PLOS ONE

RESEARCH ARTICLE

From communities to protein complexes: A local community detection algorithm on PPI networks

Saharnaz Dilmaghani[1]*, Matthias R. Brust[1]*, Carlos H. C. Ribeiro[2], Emmanuel Kieffer[3], Grégoire Danoy[1,3], Pascal Bouvry[1,3]

1 Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Esch-sur-Alzette, Luxembourg, **2** Computer Science Division, Aeronautics Institute of Technology (ITA), São Josédos Campos, Brazil, **3** Faculty of Science, Technology and Medicine (FSTM), University of Luxembourg, Esch-sur-Alzette, Luxembourg

* saharnaz.dilmaghani@uni.lu (SD); matthias.brust@uni.lu (MRB)

Citation: Dilmaghani S, Brust MR, Ribeiro CHC, Kieffer E, Danoy G, Bouvry P (2022) From communities to protein complexes: A local community detection algorithm on PPI networks. PLoS ONE 17(1): e0260484. https://doi.org/10.1371/journal.pone.0260484

Editor: Hocine Cherifi, Unviersity of Burgundy, FRANCE

Received: April 13, 2021

Accepted: November 10, 2021

Published: January 27, 2022

Data Availability Statement: The data set used in this manuscript is directly taken from the following link: https://thebiogrid.org/21816/publication/global-landscape-of-protein-complexes-in-the-yeast-saccharomyces-cerevisiae.html.

Funding: This work has been funded by the joint research programme University of Luxembourg/ SnTILNAS on Digital Trust for Smart-ICT.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Identifying protein complexes in protein-protein interaction (PPI) networks is often handled as a community detection problem, with algorithms generally relying exclusively on the network topology for discovering a solution. The advancement of experimental techniques on PPI has motivated the generation of many Gene Ontology (GO) databases. Incorporating the functionality extracted from GO with the topological properties from the underlying PPI network yield a novel approach to identify protein complexes. Additionally, most of the existing algorithms use global measures that operate on the entire network to identify communities. The result of using global metrics are large communities that are often not correlated with the functionality of the proteins. Moreover, PPI network analysis shows that most of the biological functions possibly lie between local neighbours in PPI networks, which are not identifiable with global metrics. In this paper, we propose a local community detection algorithm, (LCDA-GO), that uniquely exploits information of functionality from GO combined with the network topology. LCDA-GO identifies the community of each protein based on the topological and functional knowledge acquired solely from the local neighbour proteins within the PPI network. Experimental results using the Krogan dataset demonstrate that our algorithm outperforms in most cases state-of-the-art approaches in assessment based on *Precision*, *Sensitivity*, and particularly *Composite Score*. We also deployed LCDA, the local-topology based precursor of LCDA-GO, to compare with a similar state-of-the-art approach that exclusively incorporates topological information of PPI networks for community detection. In addition to the high quality of the results, one main advantage of LCDA-GO is its low computational time complexity.

Introduction

Proteins work cooperatively to accomplish biological functions. The physical interaction between proteins, known as *protein-protein interaction* (PPI), is the key for many biological functions [1], for example, the transcription of DNA, the translation of mRNA, and cell cycle [2]. Scientific progress on PPI is highly critical for applications such as protein function discovery [3], disease comprehension [4], and drug discovery [5].

To infer the physical interactions of proteins, a number of experimental techniques have been developed, such as *yeast-two-hybrid* (Y2H) [6] and *tandem affinity purification* (TAP) [7]. This resulted in the generation of several depositories and databases of experimental data on PPI (e.g., BIOGRID). While these screening methods facilitate the comprehension of PPI, they have been widely criticized due to the false negative (i.e., not being able to detect interacting proteins) and false positive (i.e., identifying non-interacting proteins as interacting proteins) interaction detection. Therefore, high-throughput screening methods suffer from a considerable lack of accuracy and thus, produce an incomplete map of the interactions among the proteins [8–10].

The pairwise physical interaction of proteins within the PPI data suggests a network representation where nodes are the proteins and links are the interactions among the proteins. Exploiting network structure with network analysis tools on such data has shifted the PPI analysis to the *network* level. Besides, the existence of protein complexes justifies the high-degree clusters within the PPI network [9]. PPI networks inherit both *topological* and *functional* information [1]. The first term refers to the physical interaction describing the arrangements of the nodes in the network, and is associated with the densely connected proteins namely *communities*. The latter explains the biological function of proteins that are achieved by groups of proteins that bind each other and shape *protein complexes*. The complexes are explained by the annotations available in Gene Ontology (GO) [11, 12] databases. GO provides a specific definition of protein functions and it is one of the main resources of biological information. GO provides a structured and controlled vocabulary of terms, which are subdivided into three non-overlapping ontologies: Molecular Function (MF), Biological Process (BP) and Cellular Component (CC) [13]. We utilize GO terms to enrich PPI networks with functional properties of proteins.

It is acknowledged that in several cases, those proteins that are topologically interconnected represent similar biological processes (i.e., GO terms) [14], thereby there is an overlap between the communities of proteins and complexes. Nevertheless, the two terms are distinguished entities in PPI networks. Moreover, biological networks such as PPI networks share a common feature refereed as *local cluster connectivity* [15] that highlights the locality of the biological functions in PPI networks that are possible only between local neighbours.

Because of the correlation that exists between protein communities and complexes, detecting protein complexes from PPI networks can be translated into a community detection problem [2, 16, 17]. The purpose of a community detection algorithm for PPI networks is to divide proteins into groups such that the proteins of the same group are more similar to each other rather than those in the other groups. The state-of-the-art solutions consider different objectives to divide the nodes of a given network into highly interconnected communities [18–20]. Some of these algorithms are adjusted to biological networks to tackle the protein complex detection in PPI networks [21], including C-FINDER, COACH, CLUSTERONE, MCL, CMC, MCODE, and CORE&PEEL. Even though the community detection algorithms drive optimal topological communities in PPI networks, they suffer from the particular biological nature of the network due to the disengagement of functional properties. [2, 10, 22, 23].

The extracted interactions from experimental techniques (e.g., Y2H, TAP) are sometimes biased with incorrect inferring of existing and non-existing relationships. In other words, the

available PPI networks could be incomplete and unreliable with respect to the detected nodes and links [9]. That in return will impact the results of the communities if the method depends solely on the existing topology of the network [24]. Moreover, some of the existing community detection algorithms acquire the whole network, that could be inherently incomplete, and hence results in large tangled communities of mixed or broad functionality [25] that do not explain adequately the underlying PPI network [23, 26]. In addition, such algorithms perform based on the global measures that are expensive in time complexity.

Encoding biological information in PPI networks could address the challenge of detecting higher quality communities of proteins with respect to their biological nature. The functionality hence could be achieved by incorporating biological information from the annotated databases (e.g., GO, DAVID). DCAFP [27], GMFTP [28], and MTGO [23] are some of the algorithms that are designed in a similar way. To tackle the next challenge regarding the reliability of the data and missing information, one possible solution could be to diminish the impact of network structure by focusing only on the local neighbours [29].

In this paper, we propose LCDA-GO, a local community detection algorithm that combines topological and functional properties (i.e., GO terms) of PPI networks to detect associated communities that are representing protein complexes. One of the main advantages of LCDA-GO is the strong degree of locality [30] devised in the algorithm which not only reduces the dependency to the network structure but also equips the algorithm with a considerably low time complexity when compared to other state-of-the-art approaches. We compare LCDA-GO with the state-of-the-art algorithm that incorporates the topology and functionalities by exploiting GO to detect protein complexes. We also expand our experiments by providing a comparative evaluation with state-of-the-art protein complex detection approaches relying only on the topology of the network. For this experiment, we have used the LCDA algorithm [29], the local-topology based precursor of LCDA-GO.

Related work

Many algorithms have been proposed to detect communities in PPI networks [2, 21, 31, 32]. Some of these approaches just rely on the topology of the PPI networks to detect communities, while others combine the biological functionality of the nodes to enrich the network and hence complex detection. We classify the existing community detection algorithms used for protein complexes in two categories based on the properties that an algorithm incorporates to detect the communities. We first explain community detection algorithms that perform solely on the *topology* of a network, and then, we discuss algorithms that rely on both *topology* and *functionality*.

Topological approaches

One of the earliest algorithms that has been developed for PPI networks community detection is MCODE [33]. It enjoys a level of locality, by expanding a set of high-ranked nodes (i.e., source nodes) into communities. MCODE often represents very large communities and hence the number of predicted real complexes is small. The Markov Cluster algorithm (MCL) [34] is also utilized on PPI networks. The algorithm is a robust method based on a random walk to partition the network into communities. ClusterOne is a greedy approach starting from a seed node. The nodes with high cohesiveness are added or removed from the communities in an iterative process. ClusterOne is an overlapping community detection approach and it merges those groups of proteins that satisfy an overlap score.

For the comparative evaluation we used MCODE, MCL, and ClusterOne [35] to measure the performance differences of our LCDA algorithm, a version of LCDA-GO performing based on just

local topological properties. Other algorithms such as COACH [36] and LCMA [37], and CFINDER [38] also benefit from topology of the network to find the communities. These algorithms are discussed in [2, 21, 31].

Topological and functional approaches

Recent approaches benefit from functional enrichment of the network to accurately detect the communities of proteins in PPI networks. The main motivation of such algorithms lies in the fact that protein complexes are mostly aggregated in performing common functions. One of the earliest approaches in this category is RNSC [39]. This algorithm is initialized with a random partitioning that is optimized based on the minimum cost for node exchanging. It considers density and functional homogeneity to search for better communities. Its performance, however, depends on the initial community assignment. MTGO [23] is a recent approach that combines both topological and functionality of the PPI networks to detect the communities. Similarly to RNSC, MTGO initializes the process by a random partitioning, and decides on rejoining the nodes into the communities if they share a common functionality and also if the new node increases the modularity of the community. The algorithm relies on two parameters *min* and *max* that control the size of the communities and impact the outcome. GMFTP [28] and DCAFP [27] are two other algorithms that are designed similarly by exploiting functionality, however, the biological nature of the networks are not directly involved in the main process and it is rather processed in advance by the network topology.

Our proposed LCDA-GO approach is similar to mentioned algorithms such that it combines both topological and functional information. However, unlike RNSC, MTGO, our proposed model does not rely on any random partitioning nor is restricted to initial input parameters. The results of LCDA-GO is compared to MTGO in Experiments and Results Section.

Local Community Detection Algorithm for protein complexes with Gene Ontology (LCDA-GO)

In this section, we introduce the basic notation and terminologies that will be used through the paper. We also describe how LCDA-GO is implemented to detect communities of proteins exploiting topological and functional properties based on local conditional rules.

Notation and Preliminaries

We assume an undirected and unweighted network $G = (V, E)$, where V and E represent the set of nodes and the set of links, respectively. Our purpose is to divide G into set of communities, C, such that each node $v \in V$ belongs exclusively to one community c_i, and $C = \bigcup c_i$. A high quality community is a densely intra-connected (topology property) group of proteins representing lowest variation of GO terms (functional property). LCDA-GO finds communities based on both topological and functional properties in a local manner. The algorithm allows each node to adjust its community label, *cl*, given the local neighbourhoods.

On a given PPI network, LCDA-GO represents communities by a source node that is discovered during the algorithm. A source node is one of the high-degree nodes of the community and is connected to the nodes that have similar functional properties. The distance from the source node of a community to node v is stored in hl_v. A snapshot of LCDA-GO performance is illustrated in Fig 1 showing the process for node v. In this scenario, v has three neighbours [c, d, t], such that node c and d belong to 'a' and t is from x (i.e., cl_c = 'a', cl_d = 'a', cl_t = 'x'). Besides, the numbers show *hl* of each node, that is the hop-distance from the source node of the community. According to this example node c and node d are 1 and 2 hops away from the source node of their community (i.e., a), respectively, and t is 3 hops away from its source node, x. It

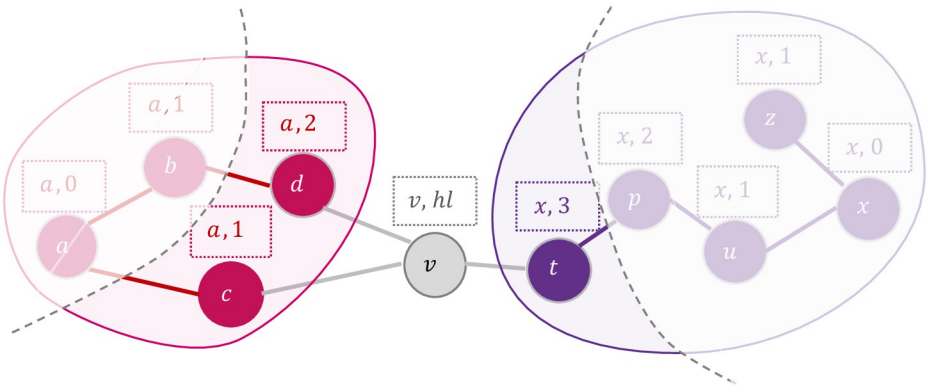

Fig 1. A snapshot of the community structures and local information that LCDA-GO is implemented on for node *v*. The transparent area is unknown zone that is not available during the operations. Thus, each node performs relying on the knowledge of its first neighbours. In this example, *c* and *d* are from community *a* and *t* is in community *x*. The community label describes the source node of the community, hence, *a* and *x* are two surrounded communities of *v*. The numbers attached to each node describes the hop-distance of the node from its community presenter. During the implementation, we have considered *hl* of a source node equal to 1 instead of 0.

https://doi.org/10.1371/journal.pone.0260484.g001

is worth mentioning that *v* does not have any other knowledge about the rest of the network as shown in the transparent zone in Fig 1.

Besides the above-mentioned topological variables, *cl* and *hl*, that are consider in LCDA-GO, *g* is also determined to store GO terms that a protein is contributed. To access a decision on the community of node *v*, LCDA-GO calculates two parameters as defined in the following:

Definition 1. *(Community influence degree.)* The community influence degree of node *v* is calculated between *v* and its neighbours from community c_i as follows:

$$\lambda(v)^{u\in[\Gamma(v)\cap c_i]} = \ln\left(\frac{k_v}{hl_v}\right).|g_v \cap g_{u\in[\Gamma(v)\cap c_i]}|, \tag{1}$$

where $|g_v \cap g_{u\in c_i}|$ is the number of common GO functions between *v* and its neighbours from community c_i. The intuition behind the community influence degree is that a node is more likely to be in the same community as a neighbour node if the following node is closer to the source node of the community, has a higher degree, and shares similar functions with the neighbour node. If in a community one node has a higher community influence degree, the node could be a potential source node.

Definition 2. *(Local community modularity.)* The local community modularity for a node *v* is calculate for a surrounded community c_i as:

$$\begin{aligned} \mu(v)^{c_i} &= \frac{E_{in} - E_{out}}{E_{in} + E_{out}} \\ &= 2\frac{E_{in}}{E_{in} + E_{out}} - 1, \end{aligned} \tag{2}$$

where E_{in} is the number of links connecting node *v* to nodes from community c_i, and E_{out} represents the links to the other nodes. The value of local community modularity can vary in the range of (−1, 1]. It takes a negative value if there is no link to community c_i. The value is positive if the number of links connected to c_i surpasses the number of links to other communities. Local community modularity performs as a measure of community extension by adding *v* to c_i, if $\mu_v^{c_i}$ is positive.

A list of the notations used in the paper is summarized in Table 1.

Table 1. Notation exploited in LCDA-GO.

G	A PPI network
C	Set of solution that consists of communities of c_i such that $C = \bigcup c_i$
v	The current node
$\Gamma(v)$	Neighbours of node v
$k(v)$	Degree of node v
$cl(v)$	Community label of v
$hl(v)$	Hop-distance from the community source node
$g(v)$	GO terms of node v (i.e., functional properties)
$\lambda(v)$	Community influence degree on node v
$\mu(v)$	Local community modularity

https://doi.org/10.1371/journal.pone.0260484.t001

Algorithm description

We propose an iterative bottom-up approach, LCDA-GO, allowing each node to take a decision of joining a community independently. Our algorithm starts from a node and discovers the network through each node's direct neighbours. LCDA-GO relies on a set of conditional rules to expand or generate new communities. The Local Community Expansion Rules (LCER) operate on each node based on the acquired local neighbourhood information as explained in Notation and Preliminaries Subsection. At each step of LCDA-GO nodes adjust their hop-distance (*hl*) value according to their distance from source nodes. If a node has a higher community influence degree and meets the conditions, it will become a source node. Thus, its *hl* is updated to 1. In this case, all neighbour nodes adjust their *hl* according to their hop-distance from the source node. LCDA-GO converges when all nodes agree with their community labels. A pseudo code of the proposed LCDA-GO is described in Alg. 1 LCDA-GO. The algorithm starts by initializing the node list R (line 1), that records the visited nodes and their neighbours. The initial node is either a given node or randomly selected from the network. As a first-time-visited node in the list, the community label *cl* of the node is assumed as it ID, in this case, v, and its hop-distance *hl* is set to the constant value of *HL* (line 2-3). We chose $HL = 4$ initially, however, it can be any value larger than 1. The next step is to adjust *v.hl*: If *v.hl* is the highest compared to v's neighbours, then it will be reduced by 1 (lines 7-8). Afterwards, $\lambda(v)$ and $\mu(v)$ is calculated (lines 10-11) and v is transmitted to Alg. 2 LCER (line 12) to make a decision regarding its *cl*. employing LCER on v, its attributes such as *cl* and *hl* will be updated consequently. Next, R expands by including neighbours of v. The processes continue such that all nodes of V is included in R and updated by LCER. Finally, if all nodes come to an agreement such that no further changes occur in community structure and each node of the network is declared in one community, the algorithm will converge. The `stopCondition` is defined as follows:

$$\texttt{stopCondition} = \begin{cases} 1, & \text{if } (R == V) \ \& \ (\text{for } v \text{ in } R,\ v.cl \text{ doesnt change}) \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

After the convergence of LCDA-GO, the set of communities is obtained by retrieving each node's *cl* from R.

Algorithm 1 LCDA-GO

```
Input: Network G
Output: C set of communities
  Initialisation:
1: R ← v from V
2: v.hl = HL
```

```
3: v.cl = v
4: v.g = GO[v]
  Procedure:
5: while stopCondition do
6:   for v in R do
7:     if k_v > max(k_Γ(v)) then
8:       v.hl ← v.hl − 1
9:     end if
10:     v.λ = λ(v)
11:     v.μ = μ(v)
12:     LCER(v)
13:     R.update ← Γ(v)
14:   end for
15: end while
16: return C.update ← cl from nodes of R
```

We defined Alg. 2 LCER to decide the corresponding community of v. For an input node v, it first calculates the local community modularity. Instead of computing the function for each c_i, we only consider the larger community(ies) which has the larger number of links to v. We assume that u is the larger community. If $\mu(v)$ is positive, v joins community u. Thus, the community label of v changes to u (line 3), and the hop-distance shift to the shortest path from v to the source node u (line 4). To measure the shortest path, we simply consider the minimum hl of the neighbours plus 1. In case $\mu(v)$ is negative or zero, one of these two scenario may occur: First, the algorithm checks for the possibility of v itself being a source node. It means that node v is selected by the neighbours as the source node, while its attributes are not updated. Hence, the attributes of v are changed to fit the condition (line 7-8). Otherwise, v changes its attributes to follow the most similar node in its neighbourhood, which is node p with highest community influence degree (line 9-10). then, either v itself is selected by the neighbours to be a new community, or it will temporarily follow the best candidate among its neighbourhoods.

Algorithm 2 LCER

```
1: if (μ(v) > 0) then
2:   v.hl = min(Γ(v).hl) + 1
3:   v.cl = u
4: else if (μ(v) <= 0) then
5:   if v.cl is u then
6:     v.hl = 1
7:     v.cl = u
8:   else
9:     v.hl = p.hl
10:     v.cl = p.cl
11:   end if
12: end if
```

Computational complexity

The complexity of the proposed algorithm is determined by two loops in the algorithms. The outer *while*-loop in Alg. 1 LCDA-GO—line 5 coordinates the convergence of LCDA-GO to ensure that all nodes have come to an agreement about their community assignments. The recurrence (t) of the outer loop is independent from the size of the network. Our experiments with various networks' sizes [29] shows that $8 \leq t \leq 15$. The inner *for*-loop of LCDA-GO described in 1 line 6, operates a set of conditional rules over each node from list R. The performance of the inner loop has the highest impact on the overall complexity of LCDA-GO.

The complexity of the inner loop on a network G of size n can be estimated as follows. The repetition of the loop changes as R is updated. The list of neighbours (i.e., R) initially starts

with the neighbours of node v. Let us assume k is the average degree of G. In this case, The initial size of R, in other words, the repetition of the first loop is k ($t_1 = k$). As R progressively is extended by adding other nodes, the next loop repetitions $t_2, t_3, \ldots, t_m$ increases as well. To calculate the complexity, we need to sum up all recurrences of the loop: $\{t_1 = k, t_2 = k^2, \ldots, t_m = k^m\}$. Considering the size of the network, the final R includes all nodes of G, therefore, $t_m = k^m = n$. Then, the complexity of the series that is combining the loops is $O(t \times n)$, with t representing the iterations over the outer *while*-loop. In addition, according to our experiments [29] t $log(n)$, hence the average-case complexity of LDA-GO is in the order of $nlog(n)$.

The worst-case scenario happens when the inner-loop runs over V instead of R. In this case, each iteration performs on n nodes instead of k. The recurrence of the inner-loop is then, $\{t_1 = n, t_2 = n, \ldots, t_m = n\}$. However, the iterations of outer-loop remains the same since it is independent from the inner-loop. Hence, the worst case complexity stays as same as the average complexity, $O(nlog(n))$.

Experiments and results

In this section, we first describe the PPI network dataset, GO [12] terms that are used to enrich the network, and the benchmark dataset. Next, we define the metrics and measures that we use to evaluate the performance of our algorithms, LCDA and LCDA-GO. Finally, we provide a comparative evaluation to show the performance of our algorithm compared to state-of-the-art algorithms.

PPI network and Gene Ontology (GO)

To evaluate LCDA-GO and LCDA, Krogan [40] dataset is selected. It includes a set of nodes (i.e., proteins) and associated links (i.e., interactions) built on yeast *Saccharomyces Cerevisiae* data. We download the dataset from BioGrid database [41]. To include the functionality we exploit Gene Ontology (GO) terms from Panther database [42]. GO terms are subdivided into three categories of Molecular Function (MF), Biological Process (BP) and Cellular Component (CC). We extract the GO terms of Krogan PPI network. For evaluating the outcome, we use gold standard protein complexes CYC2008 [43] as target sets to evaluate the predicted communities resulted from LCDA-GO. The information associated with the database and datasets are described in Table 2.

Table 2. Datasets of networks used for the experiments.

PPI Network					
Datasets	$\|V\|$	$\|E\|$	avg. degree	# CC	$\|G_{cc}\|$
Krogan [40]	2674	7079	5.29	62	2527
PPI + MF	1014	2135	4.21	7	995
PPI + BP	1154	2502	4.33	8	1130
PPI + CC	1160	2710	4.67	10	1130
PPI + All	1523	3708	4.86	9	1498
Gene Ontology (GO)					
Database	Proteins	# MF functions	# BP functions	# CC functions	All functions
Panther [42]	2358	8	11	3	22
Benchmark					
Database	Proteins	Complexes	# ∩ Krogan	# ∩ Panther	
CYC2008 [43]	1920	408	970	813	

https://doi.org/10.1371/journal.pone.0260484.t002

Krogan PPI network [40] dataset, includes 2674 proteins in total. Our analysis found 62 connected components with a giant connected component including 2527 proteins, while 42 of the components had less than 3 nodes. For the community detection, we removed all those 42 components that will not shape a community. The final PPI network includes 2590 proteins.

We generated four PPI networks from the original Krogan PPI network according to GO term categories: PPI + MF, PPI + BP, PPI + CC, PPI + ALL, such that the last network includes all the functions. We also keep the original Krogan network without annotations for further analysis. All five networks are refined by filtering the connected components with the size of less than 3 proteins.

Evaluation metrics

Before presenting the evaluation results, we describe various metrics that are mostly used in the literature [2, 23, 31, 32] to assess detected complexes in PPI networks. Exploiting these metrics, we then compare the state-of-the-art algorithms with our proposed algorithm and describe them.

Neighbour affinity score. To quantify the similarity of the detected complex $p = (V_p, E_p)$ with the benchmark $b = (V_b, E_b)$, we use the neighbour affinity score (AS) as defined in Eq 4. This metric considers both the size of the two complexes and the common proteins in the two sets to measure the similarity between the two. In case the predicted complex is exact equal to the real complex, then AS will be equal to 1. For two complexes of p and b the affinity score is defined as follows:

$$AS(p, b) = \frac{|V_p \cap V_b|^2}{|V_p|.|V_b|} \tag{4}$$

where V_p is the number of proteins from the predicted complex and V_b is the number of proteins in the benchmark complex. We define a threshold θ, $AS(p, b) \geq \theta$, to control the strength of the similarity measured by AS. We consider $\theta = 0.1$ to get results from all algorithms.

Precision, recall, and F-measure. Among the standard metrics to evaluate the predicted values based on the benchmark are *Precision*, *Recall*, and $F - measure$. However, the metrics that we have implemented in this paper for the evaluation are slightly different than the common definition for the *Precision*, *Recall*, and F-measure and are similar to [2, 44]. We use AS as defined in Eq 4 to choose a good match between the predicted and benchmark complexes. Assume that p is a predicted complex from the set of all predicted complexes P, and b is a benchmark complex from set B that includes all benchmark complexes. In this case, N_{cp} and N_{cb} are defined as follows:

$$\begin{aligned} N_{cp} &= |\{\forall p | p \in P, \exists b \in B, AS(p, b) \geq \theta\}|, \\ N_{cb} &= |\{\forall b | b \in B, \exists p \in P, AS(p, b) \geq \theta\}|. \end{aligned} \tag{5}$$

Based on the N_{cp} and N_{cb} values from Eq 5, Precision, Recall are defined as the fraction of the matched complexes from the predicted set P, and benchmark set B respectively, according to the Eq 6.

$$Precision = \frac{N_{cp}}{|P|}, \tag{6a}$$

$$Recall = \frac{N_{cb}}{|B|}. \tag{6b}$$

The harmonic average of *Precision* and *Recall*, known as F-measure, is then calculated as follows:

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{7}$$

We use these metrics to evaluate the overall performance of the detected complexes over the complexes within the benchmark.

Sensitivity, positive predicted value, and accuracy. Besides the metrics defined above, *Sensitivity* (*Sn*) (also called *Coverage*), *Positive Predicted Value* (*PPV*), and *Accuracy* (*Acc*) are used to evaluate the performance and accuracy of the detected complexes [2, 9, 32]. Consider T_{ij} equal the number of common proteins between i^{th} benchmark complexes and j^{th} predicted complex. N_i is the number of proteins the i^{th} benchmark complex. Given n is the overall number of b benchmark complexes and m predicted complexes p, then *Sn* and *PPV* are defined as follows:

$$Sn = \frac{\sum_{i=1}^{n} \max_j(T_{ij})}{\sum_{i=1}^{n} N_i}, \tag{8a}$$

$$PPV = \frac{\sum_{j=1}^{m} \max_i(T_{ij})}{\sum_{j=1}^{m} \sum_{i=1}^{n} T_{ij}}. \tag{8b}$$

Larger values of *Sn* indicate that the community detection algorithm has well-covered the proteins in the real complexes. On the other hand, *PPV* highlights the probability of true positives of protein complexes in predicted communities. The accuracy of the prediction, as a summary metric, can then be defined as as the geometric average of *Sn* and *PPV* as follows:

$$Acc = \sqrt{Sn \times PPV} \tag{9a}$$

In addition to the above-mentioned metrics, several studies [23, 35, 45] rely on another measure known as *Composite Score* [46] to make a comprehensive evaluation. Therefore, as a final global performance measure, we calculate the *Composite Score* by summing up the three values of *Precision*, *Sn*, and *Acc*. This value is important to avoid the advantage of evaluation metrics to another.

Comparative evaluation

We provide a set of experiments to compare the communities resulted from our algorithm with the state-of-the-art algorithms. We compared LCDA-GO and LCDA [29] with MCODE [33], MCL [34], CLUSTERONE [35], and MTGO [23]. We choose these algorithms to explore the benefits of topological and functional properties in the performance of protein complex detection methods.

Except our two algorithms, LCDA and LCDA-GO, other algorithms require setting up initial parameters such as *min size* of the community, in their software. Clearly, tuning the parameters could result in better performance, however, there is no principled way to discover the optimal values for these parameters rather than using their defined values. Table 3 describes a general overview of the results of employing different community detection algorithms on PPI networks. In all experiments, we benefit from the gold standard protein complexes of CYC2008 [43] as the benchmark.

To provide fair comparisons and for a detailed analysis, we have designed two experiments. In the first experiment, we only consider the communities that are detected by the algorithms

Table 3. An overview of the resulted communities from each algorithm including our method on *Saccharomyces Cerevisiae* Krogan interaction datasets.

PPI + MF					
Algorithms	MCODE	MCL	ClusterOne	LCDA	LCDA-GO
#communities	37	244	209	65	383
N_{cb}	4	160	142	69	167
N_{cp}	2	112	117	36	154
PPI + BP					
Algorithms	MCODE	MCL	ClusterOne	LCDA	LCDA-GO
#communities	38	256	236	71	416
N_{cb}	3	192	170	76	202
N_{cp}	3	149	146	51	196
PPI + CC					
Algorithms	MCODE	MCL	ClusterOne	LCDA	LCDA-GO
#communities	51	277	237	71	425
N_{cb}	6	196	180	80	210
N_{cp}	5	158	153	54	211
PPI + All					
Algorithms	MCODE	MCL	ClusterOne	LCDA	LCDA-GO
#communities	52	347	142	79	548
N_{cb}	4	213	122	78	223
N_{cp}	4	178	106	52	237

https://doi.org/10.1371/journal.pone.0260484.t003

only considering the topology of the network, namely, MCODE [33], MCL [34], ClusterOne [35], LCDA [29]. The second experiment is for evaluating the communities resulting from algorithms that are incorporating both topology and functionality. For this evaluation, we compared LCDA-GO with MTGO [23]. The next two subsections present the comparisons of these experiments.

Topological algorithms analysis. We compare our LCDA [29] algorithm that solely considers the topological interaction of the PPI network with other algorithms from the literature that perform in a similar manner. We select MCODE [33], MCL [34], and CLUSTERONE [35] for this comparison. We have used Cytoscape software [47] and exported the communities resulted from these methods. The input networks are extracted from Krogan dataset and divided based on GO functionalities. The assessments are described for all four algorithms in Table 4 based on the metrics explained earlier in this section. As presented in the table, the performance of MCODE is considerably low compared to the other algorithms, even though we have set $\theta = 0.1$ to relax the condition for *AS*. MCL has overall the highest *Recall*, *Fmeasure*, and *Acc*, while our LCDA algorithm outperforms other algorithms with the highest *Precision*, *Sn*, and particularly *Composite Score*. The performance of ClusterOne algorithm is also high and relatively close to both MCL and and our algorithm LCDA. The *Composite Score* is shown in Fig 2. The total height of each bar is the value of the *Composite Score* and the larger scores are better. The figure describes how the three algorithms are competing for a higher performance rank and LCDA is outperform them.

Topological and functional algorithms analysis. We implement and test our proposed algorithm for protein complex detection, LCDA-GO on all the networks extracted from Krogan dataset. The results are described in Table 5.

We choose MTGO to compare the results of LCDA-GO with since it also considers functionality as a parameter involved in the community detection and not as an in dependant process that

Table 4. Performance comparison of the communities of the algorithms that are based on only topology on *Saccharomyces Cerevisiae* Krogan interaction datasets. θ is 0.1.

PPI + MF							
Algorithms	*Precision*	*Recall*	*F-measure*	*Sn*	*PPV*	*Acc*	*Composite Score*
MCODE	0.05	0.01	0.02	0.02	**0.65**	0.11	0.19
MCL	0.45	**0.39**	**0.42**	0.26	0.60	**0.39**	1.11
ClusterOne	**0.55**	0.35	**0.42**	0.25	0.58	0.38	**1.19**
LCDA	**0.55**	0.16	0.26	**0.29**	0.33	0.31	1.16
PPI + BP							
Algorithms	*Precision*	*Recall*	*F-measure*	*Sn*	*PPV*	*Acc*	*Composite Score*
MCODE	0.07	0.00	0.01	0.02	**0.68**	0.12	0.22
MCL	0.58	**0.47**	**0.52**	0.34	0.62	**0.45**	1.38
ClusterOne	0.61	0.41	0.49	0.31	0.63	0.44	1.37
LCDA	**0.72**	0.17	0.30	**0.35**	0.35	0.35	**1.41**
PPI + CC							
Algorithms	*Precision*	*Recall*	*F-measure*	*Sn*	*PPV*	*Acc*	*Composite Score*
MCODE	0.10	0.01	0.02	0.03	**0.78**	0.15	0.28
MCL	0.57	**0.48**	**0.52**	0.34	0.65	**0.47**	1.39
ClusterOne	0.64	0.44	**0.52**	0.34	0.63	0.46	1.45
LCDA	**0.76**	0.20	0.31	**0.38**	0.34	0.36	**1.50**
PPI + All							
Algorithms	*Precision*	*Recall*	*F-measure*	*Sn*	*PPV*	*Acc*	*Composite Score*
MCODE	0.08	0.01	0.02	0.03	**0.75**	0.15	0.26
MCL	0.51	**0.52**	**0.51**	0.39	0.63	**0.50**	1.40
ClusterOne	**0.74**	0.30	0.45	0.30	0.60	0.42	1.46
LCDA	0.66	0.20	0.30	**0.44**	0.31	0.37	**1.47**

https://doi.org/10.1371/journal.pone.0260484.t004

could apply after community detection algorithm. We have exploited the MTGO software to run over the Krogan networks from Table 2, however, considering the large time complexity of this algorithm the final results could not converge by the time of writing this paper. Therefore, we decided to rely on the experiments attached to their studies for this comparison. We choose only *Sn*, *PPV*, and *Acc* to compare the results due to the fact that they are independent from the threshold required for *AS* score. The results are presented in Fig 3. As shown in this figure, even though MTGO has better *Sn* compared to LCDA-GO, *PPV* and *Acc* of LCDA-GO is larger. Overall, the two algorithms are competitive based on these assessments.

Computational complexity analysis. Besides, the relatively close results from LCDA-GO and MTGO is the complexity of the two algorithms. Due to the locality of LCDA-GO, our algorithm enjoys from the loglinear time complexity while MTGO is a polynomial time algorithm. Our algorithm is more than 1400 times faster than MTGO when performing on Krogan dataset with 2674 nodes. The time complexity of LCDA-GO and MTGO is compared in Table 6.

Discussion and conclusion

Identifying protein complexes is an important step for biological knowledge discovery since several biological processes are accomplished in the formation of protein complexes. In this paper, we propose a local community detection algorithm, LCDA-GO, for protein complexes by exploiting Gene Ontology (GO). LCDA-GO exploits networks' topological properties such as degree and shortest path in conjunction with protein's functional properties derived from GO

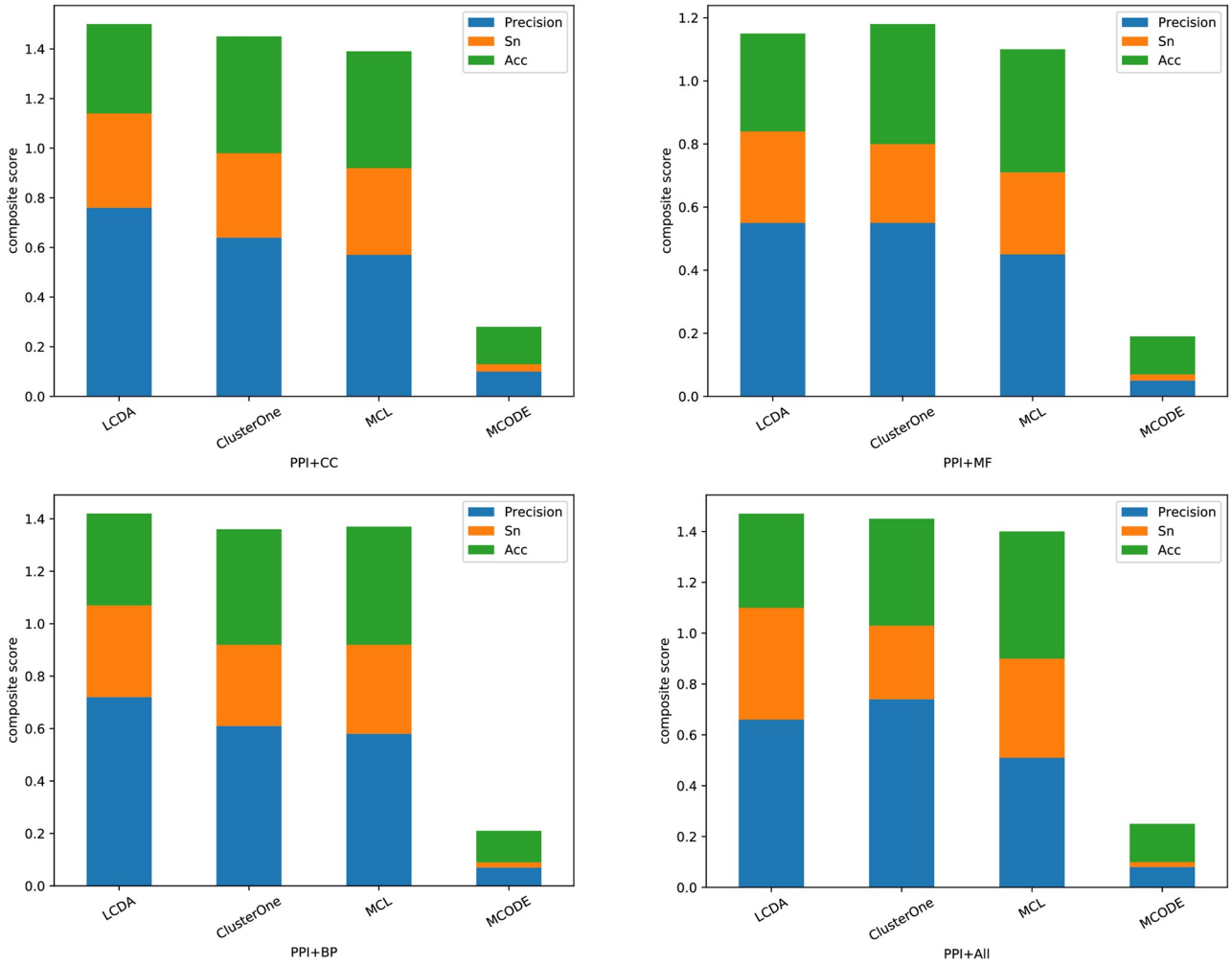

Fig 2. Composite score including *Precision*, *Sn*, and *Acc*.

https://doi.org/10.1371/journal.pone.0260484.g002

databases. Our algorithm employs both topological and functional properties in local measures to perform on PPI networks in a local procedure.

We evaluate LCDA-GO and another variation of the algorithm called LCDA, the latter relying only on the topology of the network. Experimental results demonstrate their performance on real-world PPI networks from the Krogan dataset and their capabilities in finding protein complexes.

In addition, the promising performance of LCDA and LCDA-GO show the capability of our algorithms in successfully detecting protein complexes in PPI network with significantly lower

Table 5. Performance of LCDA-GO on *Saccharomyces Cerevisiae* from Krogan interaction datasets.

Network	*Precision*	*Recall*	*F-measure*	*Sn*	*PPV*	*Acc*	*Composite Score*
PPI + MF	0.40	0.41	0.41	0.19	0.62	0.35	0.94
PPI + BP	0.72	0.17	0.30	0.35	0.35	0.35	1.41
PPI + CC	0.50	0.51	0.51	0.27	0.64	0.41	1.17
PPI + All	0.43	0.55	0.48	0.28	0.65	0.43	1.15

https://doi.org/10.1371/journal.pone.0260484.t005

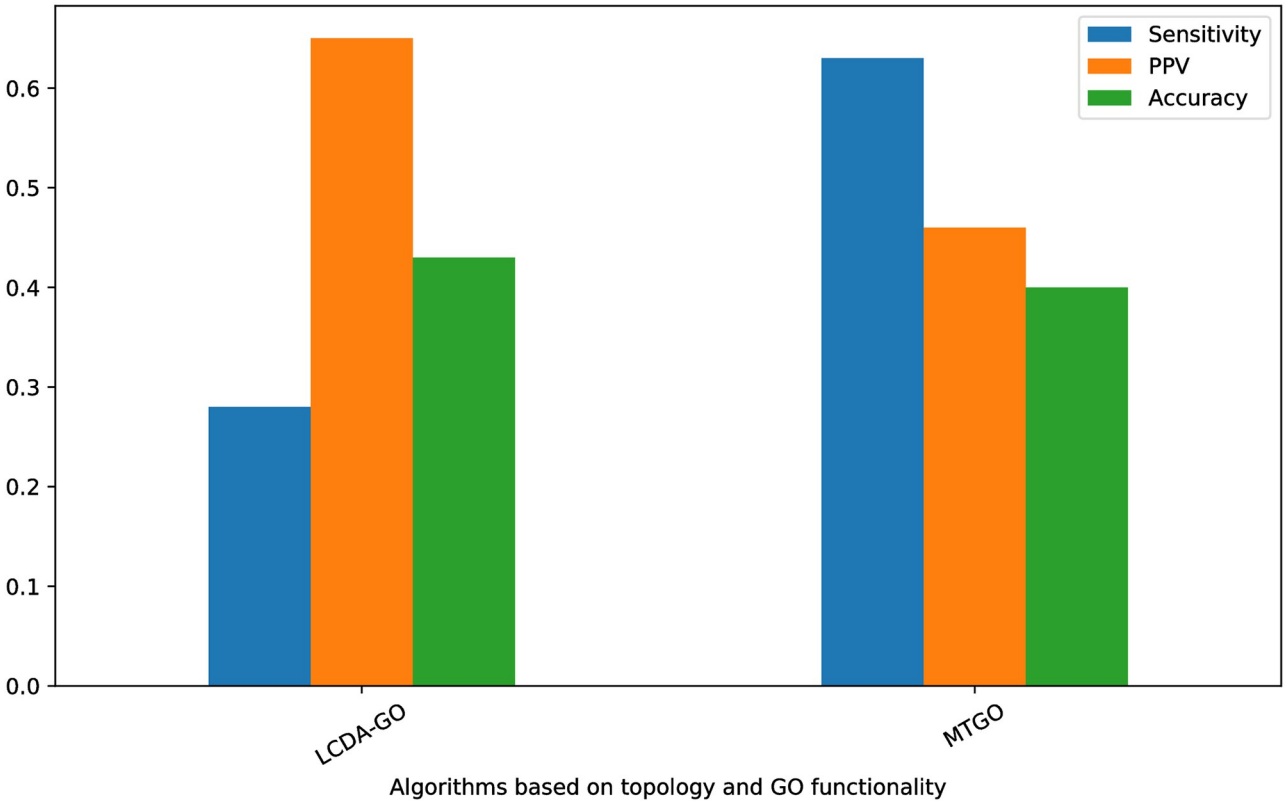

Fig 3. Comparing the results of LCDA-GO with MTGO on Krogan dataset.

https://doi.org/10.1371/journal.pone.0260484.g003

Table 6. Complexity and run time of algorithms incorporating GO on Krogan network.

Algorithm	Time (sec)	Complexity
LCDA-GO	47.05	$O(nlog(n))$
MTGO	54000	$O(kn^3)$

https://doi.org/10.1371/journal.pone.0260484.t006

time complexity than the state-of-the-art. LCDA-GO surpasses the state-of-the-art algorithms by performing on a log-linear time complexity, while recent algorithms such as MTGO run on polynomial time complexity.

One of the limitations of LCDA-GO is that it can only discover networks including one connected component. The algorithm relies on breadth-first search to discover the network, it thus could not converge if the network consists of more than one connected components. One solution to avoid this issue is to identify the connected components of the network before executing LCDA-GO and provide one node from each component as the input for the algorithm.

To extend our algorithm, we plan to evaluate LCDA-GO from functionality aspects. A GO term analysis could provide an evaluation on the significance of the functions within each community. Moreover, considering the various attributes utilized in PPI networks, we plan to analyze PPI networks from *attributed network* [48] prospect. We believe that the algorithm could expand for applications in the context of attributed networks.

Author Contributions

Conceptualization: Saharnaz Dilmaghani, Matthias R. Brust, Carlos H. C. Ribeiro.

Data curation: Saharnaz Dilmaghani.

Formal analysis: Saharnaz Dilmaghani, Carlos H. C. Ribeiro.

Funding acquisition: Pascal Bouvry.

Investigation: Saharnaz Dilmaghani.

Methodology: Saharnaz Dilmaghani.

Project administration: Matthias R. Brust, Pascal Bouvry.

Supervision: Matthias R. Brust, Pascal Bouvry.

Validation: Matthias R. Brust, Carlos H. C. Ribeiro, Grégoire Danoy.

Visualization: Saharnaz Dilmaghani.

Writing – original draft: Saharnaz Dilmaghani.

Writing – review & editing: Saharnaz Dilmaghani, Matthias R. Brust, Carlos H. C. Ribeiro, Emmanuel Kieffer, Grégoire Danoy.

References

1. Hartwell LH, Hopfield JJ, Leibler S, Murray AW. From molecular to modular cell biology. Nature. 1999; 402(6761):C47–C52. https://doi.org/10.1038/35011540 PMID: 10591225
2. Li X, Wu M, Kwoh CK, Ng SK. Computational approaches for detecting protein complexes from protein interaction networks: a survey. BMC genomics. 2010; 11(1):1–19. https://doi.org/10.1186/1471-2164-11-S1-S3 PMID: 20158874
3. Li M, Wu X, Wang J, Pan Y. Towards the identification of protein complexes and functional modules by integrating PPI network and gene expression data. BMC bioinformatics. 2012; 13(1):1–15. https://doi.org/10.1186/1471-2105-13-109 PMID: 22621308
4. Safari-Alighiarloo N, Taghizadeh M, Rezaei-Tavirani M, Goliaei B, Peyvandi AA. Protein-protein interaction networks (PPI) and complex diseases. Gastroenterology and Hepatology from bed to bench. 2014; 7(1):17. PMID: 25436094
5. Mujawar S, Mishra R, Pawar S, Gatherer D, Lahiri C. Delineating the plausible molecular vaccine candidates and drug targets of multidrug-resistant Acinetobacter baumannii. Frontiers in cellular and infection microbiology. 2019; 9:203. https://doi.org/10.3389/fcimb.2019.00203 PMID: 31281799
6. Ito T, Chiba T, Ozawa R, Yoshida M, Hattori M, Sakaki Y. A comprehensive two-hybrid analysis to explore the yeast protein interactome. Proceedings of the National Academy of Sciences. 2001; 98 (8):4569–4574. https://doi.org/10.1073/pnas.061034498 PMID: 11283351
7. Puig O, Caspary F, Rigaut G, Rutz B, Bouveret E, Bragado-Nilsson E, et al. The tandem affinity purification (TAP) method: a general procedure of protein complex purification. Methods. 2001; 24(3):218–229. https://doi.org/10.1006/meth.2001.1183 PMID: 11403571
8. Gavin AC, Bösche M, Krause R, Grandi P, Marzioch M, Bauer A, et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. Nature. 2002; 415(6868):141–147. https://doi.org/10.1038/415141a PMID: 11805826
9. Ji J, Zhang A, Liu C, Quan X, Liu Z. Survey: Functional module detection from protein-protein interaction networks. IEEE Transactions on Knowledge and Data Engineering. 2012; 26(2):261–277. https://doi.org/10.1109/TKDE.2012.225
10. Srihari S, Leong HW. A survey of computational methods for protein complex prediction from protein interaction networks. Journal of bioinformatics and computational biology. 2013; 11(02):1230002. https://doi.org/10.1142/S021972001230002X PMID: 23600810
11. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, et al. Gene ontology: tool for the unification of biology. Nature genetics. 2000; 25(1):25–29. https://doi.org/10.1038/75556 PMID: 10802651
12. Consortium GO. The gene ontology project in 2008. Nucleic acids research. 2008; 36(suppl_1):D440–D444. https://doi.org/10.1093/nar/gkm883

13. Milano M. Gene Prioritization Tools. 2019.

14. Barabási AL, Gulbahce N, Loscalzo J. Network medicine: a network-based approach to human disease. Nature reviews genetics. 2011; 12(1):56–68. https://doi.org/10.1038/nrg2918 PMID: 21164525

15. Muff S, Rao F, Caflisch A. Local modularity measure for network clusterizations. Physical Review E. 2005; 72(5):056107. https://doi.org/10.1103/PhysRevE.72.056107 PMID: 16383688

16. Spirin V, Mirny LA. Protein complexes and functional modules in molecular networks. Proceedings of the national Academy of sciences. 2003; 100(21):12123–12128. https://doi.org/10.1073/pnas.2032324100 PMID: 14517352

17. Li XL, Foo CS, Ng SK. Discovering protein complexes in dense reliable neighborhoods of protein interaction networks. In: Computational Systems Bioinformatics: (Volume 6). World Scientific; 2007. p. 157–168.

18. Porter MA, Onnela JP, Mucha PJ. Communities in networks. Notices of the AMS. 2009; 56(9).

19. Schaeffer SE. Graph clustering. Computer science review. 2007; 1(1):27–64. https://doi.org/10.1016/j.cosrev.2007.05.001

20. Fortunato S. Community detection in graphs. Physics reports. 2010; 486(3-5). https://doi.org/10.1016/j.physrep.2009.11.002

21. Pellegrini M. Community detection in biological networks. 2019.

22. Bhowmick SS, Seah BS. Clustering and summarizing protein-protein interaction networks: A survey. IEEE Transactions on Knowledge and Data Engineering. 2015; 28(3):638–658. https://doi.org/10.1109/TKDE.2015.2492559

23. Vella D, Marini S, Vitali F, Di Silvestre D, Mauri G, Bellazzi R. MTGO: PPI network analysis via topological and functional module identification. Scientific reports. 2018; 8(1):1–13. https://doi.org/10.1038/s41598-018-23672-0 PMID: 29615773

24. Dilmaghani S, Brust MR, Piyatumrong A, Danoy G, Bouvry P. Link definition ameliorating community detection in collaboration networks. Frontiers in Big Data. 2019; 2:22. https://doi.org/10.3389/fdata.2019.00022 PMID: 33693345

25. Rahiminejad S, Maurya MR, Subramaniam S. Topological and functional comparison of community detection algorithms in biological networks. BMC bioinformatics. 2019; 20(1):1–25. https://doi.org/10.1186/s12859-019-2746-0 PMID: 31029085

26. Guimera R, Amaral LAN. Cartography of complex networks: modules and universal roles. Journal of Statistical Mechanics: Theory and Experiment. 2005; 2005(02):P02001. https://doi.org/10.1088/1742-5468/2005/02/P02001 PMID: 18159217

27. Hu L, Chan KC. A density-based clustering approach for identifying overlapping protein complexes with functional preferences. BMC bioinformatics. 2015; 16(1):1–16. https://doi.org/10.1186/s12859-015-0583-3 PMID: 26013799

28. Zhang XF, Dai DQ, Ou-Yang L, Yan H. Detecting overlapping protein complexes based on a generative model with functional and topological properties. BMC bioinformatics. 2014; 15(1):1–15. https://doi.org/10.1186/1471-2105-15-186 PMID: 24928559

29. Dilmaghani S, Brust MR, Danoy G, Bouvry P. Local Community Detection Algorithm with Self-defining Source Nodes. In: International Conference on Complex Networks and Their Applications. Springer; 2020. p. 200–210.

30. Dilmaghani S, Brust MR, Danoy G, Bouvry P. Community Detection in Complex Networks: A Survey on Local Approaches. In: Intelligent Information and Database Systems. Cham: Springer International Publishing; 2021. p. 757–767.

31. Wu M, Li X, Kwoh CK. Algorithms for detecting protein complexes in PPI networks: an evaluation study. In: Proceedings of third IAPR international conference on pattern recognition in bioinformatics (PRIB 2008); 2008. p. 15–17.

32. Brohee S, Van Helden J. Evaluation of clustering algorithms for protein-protein interaction networks. BMC bioinformatics. 2006; 7(1):1–19. https://doi.org/10.1186/1471-2105-7-488 PMID: 17087821

33. Bader GD, Hogue CW. An automated method for finding molecular complexes in large protein interaction networks. BMC bioinformatics. 2003; 4(1):1–27. https://doi.org/10.1186/1471-2105-4-2 PMID: 12525261

34. vanDongen S. A cluster algorithm for graphs. Information Systems [INS]. 2000;(R 0010).

35. Nepusz T, Yu H, Paccanaro A. Detecting overlapping protein complexes in protein-protein interaction networks. Nature methods. 2012; 9(5):471. https://doi.org/10.1038/nmeth.1938 PMID: 22426491

36. Wu M, Li X, Kwoh CK, Ng SK. A core-attachment based method to detect protein complexes in PPI networks. BMC bioinformatics. 2009; 10(1):1–16. https://doi.org/10.1186/1471-2105-10-169 PMID: 19486541

37. Li XL, Foo CS, Tan SH, Ng SK. Interaction graph mining for protein complexes using local clique merging. Genome Informatics. 2005; 16(2):260–269. PMID: 16901108

38. Palla G, Derényi I, Farkas I, Vicsek T. Uncovering the overlapping community structure of complex networks in nature and society. nature. 2005; 435(7043):814–818. https://doi.org/10.1038/nature03607 PMID: 15944704

39. King AD, Pržulj N, Jurisica I. Protein complex prediction via cost-based clustering. Bioinformatics. 2004; 20(17):3013–3020. https://doi.org/10.1093/bioinformatics/bth351 PMID: 15180928

40. Krogan NJ, Cagney G, Yu H, Zhong G, Guo X, Ignatchenko A, et al. Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. Nature. 2006; 440(7084):637–643. https://doi.org/10.1038/nature04670 PMID: 16554755

41. Stark C, Breitkreutz BJ, Reguly T, Boucher L, Breitkreutz A, Tyers M. BioGRID: a general repository for interaction datasets. Nucleic acids research. 2006; 34(suppl_1):D535–D539. https://doi.org/10.1093/nar/gkj109 PMID: 16381927

42. Thomas PD, Kejariwal A, Campbell MJ, Mi H, Diemer K, Guo N, et al. PANTHER: a browsable database of gene products organized by biological function, using curated protein family and subfamily classification. Nucleic acids research. 2003; 31(1):334–341. https://doi.org/10.1093/nar/gkg115 PMID: 12520017

43. Pu S, Wong J, Turner B, Cho E, Wodak SJ. Up-to-date catalogues of yeast protein complexes. Nucleic acids research. 2009; 37(3):825–831. https://doi.org/10.1093/nar/gkn1005 PMID: 19095691

44. Geva G, Sharan R. Identification of protein complexes from co-immunoprecipitation data. Bioinformatics. 2011; 27(1):111–117. https://doi.org/10.1093/bioinformatics/btq652 PMID: 21115439

45. Dai Q, Guo M, Guo Y, Liu X, Liu Y, Teng Z. A least square method based model for identifying protein complexes in protein-protein interaction network. BioMed research international. 2014; 2014. https://doi.org/10.1155/2014/720960 PMID: 25405206

46. Maulik U, Mukhopadhyay A, Bhattacharyya M, Kaderali L, Brors B, Bandyopadhyay S, et al. Mining quasi-bicliques from HIV-1-human protein interaction network: a multiobjective biclustering approach. IEEE/ACM transactions on computational biology and bioinformatics. 2012; 10(2):423–435. https://doi.org/10.1109/TCBB.2012.139

47. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, et al. Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome research. 2003; 13(11):2498–2504. https://doi.org/10.1101/gr.1239303 PMID: 14597658

48. Interdonato R, Atzmueller M, Gaito S, Kanawati R, Largeron C, Sala A. Feature-rich networks: going beyond complex network topologies. Applied Network Science. 2019; 4(1):1–13. https://doi.org/10.1007/s41109-019-0111-x