Neural Computing and Applications
https://doi.org/10.1007/s00521-018-3508-z

S.I. : EMERGING INTELLIGENT ALGORITHMS FOR EDGE-OF-THINGS COMPUTING

Module overlapping structure detection in PPI using an improved link similarity-based Markov clustering algorithm

L. Gu[1] · Y. Han[1] · C. Wang[1] · Wei Chen[1] · Jun Jiao[1] · X. Yuan[2]

Received: 8 January 2018 / Accepted: 24 April 2018

Abstract

The identification and analysis of functional modules in protein–protein interaction (PPI) networks provide insight into understanding the organization and function of biological systems. A lot of overlapping structures are shared by the functional modules in PPI networks, which indicates there are some proteins play indispensable roles in different biological processes. Markov clustering (MCL) is a popular algorithm for clustering networks in bioinformatics. In this paper, to identify the overlapping structures among the functional modules and find more modules with biological significance in PPI networks, we propose a Markov clustering algorithm based on link similarity (MLS). First of all, the weighted link similarity is calculated and the link similarity matrix which measures the association strength of the protein interactions can be gotten. Then, the link similarity matrix is divided by applying Markov clustering, and the clustering results are mapped to original networks to analyze the protein modules. The method has been experimented on three databases, including DIP, Gavin and Krogan. Our results show that the MLS cannot only accurately identify the functional modules, but also outperform the original MCL algorithm and the F-measure value improved 5–10% compared with it.

Keywords Markov clustering · Link similarity · Functional modules · PPI networks

1 Introduction

PPI networks, as one of the major research fields of bioinformatics, have gradually become a focus recently. At present, how to effectively and accurately identify the protein complexes and the functional modules is the main direction of PPI networks. Modules reflect biological significance, within which the protein connections are always denser than those among other modules, but between which they are sparser. Therefore, the acquisition and analysis of module structures is the key to identifying protein complexes and functional modules in a PPI network. We can detect biological protein families based on graph theory.

Traditional clustering algorithms of PPI networks can be classified into the density-based method, hierarchical clustering method, partition-based method and flow simulation method, etc. MCODE method, as one of the density-based methods, was proposed by Bader et al. [1], could effectively detect substructures with high density, but lose much nodes information in a graph with less relevance. Although the algorithm based on hierarchical clustering, such as the GN algorithm [2], can reliably and sensitively extract community structure from artificially generated networks, it discards some functional modules with overlapping structures because of its results showed in form of the dendrogram. The partition-based method, such as the RNSC algorithm, which was proposed by King et al. [3], can be used to make biological experiments more efficient and less expensive. But it could not determine the number

✉ L. Gu
glc@ahau.edu.cn

Y. Han
15720920@ahau.edu.cn

C. Wang
wangchao_ICLE@ahau.edu.cn

Jun Jiao
jiaojun2000@ahau.edu.cn

X. Yuan
xiaohui.yuan@unt.edu

[1] School of Computer and Information, Anhui Agricultural University, No.130 ChangJiang Road, Hefei 230036, Anhui, China

[2] Department of Computer Science and Engineering, University of North Texas, Denton, TX, USA

Published online: 19 May 2018

of initial clustering which had a great effect on the final conclusion, so that the clustering results may be less definitive. MCL algorithm, a representative method based on flow simulation, was proposed by Enright et al. [4], which was applied to detect protein functional modules in PPI networks. It performs "inflate" and "expand" repeatedly to change the transition probabilities for discovering clusters and requires no knowledge of the number of the initial clusters. Most of these modules detection methods limit that the node belongs to one module. Besides, Yuan et al. proposed a method to find biomarkers by extracting proteins network from the full text of online articles [5]. However, many real networks are composed of highly overlapping nodes. And for the reason of biology, some proteins of PPI network perform complex biological functions in multiple modules.

To identify the overlaps among modules, a lot of different methods based on nodes information have been submitted. Nepusz et al. [6] proposed a method for identifying overlapping protein complexes from PPI networks named ClusterONE algorithm (clustering with overlapping neighborhood expansion) which derived complexes with better relevance. However, it only relies on the cohesive force formula, and there may be deviations in the algorithm. For example, there may be nodes that reduce cohesion, but in fact, the nodes do belong to candidate protein complexes. Brohee et al. [7] compared the performance for identifying modules in the PPI network among the MCL, RNSC clustering algorithm and others, and pointed out that the MCL algorithm had outperformed others. However, the widespread use of the algorithm was hindered by its lack of scalability. Regularized MCL algorithm (R-MCL) was proposed by Venu Satuluri et al. [8], it not only improved the accuracy of traditional MCL but also redressed the weakness of output fragmentation. Because R-MCL algorithm only derived non-overlapping classes and cannot match overlapping proteins among modules, Shih et al. [9] proposed the soft Markov algorithm (SR-MCL) to detect highly overlapping and hierarchical modules of PPI networks based on the R-MCL algorithm. The value of F-measure in the method was significantly higher than the value of it in the R-MCL algorithm, the main reason was that the value of recall increased faster than precision.

Different from the methods mentioned above, Ahn et al. [10] first proposed an algorithm whose input network consists of links rather than nodes, which means edges are clustered. And Fortunato has introduced almost all methods on clustering edges completely [11]. A Markov clustering algorithm based on link clustering (MLC) was proposed by Wang et al. [12] to detect the overlaps of modules which exist in PPI network, but this algorithm is not tested in complex datasets.

Considering the data characteristic of PPI network, a Markov clustering algorithm based on link similarity (MLS) is proposed in this paper to divide the PPI networks and find the functional modules with overlapping and non-overlapping structures. MLS includes three steps. Firstly, calculate the edges' similarity. Secondly, derive the weighted links similarity matrix. Thirdly, use the Markov clustering to partition the matrix in order to identify functional modules. Evaluation results on the DIP, GAVIN and KROGAN datasets show that the proposed MLS algorithm can effectively identify overlapping functional modules and outperform most of the similar methods.

2 Terminology

2.1 Link similarity

Link networks. We denote e_{ik} as the link that connects the nodes i and k. The link similarity (LS) [13] between the pair links e_{ik} and e_{jk} sharing the same node k is measured following the Jaccard index [10]:

$$\mathrm{LS}\left(e_{ik}, e_{jk}\right) = \frac{|n_+(i) \cap n_+(j)|}{|n_+(i) \cap n_+(j)|} \tag{1}$$

where $n_+(i)$ is denoted as the set of nodes, which consists of the node i and its cs. Then, cutting some links at predefined threshold builds a link similarity matrix.

2.2 Networks with weighted links

To extend the application field of the similarity between links, we introduce the Tanimoto coefficient [14] into the Jaccard index when it is used in networks with weighted links (without self-loops). Consider a vector $\mathbf{a}_i = \left(\tilde{A}_{i1}, \ldots, \tilde{A}_{iN}\right)$ with

$$\tilde{A}_{ij} = \frac{1}{k_i} \sum_{i' \in n(i)} w_{ii'} \delta_{ij} + w_{ij} \tag{2}$$

where w_{ij} represents the weight on edge e_{ij}, $n(i) = \left\{j | w_{ij} > 0\right\}$ represents the set of all neighbors of node i, $k_i = |n(i)|$, if $i = j$, $\delta_{ij} = 1$, otherwise $\delta_{ij} = 0$. Then, the similarity between edges e_{ik} and e_{jk} similar to Eq. (1) is now:

$$WLS\left(e_{ik}, e_{jk}\right) = \frac{\mathbf{a}_i \cdot \mathbf{a}_j}{|\mathbf{a}_i|^2 + |\mathbf{a}_j|^2 - \mathbf{a}_i \cdot \mathbf{a}_j} \tag{3}$$

3 Methodology

3.1 Markov clustering (MCL)

In MCL algorithm, the two processes of expansion and inflation are alternated between repeatedly of the stochastic matrix M. The aim of expansion operator is to make flow to connect different regions of the graph. The aim of inflation operator is to both strengthen and weaken of current. The matrix M of transition probabilities changes constantly until it has been convergent. The matrix M is defined as follows:

$$M(i,j) = \frac{W(i,j)}{\sum_{k=1}^{n} W(k,j)^{r}} \tag{4}$$

where W is the adjacent matrix with self-loops.

(1) Expand: Input M and the value of e, output $M_{\exp}$

$$M_{\exp} = Expand(M) = M^{e} \tag{5}$$

(2) Inflate: Input M and the value of r, output $M_{\inf}$

$$M_{\inf}(i,j) = \frac{M(i,j)^{r}}{\sum_{k=1}^{n} M(k,j)^{r}} \tag{6}$$

Repeat steps (1) and (2) until the matrix M is convergent.

Interpret resulting matrix to find clusters.

For the derived convergent matrix M, the vertices are split into attractors and vertices that are being attracted by the attractors. The column jth has only one nonzero value, whose line ith represents the attractor of node v_j. Attractors and the elements they attract, which are similar to node v_j, are swept together into the same cluster. MCL algorithm of pseudocode is shown in Algorithm 1.

```
Algorithm 1 MCL
A : = A + I   // Add self-loops to each node
M : = AD^(-1) // Create the stochastic transition matrix
    Do
        M : = $M_{exp}$ (M) = Expand (M)
        M : = $M_{inf}$ (M) = Inflate (M)
    Until M converges
Interpret M to discover clusters
```

3.2 Link similarity-based Markov clustering (MLS)

Although MCL produces good clustering results, it cannot find overlapping clusters. And it usually merges functional modules sharing the same (bridge) node(s). As shown in Fig. 1, MCL only produces two clusters (1, 2, 3, 4, 5, 6) and (7, 8, 9). In order to overcome the issue, we have improved the clustering process of detecting the protein modules based on the extended weighted link similarity definition as formula (3) and MCL. We use the links similarity rather than nodes interactions of PPI networks in calculating the link similarity adjacent matrix, which is applied to the MCL clustering process. Moreover, if the bridge node is likely to be the attractor in MCL, at the same time, by processing the link similarity adjacent matrix, we can correctly produce clusters sharing the same bridge node. For example, in Fig. 1, MCL only identifies two clusters, but in MLS, the attractor node 4 is included in the edge(4–5), edge(3–4) and other edges, after processed, so MLS could produce three clusters matching the three modules, (1, 2, 3, 4), (4, 5, 6) and (7, 8, 9).

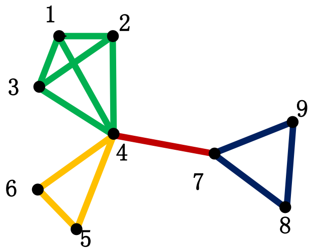

Fig. 1 An example pointing out the problem of MCL. All edges have the weight 1

The procedure of Markov clustering based on link similarity (MLS) is shown in Algorithm 1. Firstly, use the WLS formula (3) to calculate all the link similarity between the edges that exist in the network. Secondly, create the associated matrix M with the calculated link similarity. Finally, apply the Markov clustering method to process the similarity matrix M, and interpret the resulting matrix to map the nodes in the community of the original network.

As a traditional clustering algorithm, LC method is a greedy algorithm. That is to say, a link may merge in a local optimal result. On the other hand, when using the function D to cut the link dendrogram, because of the calculation of D, the computational complexity is always O (n^2). And then a lot of small clusters are split out in the network.

The difference between MLS and LC, which is guaranteed has the advantages for identification of overlapping structures, is that the former employs Markov clustering (MCL) on the similarity matrix clustering, and the latter uses HC and PD. LC may be trapped in a local optimal and generates fragmentation of output, which is redressed by MLS algorithm. Its core idea is the simulation of random walk on a graph based on the visiting vertex in the graph. The process may not end until most of its vertexes in the subgraph have been visited [15]. Moreover, it is not necessary for MCL method to predefine number of clusters and need less parameters compared with LC.

4 Results and discussion

4.1 Datasets

In this paper, we use three PPI networks of *Saccharomyces cerevisiae* extracted from DIP [16], Gavin [17] and Kragon [18], the details of these three networks are shown in Table 1, to test algorithm's improvement effect, because the yeast PPI network is the most complete and reliable in all species. In order to evaluate the derived sets of functional modules from the MLS algorithm, we put the protein complexes datasets from MIPS datasets and CYC2008 [19] datasets as standard. Because the CYC2008 is known as functional module sets, which contains 408 functional modules obtained by biological method, we can use it to evaluate the quality of the predicted results. The protein functional annotation table is chosen from funcat-2.1_-data_20070316 (ftp://ftpmips.helmholtz-muenchen.de/fungi/Saccharomycetes/CYGD/catalogues/funcat/funcat2.1_data_20070316). Experimental environment is a Windows 7 64-bit PCS, processor type is Intel i7-2600, 3.40 GHz CPU, 4 GB of memory, with python2.7 as a development environment.

4.2 Metrics

The MLS algorithm has run in the above three datasets, and at the same time been compared with link clustering algorithm and MCL algorithm. Then, several common modularity indexes of evaluating the overlapping structures, such as EQ, coverage rate (CR), and some common algorithm performance indicators, such as S_n, S_p, *F-score*, the *Precision*, *Recall*, *F-measure*, are used to compare the performance of those three algorithms used in these three datasets.

Table 1 Information of the three yeast networks used in the experiment

Name	|V|	|E|	Average degree of vertices
DIP	4936	17,203	6.98
Krogan	2675	7080	5.29
Gavin	1430	6531	9.13

4.2.1 EQ

A modularity measure Q was defined by Newman et al. [20] to be used in evaluating the degree of modularization in networks. Shen et al. [21] proposed an extended modularity function EQ to evaluate the goodness of overlapped community decomposition. EQ_l, which represents a single community, is denoted as follows:

$$EQ_l = \frac{2}{|M|} \sum_{i \in H_l, j \in H_l} \frac{1}{O_i O_j} \left(A_{ij} - \frac{n_i n_j}{2|M|} \right) \tag{7}$$

Here, H_l is the set of nodes after the network being divided into k communities. M is the set of links in the network. $|M|$ is the total number of links in the network. O_i is the number of communities that node i belongs to. If there is a link between nodes i and j, A_{ij} = 1, otherwise A_{ij}= 0. n_i is the degree of i. The *EQ* of the whole networks is defined as:

$$EQ = \sum_{l=1}^{k} EQ_l \tag{8}$$

A higher value of *EQ* indicates networks with stronger community structure. Thus, if the value of *EQ* is 0, all the nodes belong to the same cluster.

4.2.2 Coverage rate (CR)

Coverage Rate [22, 23] is used in estimating how many proteins exit in real interaction module. When a real module set and a predicted module set of a network are given, the coverage rate is defined as follows:

$$Coverage\ rate = \frac{\sum Max\{P_{ij}\}}{\sum N_i} \tag{9}$$

Here, P_{ij} is the number of nodes which are shared proteins by the $\underline{i\text{th}}$ real module and the $\underline{j\text{th}}$ predicted module of a network. N_i is the number of proteins in the $\underline{i\text{th}}$ real interaction module.

4.2.3 Precision, Recall, F-measure, Sp, Sn, F-score

Precision [22] is the ratio of the number of nodes in correct clustering to the nodes in the experimental results. *Recall* [24] is the ratio of the number of correct clustering nodes to the nodes in the standard database. *F-measure* [25] is the harmonic mean of *Precision* and *Recall*. Its calculation formula is shown as follows:

$$precision(c_i, s_j) = \frac{|c_i \cap s_j|}{|c_i|} \tag{10}$$

$$recall(c_i, s_j) = \frac{|c_i \cap s_j|}{|s_j|} \tag{11}$$

$$F\text{-}measure = \frac{2 * precision * recall}{precision + recall} \tag{12}$$

Here, $C = \{C_1, C_2, \ldots, C_i, \ldots C_k\}$ represents clustering results, C_i is a cluster in it, S_j is the standard cluster in the database. The larger the number of nodes in clustering results is, the higher the *Recall* value is. The smaller the number of nodes in clustering results is, the higher the *Precision* value is. Thus, that is not reasonable enough to use precision and recall only in evaluating the clustering results, while *F-measure* can be applied to evaluate the performance of the clustering results derived from the algorithm and the biological significance of functional modules comprehensively.

Due to nodes' dispersion of clustering results, the value of *Precision* and *Recall* will be affected. As a result, the performance of clustering results is not very clear. In this paper, we not only calculate those indexes mentioned above but also supplement the experiment measure indexes with S_p, S_n [26] and *F*-score for further evaluation.

S_p is the ratio of the number of nodes which is more than predefined threshold shared by module c_i mined by algorithm and standard clustering module s_j to the total number of nodes. S_n is the ratio of the number of nodes which is more than predefined threshold shared by standard clustering module s_i and module c_j mined by algorithm to the total number of nodes. *F-score* [27] is the harmonic mean of S_p and S_n. It is calculated as follows:

$$Sp = \frac{|\{c_i | c_i \in C, \exists s_j \in S, Overlap\_score(c_i, s_j) \geq \alpha\}|}{|C|} \tag{13}$$

$$Sn = \frac{|\{s_i | s_i \in S, \exists c_j \in C, Overlap\_score(s_i, c_j) \geq \alpha\}|}{|S|} \tag{14}$$

$$F\text{-}score = \frac{2 * Sp * Sn}{Sp + Sn} \tag{15}$$

Here, α is the threshold, the value is generally set as 0.2, $Overlap\_score(c_i, s_j)$ [28] represents the similarity between module c_i mined by the algorithm and the standard clustering module. Its calculation formula is shown as follows:

$$Overlap\_score(c_i, s_j) = \frac{|c_i \cap s_j|^2}{|c_i| \times |s_j|} \tag{16}$$

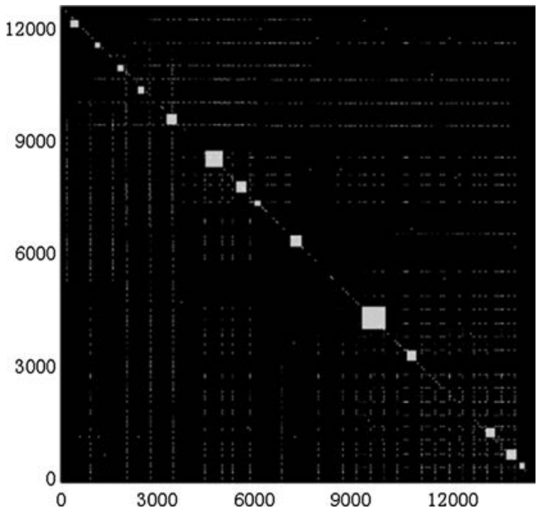

Fig. 2 Link similarity matrix of PPI network from DIP database

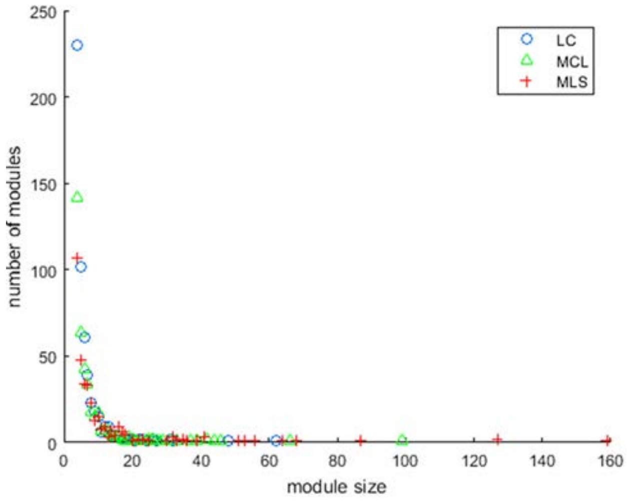

Fig. 3 MLS result on DIP. It represents the size distribution of DIP PPI modules obtained by MLS. The x-axis indicates the size of modules, i.e., the number of proteins in each module. The y-axis shows the number of modules with the size corresponding to the x-axis

4.3 Performance on DIP datasets

After deriving the expansion link similarity matrix from the DIP database, we process it with the MLS method. Its WLS similarity matrix and clustering results are shown in Fig. 2 and Fig. (3-1). As shown in Fig. (3-1), there are totally 396 modules and most of these modules contain less than 10 proteins, rarely more than 10 proteins are contained in modules. This is consistent to networks with a mainly scale-free degree distribution and small-world [29] phenomenon. Namely, most of the proteins interact with a few other proteins, while a few proteins interact with many other proteins.

Table 2 Comparison with three methods on DIP network by different evaluations

	EQ	*CR*	*Sp*	*Sn*	*F-score*	*Precision*	*Recall*	*F-measure*	Time/s
MLS	0.5712	0.5569	0.6013	0.3426	0.4365	0.5696	0.3090	0.4007	125
MCL	0.2603	0.5204	0.6420	0.2103	0.3168	0.5883	0.1957	0.2937	71
LC	0.2497	0.3386	0.3151	0.4667	0.3762	0.4022	0.2061	0.2725	19

Table 3 MLS Modular example 1 in DIP database

Modular proteins	GO functions	*P* value
Module 1:	Biological process	2.17e−29
YOR076C, YDR293C, YDL111C, YHR081W, YMR287CYOL021C, YGR158C, YCR035C, YLR398C, YOR001W, YDR280W, YOL142W, YGR195W, YNL232W, YGR095C, YHR069C, YMR093W, YLR129W, YHR169W, YLL011W, YJL050W, YMR229C, YLR222C, YMR128W, YGL171W, YNL299W, YDL175C, YJL109C, YLR409C	GO: 0016078 ∼ tRNA catabolic process	
	Cellular component	2.50e−25
	GO: 0000178 ∼ exosome (RNase complex)	
	Molecular functions	7.52e−18
	GO: 0003723 ∼ RNA binding	
Module 2:	Biological process	3.73e−27
YOR076C, YDR293C, YDL111C, YHR081W, YMR287C, YOL021C, YGR158C, YCR035C, YLR398C, ***YOR001W****, YDR280W, YOL142W, YGR195W, YNL232W, YGR095C, YHR069C, YMR093W, YHR169W,* ***YJL050W***	GO: 0071051 ∼ polyadenylation-dependent snoRNA 3′-end processing	
	Cellular component	4.77e−29
	GO: 0000178 ∼ exosome (RNase complex)	
	Molecular functions	2.13e−13
	GO: 0003723 ∼ RNA binding	
Module 3:	Biological process	4.13e−16
YJL050W*,* ***YOR001W****, YMR229C, YLR129W, YLR222C, YMR128W, YGL171W, YNL299W, YDL175C, YJL109C, YLR409C, YLL011W*	GO: 0016072 ∼ rRNA metabolic process	
	Cellular component	5.41e−16
	GO: 0005730 ∼ nucleolus	
	Molecular functions	2.32e−07
	GO: 0003723 ∼ RNA binding	

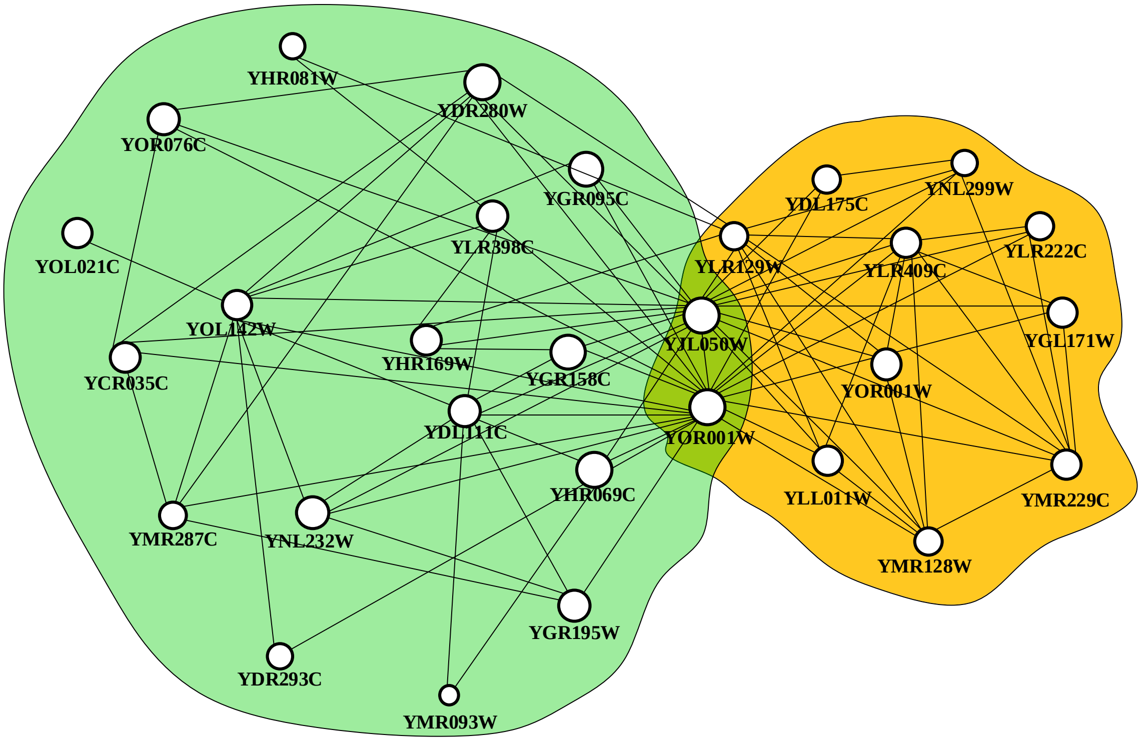

Fig. 4 Modular 1, 2 and 3 with two overlapping proteins

Table 4 Comparison with three methods on Krogan and Gavin by different evaluations

Dataset	Method	*EQ*	*CR*	*Sp*	*Sn*	*F-score*	*Precision*	*Recall*	*F-measure*	Time/s
Krogan	MLS	0.5502	0.5337	0.5023	0.6204	0.5551	0.5574	0.1374	0.2205	112
	MCL	0.2235	0.4729	0.5719	0.4025	0.4725	0.6310	0.0943	0.1640	60
	LC	0.1796	0.4174	0.6698	0.2777	0.3926	0.5272	0.1040	0.1748	12
Gavin	MLS	0.3073	0.4354	0.7835	0.4373	0.5659	0.6235	0.4873	0.5470	94
	MCL	0.2661	0.3092	0.8211	0.3374	0.4782	0.7188	0.3279	0.4504	53
	LC	0.1865	0.1750	0.5479	0.4418	0.4892	0.8593	0.1346	0.2327	10

As shown in Table 2, the higher evaluation value of *EQ*/*CR*/*F*-score indicates that MLS outperforms LC and MCL algorithm in these tasks, which means that the MLS clustering results are closer to actual clusters, namely it has more functional modules with biological significance. And higher *EQ* score represents better community modularity. What is more, the lower *CR* value of LC and MCL methods reflects the lower coverage of original nodes in networks and most nodes information is lost. There are 396, 351 and 545 modules derived from DIP database after being processed by MLS, MCL, LC methods. The module size distribution is shown in Fig. 3. In order to evaluate the functional correlations of the clustering results, we analyze the instantiation of the results derived from the MLS and MCL.

The higher *EQ*/*F*-score value of MLS algorithm can be obtained, because the MCL method is used in processing link similarity matrix after calculating the extended WLS. The results show that the algorithm will divide the clustering results further obtained from originally MCL algorithm, because MLS is based on edges rather than nodes. So it can identify overlapping structures, the GO function for module 1, 2 and 3 is shown in Table 3, which illustrates modules based on biological process, molecular functions and cellular component of Omicsbean GO enrichment, the module 1 derived from MCL is decomposed into two modules derived from MLS with overlapping structure of module 2 and 3, there are 2 overlapping nodes RRP6 (YOR001W, Nuclear exosome exonuclease component, involved in RNA processing, maturation, surveillance, degradation, tethering and export), MTR4 (YJL050W, Cofactor for the exosome complex, involved in nuclear RNA processing and degradation both as a component of the TRAMP complex and in TRAMP independent processes, has a KOW domain that shows RNA binding activity), shown in Fig. 4.

In addition, 13 out of 29 genes in Modular 1 belong to same GO biological process category (GO: 0016078), 10 out of 29 genes belong to same GO biological process category (GO: 0016072) and 22 out of 29 genes belong to same GO molecular function (GO: 0003723), the two modules are denser. Stronger relevance between proteins is further subdivided into a module; then, more modular proteins with biological significance can be obtained. Likewise, 13 genes from Modular 2, including RRP6 and MTR4, are in the same GO biological process (GO: 0071051) and 12 out of 12 genes from Modular 3, including RRP6 and MTR4, are in the same GO biological process (GO: 0016072).

Our analysis indicates that RRP6 and MTR4 may be functionally involved in both Modular 2 and Modular 3 and we also found studies that supported this discovery. In Schuch's paper [30], Rrp6 and Mtr4 physically and functionally interacts with the 10-subunit core complex of the exosome(Exo-10), and they provide detailed structural insight into the interaction between the Rrp6 complex and Mtr4 that medicates an important link between Mtr4 and the core exosome.

But the MLS algorithm is based on link clustering. It is necessary to calculate the link similarity among links to get the link similarity matrix. The increase in link similarity

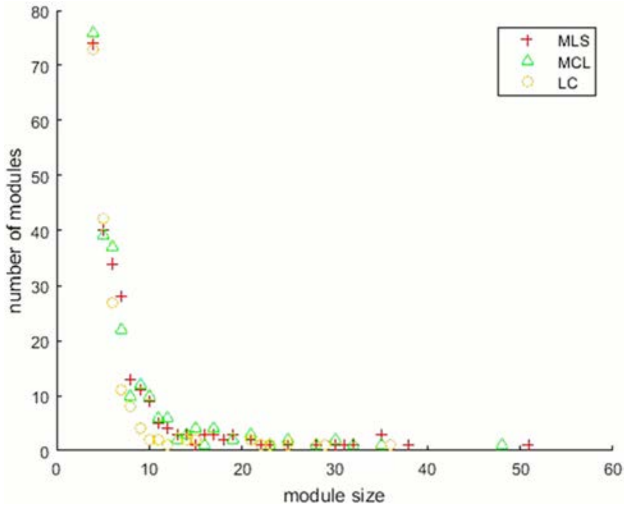

Fig. 5 MLS result on Krogan. It represents the size distribution of DIP PPI modules obtained by MLS. The x-axis indicates the size of modules, i.e., the number of proteins in each module. The y-axis shows the number of modules with the size corresponding to the x-axis

matrix processed in MCL clustering is always $O(n^2)$, and the classical MCL based on nodes is $O(n)$. And the MCL clustering of similarity matrix is a random walk process, it will not stop until the matrix becomes convergent, while the process of LC is looking for a local optimum. Thus, the computational complexity of MLS algorithm is higher than MCL and LC. However, considering the algorithm in cloud computing environment [31] may be a better approach to decrease its computational complexity.

4.4 Performance on Krogan and Gavin PPI networks

After getting the WLS similarity matrix of Krogan and Gavin, we applied them to the MLS method and output the clustering results, using the listed evaluation indexes above in analyzing the further performance of the algorithm. As shown in Table 4, the value of *EQ*/*CR*/*F*-measure/*F*-score indicates the MLS algorithm outperforms LC and MCL. The higher *F-score* value indicates that the MLS clustering results closer to the actual clusters, namely it has more functional modules with biological significance, and the higher EQ score represents better community modularity. The lower CR value of LC and MCL methods reflects the lower coverage of original nodes in networks, and most nodes information is lost. There are 362, 371 and 424 modules derived from Krogan after being processed by MLS, MCL, LC methods. And the module size distribution is shown in Fig. 5. In order to evaluate the functional relevance of these clustering results, we analyzed the instantiation of the results derived from the MLS algorithm. There are a total of 66 modules are shown in Fig. 6. There are 66, 101 and 206 modules derived from Gavin after being processed by MLS, MCL, LC methods. And the module size distribution is shown in Fig. 7. Comparison with

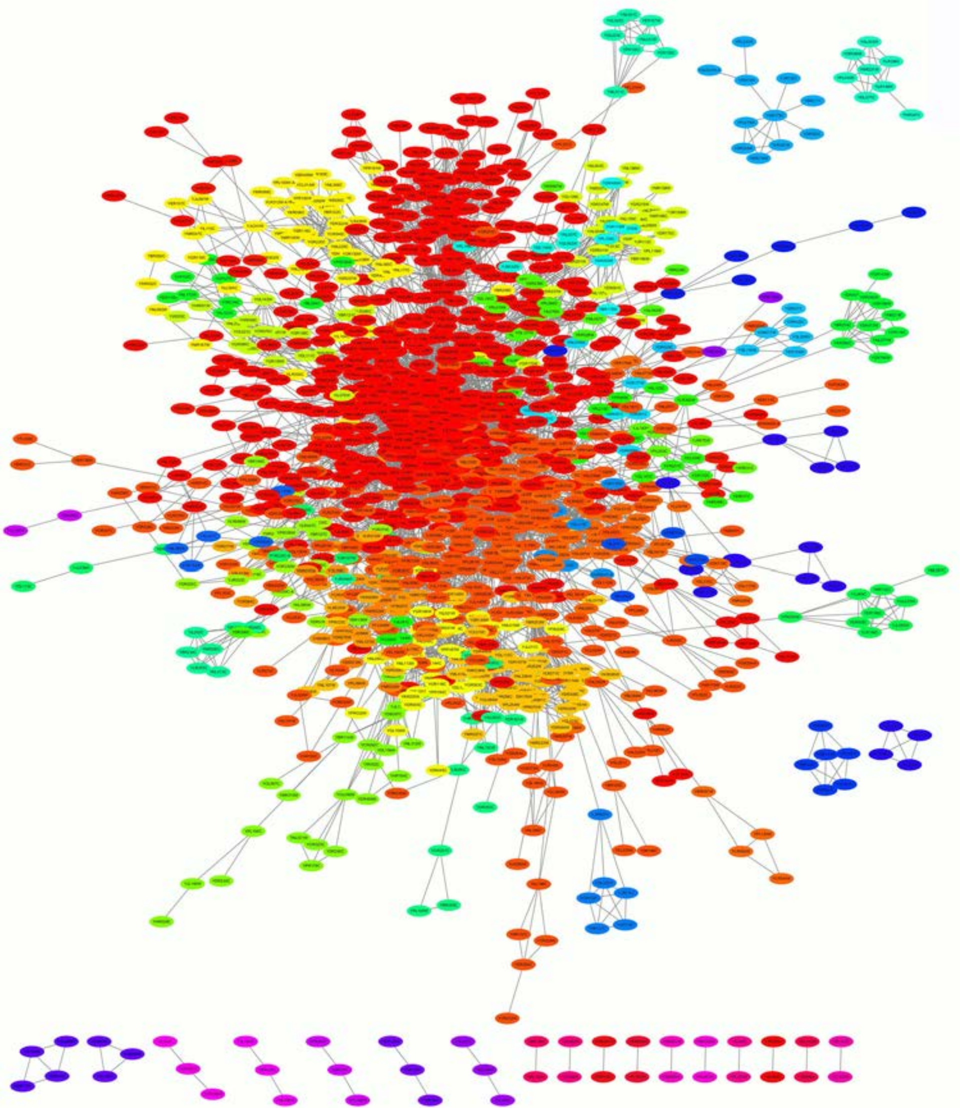

Fig. 6 MLS result on Gavin. Sixty-six modules in PPI network by using MLC. The inset shows the overlapping nodes of module of the network. Nodes in the inset represent modules on the original network

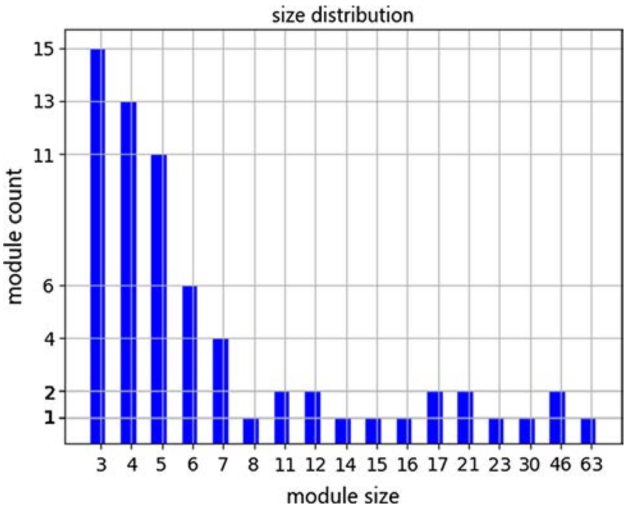

Fig. 7 MLS result on Gavin. It represents the size distribution of DIP PPI modules obtained by MLS. The x-axis indicates the size of modules, i.e., the number of proteins in each module. The y-axis shows the number of modules with the size corresponding to the x-axis

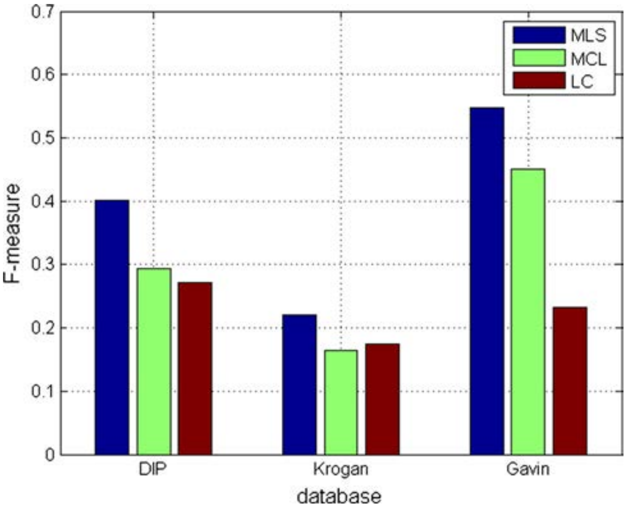

Fig. 8 F-measure of each network MLS/MCL/LC clustering algorithms on DIP, Krogan and Gavin

MLS/MCL/LC methods on Krogan and Gavin by different evaluations is shown in Table 4. Figure 8 shows that the *F-measure* derived from MLS/MCL/LC clustering algorithms of each network on DIP, Krogan and Gavin.

5 Conclusion

Clustering protein–protein interaction networks for the discovery of protein functional modules has drawn a lot of attention. However, existing clustering algorithms do not take into account the fact that overlapping structures in PPI networks are usually important for biological significance. Therefore, in an attempt to identify overlapping structure of PPI networks and improve the accuracy of detecting protein functional modules, we propose a clustering algorithm based on links that aims at avoiding local optima and detecting the overlapping structures. It is based on the link similarity matrix of the transition probabilities of a random walk; thus, it can effectively solve the bridge partition among classes and divide the overlapping structures into tightly formed functional modules, which have the same protein annotation. The results of experiments on the three PPI networks have indicated that the F-measure/EQ/CR value of MLS algorithm has larger ascension than the original LC algorithm and classical MCL algorithm. There are still shortcomings, such as the high computational complexity, the large number of big clusters and the large number of clusters. As part of future work, we will continue improving our MLS method after taking the dynamic and emergent properties of PPI network into account in order to discover more functional modules which are crucial in multiple species.

Acknowledgements We thank the reviewers for their thoughtful comments and suggestions. This work was supported by the National Natural Science Foundation of China (Grant Nos. 31771679, 31371533), the Special Fund for Key Program of Science and Technology of Anhui Province of China (Grant No.15czz03131, 16030701092), Project supported by the Natural Science Foundation of the Anhui Higher Education Institutions of China (Grant No. kJ2016A836).

References

1. Bader GD, Hogue CWV (2003) An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinf 4(1):1471–2105
2. Girvan M, Newman MEJ (2002) Community structure in social and biological networks. Proc Natl Acad Sci 99(12):7821–7826
3. King AD, Pržulj N, Jurisica I (2004) Protein complex prediction via cost-based clustering. Bioinformatics 20(17):3013–3020
4. Enright AJ, Van Dongen S, Van Ouzounis CA (2002) An efficient algorithm for large-scale detection of protein families. Nucl Acids Res 30(7):1575–1584
5. Samuel J, Yuan X, Yuan X, et al (2010) Mining online full-text literature for novel protein interaction discovery. In: IEEE international conference on bioinformatics and biomedicine workshops (BIBMW). IEEE, pp 277–282
6. Nepusz T, Yu H, Paccanaro A (2012) Detecting overlapping protein complexes in protein-protein interaction networks. Nat Methods 9(5):471–472
7. Brohée S, Helden JV (2006) Evaluation of clustering algorithms for protein-protein interaction networks. BMC Bioinf 7(1602):2791–2797
8. Satuluri V, Parthasarathy S (2009) Scalable graph clustering using stochastic flows: applications to community discovery. In: ACM SIGKDD international conference on knowledge discovery and data mining, Paris, France, June 28–July, 2009, DBLP, pp 737–746

9. Shih YK, Parthasarathy S (2012) Identifying functional modules in interaction networks through overlapping Markov clustering. Bioinformatics 28(18):i473–i479
10. Ahn YY, Bagrow JP, Lehmann S (2010) Link communities reveal multiscale complexity in networks. Nature 466(7307):761–764
11. Fortunato S (2010) Community detection in graphs. Phys Rep 486(3):75–174
12. Wang Y, Wang G, Meng D, et al (2014) A Markov clustering based link clustering method for overlapping module identification in yeast protein-protein interaction networks. In: Proceedings of the 10th international symposium on bioinformatics research and applications, ISBRA, Zhangjiajie, China, June 28–30. Springer, 8492, p 385
13. Yao FY, Chen L (2014) Similarity propagation based link prediction in bipartite networks. In: Proceedings of the 2014 international conference on network security and communication engineering (NSCE 2014), Hong Kong, Dec 25–26. CRC Press, pp 295–297
14. Meyer AS, Garcia AAF, Souza AP et al (2004) Comparison of similarity coefficients used for cluster analysis with dominant markers in maize (*Zea mays* L. Genet Mol Biol 27(1):83–91
15. Leger JB, Daudin JJ, Vacher C (2015) Clustering methods differ in their ability to detect patterns in ecological networks. Methods Ecol Evol 6(4):474–481
16. Xenarios I, Salwinski L, Duan XJ et al (2002) DIP, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. Nucl Acids Res 30(1):303–305
17. Gavin AC, Bösche M, Krause R et al (2002) Functional organization of the yeast proteome by systematic analysis of protein complexes. Nature 415(6868):141–147
18. Krogan NJ, Cagney G, Yu H, et al (2006) Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. Nature 440(7084):637–643. https://search.proquest.com/docview/204545168?accountid=45184
19. Pu S, Wong J, Turner B et al (2009) Up-to-date catalogues of yeast protein complexes. Nucl Acids Res 37(3):825–831
20. Newman MEJ, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E 69(2):026113
21. Shen H, Cheng X, Cai K et al (2009) Detect overlapping and hierarchical community structure in networks. Physica A 388(8):1706–1712
22. Li M, Wang J, Chen J (2008) A fast agglomerate algorithm for mining functional modules in protein interaction networks. In: International Conference on BMEI. IEEE, 1:3–7
23. Li IH, Huang JY, Liao IE, et al (2013) A sequence classification model based on pattern coverage rate. In: International conference on grid and pervasive computing. Springer, Berlin, pp 737–745
24. Rhrissorrakrai K, Gunsalus KC (2011) MINE: module identification in networks. BMC Bioinformatics 12(1):192
25. Zhao B, Wang J, Li M et al (2016) A new method for predicting protein functions from dynamic weighted interactome networks. IEEE Trans Nanobiosci 15(2):131–139
26. Zuo YC, Su WX, Zhang SH et al (2015) Discrimination of membrane transporter protein types using K-nearest neighbor method derived from the similarity distance of total diversity measure. Mol BioSyst 11(3):950–957
27. Sætre R, Sagae K, Tsujii JI (2007) Syntactic features for protein-protein interaction extraction. In: Short paper proceedings of the international symposium on languages in biology and medicine, DBL
28. Zhao B, Wang J, Li M et al (2014) Detecting protein complexes based on uncertain graph model. IEEE/ACM Trans Comput Biol Bioinf (TCBB) 11(3):486–497
29. Butz M, Steenbuck ID, van Ooyen A (2014) Homeostatic structural plasticity increases the efficiency of small-world networks. Front Synaptic Neurosci 6:7
30. Schuch B, Feigenbutz M, Makino DL et al (2014) The exosome-binding factors Rrp6 and Rrp47 form a composite surface for recruiting the Mtr4 helicase. EMBO J 33(23):2829–2846
31. Gu L, Wang C, Zhang Y et al (2014) Trust model in cloud computing environment based on fuzzy theory. Int J Comput Commun Control 9(5):570–583