponding significance *p*-values. The obtained *p*-values are notably small, indicating that these clusters have significant biological significance. The effectiveness of the GCAPL algorithm is demonstrated in its ability to identify protein complexes with multiple network structures.

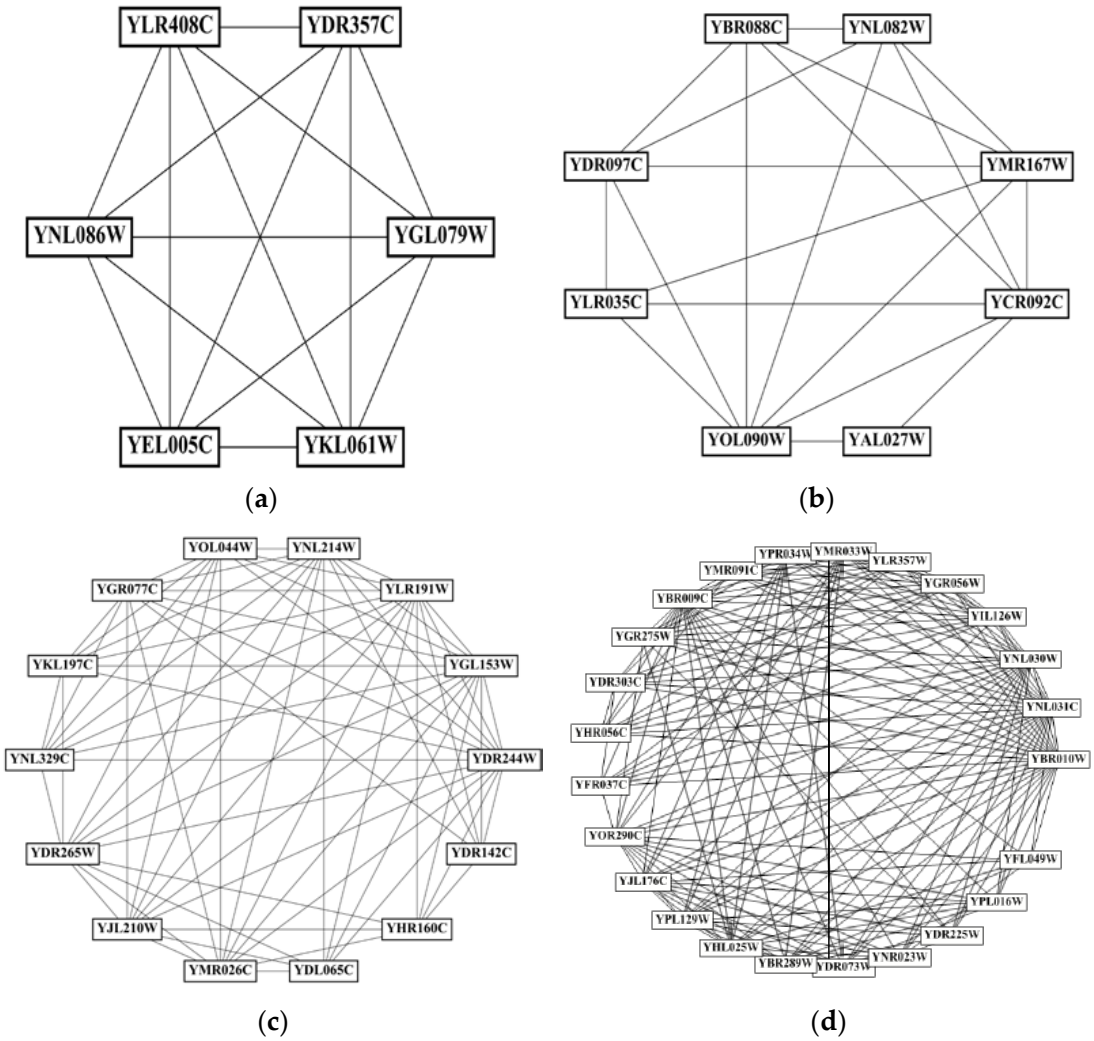


Figure 6. Examples of predicted protein complexes: (**a**) cluster a; (**b**) cluster b; (**c**) cluster c; (**d**) cluster d.

Table 4. Gene ontology annotations of the four predicted protein complexes.

ID	Processes		Functions		Components	
	Gene Ontology Term	***p*-Value**	**Gene Ontology Term**	***p*-Value**	**Gene Ontology Term**	***p*-Value**
a	endosome organization (GO:0007032)	4.04×10^{-15}	molecular function (GO:0003674)	0.00194	BLOC complex (GO:0031082)	1.06×10^{-19}
b	DNA repair (GO:0006281)	1.37×10^{-9}	DNA binding (GO:0003677)	9.64×10^{-8}	nucleus (GO:0005634)	0.00506
c	protein targeting to peroxisome (GO:0006625)	1.14×10^{-35}	Binding (GO:0005488)	0.00206	microbody part (GO:0044438)	3.31×10^{-29}
d	DNA-templated transcription (GO:0006351)	2.39×10^{-22}	DNA binding (GO:0003677)	6.11×10^{-7}	nuclear chromosome part (GO:0044454)	1.59×10^{-36}

5. Conclusions

Detecting protein complexes is of great significance for understanding biological mechanisms. This paper proposes a network clustering algorithm fused with power-law distribution for protein complex detection. The algorithm begins by calculating node weights, taking into account micro-topological structure metrics. Subsequently, the algorithm selects the non-clustered nodes with the higher weights as seeds and forms initial clusters around the seeds. Next, the algorithm greedily adds candidate nodes into the initial clusters based on the characteristics of scale-free power-law distribution to generate candidate clusters. A power-law distribution function, based on the macro-topological structure feature of power-law distribution about cluster size and number, is established to guide the cluster generation process. The power-law distribution function is employed to determine whether a candidate cluster qualifies as a final cluster. Compared with other algorithms, the *F-measure* + *Accuracy* of GCAPL improves by an average of 12.23% and 10.97% on the CYC2008 and MIPS benchmarks, respectively. The experimental analysis reveals that the proposed algorithm exhibits distinct advantages over other approaches.

The GCAPL algorithm mainly considers the biological network whose community size conforms to the power-law distribution characteristics. The algorithm does not take into account other distribution characteristics of the community size and fully considers the preferential attachment. The above information may further improve the performance of our algorithm to detect protein complexes. In addition, in real PPI networks, the connections between nodes are subject to constant changes, leading to variations in network topological structures. To mine functional modules in dynamic PPI networks, our future work will also focus on constructing dynamic networks and developing dynamic protein complex identification methods.

Author Contributions: Conceptualizing the algorithm, designing the method and revising the draft, J.W.; implementation of the computer code and writing the original draft, Y.J.; revising the manuscript, A.K.S.; visualizing and curating data, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was funded by the National Natural Science Foundation of China (No. 62006145); the Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi, China (No. 2020L0245); the Youth Science Foundation of Shanxi University of Finance and Economics, China (No. QN-202016); and Shandong Provincial Natural Science Foundation, China (No. ZR2020MF146).

Data Availability Statement: The datasets used in this study are publicly available and downloaded from the BioGRID database (https://downloads.thebiogrid.org/BioGRID, accessed on 1 March 2023), MIPS database (http://mips.gsf.de, accessed on 8 September 2019), and CYC2008 complexes database (http://wodaklab.org/cyc2008/, accessed on 12 April 2023).

Acknowledgments: This study received support from the Teaching and Research Department of Computer Science and Technology, Shanxi University of Finance and Economics, and all authors would like to express their gratitude for this.

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

References

1. Wu, L.; Huang, S.; Wu, F.; Jiang, Q.; Yao, S.; Jin, X. Protein Subnuclear Localization Based on Radius-SMOTE and Kernel Linear Discriminant Analysis Combined with Random Forest. *Electronics* **2020**, *9*, 1566. [CrossRef]
2. Ito, T.; Chiba, T.; Ozawa, R.; Yoshida, M.; Hattori, M.; Sakaki, Y. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci. USA* **2001**, *98*, 4569–4574. [CrossRef] [PubMed]
3. Causier, B.; Davies, B. Analysing protein-protein interactions with the yeast two-hybrid system. *Plant Mol. Biol.* **2002**, *50*, 855–870. [CrossRef] [PubMed]
4. Puig, O.; Caspary, F.; Rigaut, G.; Rutz, B.; Bouveret, E.; Bragado-Nilsson, E.; Wilm, M.; Séraphin, B. The tandem affinity purification (TAP) method: A general procedure of protein complex purification. *Methods* **2001**, *24*, 218–229. [CrossRef] [PubMed]
5. Rahiminejad, S.; Maurya, M.R.; Subramaniam, S. Topological and functional comparison of community detection algorithms in biological networks. *BMC Bioinform.* **2019**, *20*, 212. [CrossRef]
6. Spirin, V.; Mirny, L.A. Protein complexes and functional modules in molecular networks. *Proc. Natl. Acad. Sci. USA* **2003**, *100*, 12123–12128. [CrossRef]
7. Bai, L.; Cheng, X.; Liang, J.; Guo, Y. Fast graph clustering with a new description model for community detection. *Inf. Sci.* **2017**, *388–389*, 37–47. [CrossRef]
8. Saxena, A.; Prasad, M.; Gupta, A.; Bharill, N.; Patel, O.P.; Tiwari, A.; Er, M.J.; Ding, W.; Lin, C.-T. A review of clustering techniques and developments. *Neurocomputing* **2017**, *267*, 664–681. [CrossRef]
9. Emmons, S.; Kobourov, S.; Gallant, M.; Börner, K. Analysis of network clustering algorithms and cluster quality metrics at scale. *PLoS ONE* **2016**, *11*, e0159161. [CrossRef]
10. Bhowmick, S.S.; Seah, B.S. Clustering and summarizing protein-protein interaction networks: A survey. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 638–658. [CrossRef]
11. Pan, Y.; Guan, J.; Yao, H.; Shi, Y.; Zhou, Y. Computational methods for protein complex prediction: A survey. *J. Front. Comput. Sci. Technol.* **2022**, *16*, 1–20.
12. Manipur, I.; Giordano, M.; Piccirillo, M.; Parashuraman, S.; Maddalena, L. Community Detection in Protein-Protein Interaction Networks and Applications. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2021**, *20*, 217–237. [CrossRef]
13. Liu, G.; Wong, L.; Chua, H.N. Complex discovery from weighted PPI networks. *Bioinformatics* **2009**, *25*, 1891–1897. [CrossRef]
14. Bader, G.D.; Hogue, C.W.V. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinform.* **2003**, *4*, 2. [CrossRef]
15. Palla, G.; Derényi, I.; Farkas, I.; Vicsek, T. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **2005**, *435*, 814–818. [CrossRef] [PubMed]
16. Amin, A.U.; Shinbo, Y.; Mihara, K.; Kurokawa, K.; Kanaya, S. Development and implementation of an algorithm for detection of protein complexes in large interaction networks. *BMC Bioinform.* **2006**, *7*, 207. [CrossRef]
17. Li, M.; Chen, J.-E.; Wang, J.-X.; Hu, B.; Chen, G. Modifying the DPClus algorithm for identifying protein complexes based on new topological structures. *BMC Bioinform.* **2008**, *9*, 398. [CrossRef] [PubMed]
18. Wang, J.; Zheng, W.; Qian, Y.; Liang, J. A seed expansion graph clustering method for protein complexes detection in protein interaction networks. *Molecules* **2017**, *22*, 2179. [CrossRef]
19. Leung, H.C.; Xiang, Q.; Yiu, S.M.; Chin, F.Y. Predicting protein complexes from PPI data: A core-attachment approach. *J. Comput. Biol.* **2009**, *16*, 133–144. [CrossRef] [PubMed]
20. Yue, L.; Jun, X.; Sihang, Z.; Siwei, W.; Xifeng, G.; Xihong, Y.; Ke, L.; Wenxuan, T.; Wang, L.X. A survey of deep graph clustering: Taxonomy, challenge, and application. *arXiv* **2022**, arXiv:2211.12875.
21. Sun, H.; He, F.; Huang, J.; Sun, Y.; Li, Y.; Wang, C.; He, L.; Sun, Z.; Jia, X. Network embedding for community detection in attributed networks. *ACM Trans. Knowl. Discov. Data* **2020**, *14*, 1–25. [CrossRef]
22. Kumar, S.; Panda, B.S.; Aggarwal, D. Community detection in complex networks using network embedding and gravitational search algorithm. *J. Intell. Inf. Syst.* **2021**, *57*, 51–72. [CrossRef]
23. Wang, R.; Ma, H.; Wang, C. An ensemble learning framework for detecting protein complexes from PPI networks. *Front. Genet.* **2022**, *13*, 839949. [CrossRef]
24. Liu, X.; Yang, Z.; Zhou, Z.; Sun, Y.; Lin, H.; Wang, J.; Xu, B. The impact of protein interaction networks' characteristics on computational complex de-tection methods. *J. Theor. Biol.* **2018**, *439*, 141–151. [CrossRef]

25. Cherifi, H.; Palla, G.; Szymanski, B.K.; Lu, X. On community structure in complex networks: Challenges and opportunities. *Appl. Netw. Sci.* **2019**, *4*, 117. [CrossRef]
26. Huang, Z.; Zhong, X.; Wang, Q.; Gong, M.; Ma, X. Detecting community in attributed networks by dynamically exploring node attributes and topological structure. *Knowl.-Based Syst.* **2020**, *196*, 105760. [CrossRef]
27. Ghalmane, Z.; Cherifi, C.; Cherifi, H.; El Hassouni, M. Centrality in complex networks with overlapping community structure. *Sci. Rep.* **2019**, *9*, 10133. [CrossRef]
28. Rajeh, S.; Savonnet, M.; Leclercq, E.; Cherifi, H. Characterizing the interactions between classical and community-aware centrality measures in complex networks. *Sci. Rep.* **2021**, *11*, 10088. [CrossRef] [PubMed]
29. Girvan, M.; Newman, M.E.J. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826. [CrossRef]
30. Sangaiah, A.K.; Rezaei, S.; Javadpour, A.; Zhang, W. Explainable AI in big data intelligence of community detection for digitalization e-healthcare services. *Appl. Soft Comput.* **2023**, *136*, 110119. [CrossRef]
31. Ma, J.; Fan, J. Local optimization for clique-based overlapping community detection in complex networks. *IEEE Access* **2019**, *8*, 5091–5103. [CrossRef]
32. Kustudic, M.; Xue, B.; Zhong, H.; Tan, L.; Niu, B. Identifying Communication Topologies on Twitter. *Electronics* **2021**, *10*, 2151. [CrossRef]
33. Watts, D.J.; Strogatz, S.H. Collective dynamics of 'small-world' networks. *Nature* **1998**, *393*, 440–442. [CrossRef]
34. Gavin, A.C.; Bösche, M.; Krause, R.; Grandi, P.; Marzioch, M.; Bauer, A.; Schultz, J.; Rick, J.M.; Michon, A.M.; Cruciat, C.M.; et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* **2002**, *415*, 141–147. [CrossRef] [PubMed]
35. Gavin, A.-C.; Aloy, P.; Grandi, P.; Krause, R.; Boesche, M.; Marzioch, M.; Rau, C.; Jensen, L.J.; Bastuck, S.; Dümpelfeld, B.; et al. Proteome survey reveals modularity of the yeast cell machinery. *Nature* **2006**, *440*, 631–636. [CrossRef]
36. Krogan, N.J.; Cagney, G.; Yu, H.; Zhong, G.; Guo, X.; Ignatchenko, A.; Li, J.; Pu, S.; Datta, N.; Tikuisis, A.P.; et al. Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. *Nature* **2006**, *440*, 637–643. [CrossRef]
37. Stark, C.; Breitkreutz, B.J.; Reguly, T.; Boucher, L.; Breitkreutz, A.; Tyers, M. BioGRID: A general repository for interaction datasets. *Nucleic Acids Res.* **2006**, *34* (Suppl. S1), D535–D539. [CrossRef]
38. Pu, S.; Wong, J.; Turner, B.; Cho, E.; Wodak, S.J. Up-to-date catalogues of yeast protein complexes. *Nucleic Acids Res.* **2009**, *37*, 825–831. [CrossRef] [PubMed]
39. Brohée, S.; Van Helden, J. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinform.* **2006**, *7*, 488. [CrossRef] [PubMed]
40. Li, X.; Wu, M.; Kwoh, C.-K.; Ng, S.-K. Computational approaches for detecting protein complexes from protein interaction networks: A survey. *BMC Genom.* **2010**, *11*, S3. [CrossRef] [PubMed]
41. Ma, X.; Gao, L. Predicting protein complexes in protein interaction networks using a core-attachment algorithm based on graph communicability. *Inf. Sci.* **2012**, *189*, 233–254. [CrossRef]