ISSN: 0067-2904

Improving the Reliability of Evolutionary Algorithm for Complex Detection in Noisy Protein-Protein Interaction Networks

Safa Ahmed Abdulsahib, Bara'a Ali Attea
Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq.

Received: 5/9/2023 Accepted: 16/1/2024 Published: 30/1/2025

Abstract

Evolutionary algorithms are better than heuristic algorithms at finding protein complexes in protein-protein interaction networks (PPINs). Many of these algorithms depend on their standard frameworks, which are based on topology. Further, many of these algorithms have been exclusively examined on networks with only reliable interaction data. The main objective of this paper is to extend the design of the canonical and topological-based evolutionary algorithms suggested in the literature to cope with noisy PPINs. The design of the evolutionary algorithm is extended based on the functional domain of the proteins rather than on the topological domain of the PPIN. The gene ontology annotation in each molecular function, biological process, and cellular component is used to get the functional domain. The reliability of the proposed algorithm is examined against the algorithms proposed in the literature. To this end, a yeast protein-protein interaction dataset is used in the assessment of the final quality of the algorithms. To make fake negative controls of PPIs that are wrongly informed and are linked to the high-throughput interaction data, different noisy PPINs are created. The noisy PPINs are synthesized with a different and increasing percentage of misinformed PPIs. The results confirm the effectiveness of the extended evolutionary algorithm design to utilize the biological knowledge of the gene ontology. Feeding EA design with GO annotation data improves reliability and produces more accurate detection results than the counterpart algorithms.

Keywords: Complex detection, Evolutionary algorithm, Missing PPI, Modularity, Protein–protein interaction, Unreliable PPI.

تحسين موثوقية الخوارزمية التطورية للكشف عن المركبات البروتينية في شبكات التفاعل البروتينية الضوضائية

صفا أحمد عبدالصاحب، براء علي عطية

قسم علوم الحاسوب ، كلية العلوم ، جامعة بغداد ، بغداد ، العراق

*Email: safa.a@sc.uobaghdad.edu.iq

الخلاصة:

إن أحد التحديات الرئيسية التي تستنفد بشكل حاسم الموثوقية في تحليل شبكات تفاعل البروتين البروتين (PPINs) هو الضوابط السلبية للتجارب واسعة النطاق التي تُرجع العديد من الإيجابيات الكاذبة (أي التفاعلات غير الموثوقة أو الزائفة) والسلبيات الكاذبة (أي عدم وجود التفاعلات). تزيد مؤشرات PPI المضللة بشكل كبير من التعقيد الطوبولوجي للشبكات ، مما يؤدي إلى نتائج غير موثوقة لخوارزميات الكشف الطوبولوجية. بالإضافة إلى نجاح الخوارزميات التطورية على خوارزميات الكشف عن مجريات الأمور لاكتشاف مجمعات البروتين في PPINs ، تعتمد العديد من هذه الخوارزميات على ما يبدو على أطرها الأساسية ذات المكونات الطوبولوجية. علاوة على ذلك ، تم فحص العديد من هذه الخوارزميات حصريًا على شبكات ذات بيانات تفاعل موثوقة فقط. الهدف الرئيسي من هذا البحث هو توسيع تصميم الخوارزميات التطورية النمطية والقائمة على الطوبولوجيا المقترحة في الأدبيات للتعامل مع الشبكات البروتينية الضوضائية. تم توسيع تصميم الخوارزمية التطورية بناءً على المجال الوظيفي للبروتينات بدلاً من المجال الطوبولوجي لـ PPIN. يُشتق المجال الوظيفي من شرح علم الوجود الجيني في كل من الوظيفة الجزيئية والعملية البيولوجية والمكون الخلوي. يتم فحص موثوقية الخوارزمية المقترحة مقابل الخوارزميات المقترحة في الأدبيات. تحقيقا لهذه الغاية ، يتم استعمال مجموعة بيانات تفاعل بروتين البروتين الخميرة في تقييم الجودة النهائية للخوارزميات. يتم إنشاء شبكات بروتينية ضوضائية مختلفة لمحاكاة الضوابط السلبية لمؤشرات PPI المضللة المرتبطة ببيانات التفاعل عالي الإنتاجية. يتم تصنيع هذه الشبكات البروتينية الضوضائية مع نسبة مختلفة ومتزايدة من معلومات مضللة للPPIs. تؤكد النتائج فعالية تصميم الخوارزمية التطورية الموسعة للاستفادة من المعرفة البيولوجية لعلم الوجود الجيني. يؤدي تزويد تصميم EA بمعلومات البيولوجية لـ GO إلى تحسين الموثوقية ويجعل نتائج الكشف أكثر دقة من الخوارزميات المناظرة.

1. Introduction

Networked systems tend to organize nodes into cohesive modules or communities, but identifying these communities is a challenging task in network research with broad applications in biological networks, social network modeling, and communication pattern analysis [1–7]. Protein-protein molecular interactions (PPIs) in every organism are regularly organized as networks, noted as protein-protein interaction networks (PPINs). PPINs make it possible for graph theory and network topology to reveal and study the hidden details, like functional modules or complexes connected with how cells are organized, how processes work, and how the networks in these organisms do their jobs.

A protein complex is defined as a group of proteins that work together to carry out a specific biological process or activity. For example, in Y*east Saccharomyces Cerevisiae* PPIN (depicted in Figure 1), there are 990 distinct proteins being connected with 4687 different interactions. Based on the Munich Information Center for Protein Sequences (MIPS) golden reference set, the proteins in this PPIN are structured with 78 uncoupled complexes [8]. In Figure 1, complex C_{48} with 13 proteins and their interconnections is zoomed out. Figure 2 depicts the names of the 13 proteins and their interconnections. Note that not all biological processes are connected, and interactions between proteins can be classified as within-complex (intra-connection) or between-complexes (inter-connection), as shown in Figure 3.

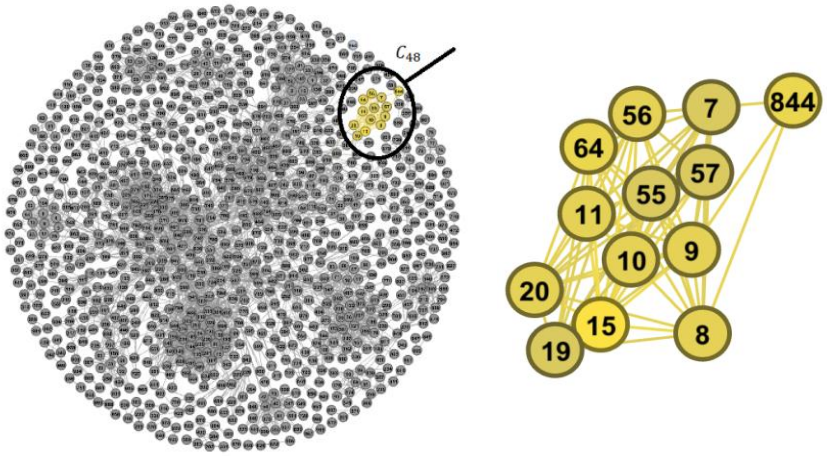


Figure 1: The y*east Saccharomyces cerevisiae* network (left) and one complex (C_{48}) is zoomed out in the right

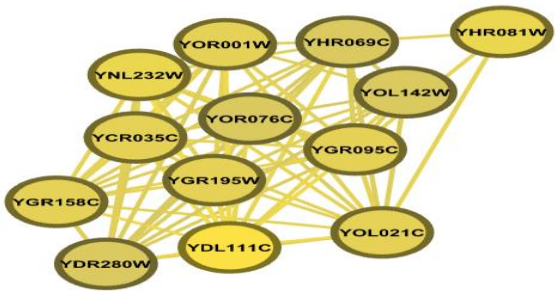


Figure 2: An illustrative example of complex (C_{48}) from yeast Saccharomyces cerevisiae PPIN with 13 proteins (depicted with their identity names) and their intra connections

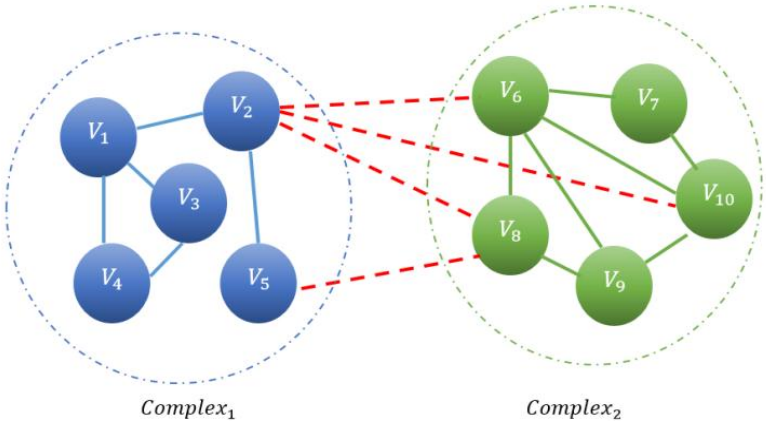


Figure 3: A small PPIN of 10 proteins being decomposed into two complexes. The nodes within a dashed circle form one complex. The edges inside the dashed circle are intra-connections, while those connecting the two separate complexes are inter-connections.

Detecting protein complexes from a PPIN is proven to be a non-deterministic polynomial-time hard (NP-hard) problem, which makes it computationally difficult to solve. Also, one of the biggest problems with studying protein-protein interaction networks (PPINs) is that large-scale experiments often give back a lot of false positives (i.e., interactions that aren't real or aren't expected) and false negatives (i.e., interactions that aren't there). Such misinformed interactions considerably increase the topological complexity of the networks, thus making the results unreliable for complex detection algorithms.

The main goal of this paper is to look into how well evolutionary-based complex detection algorithms (ECDs) can find things when there are noisy interactions between proteins. Different types of noise are simulated in the experiments. To this end, several noisy PPINs are synthesized from the well-known yeast *Saccharomyces cerevisiae* PPIN. Also, the proposed ECD doesn't just use topological-based parts; it also uses information from the functional domain of proteins. This comes from Gene Ontology Annotations (GOA) in Molecular Function (MF), Biological Process (BP), and Cellular Component (CC). The performance of the proposed Gene Ontology-based ECD (GO-based ECD) is compared against the performance of two state-of-the-art ECDs. These are the canonical ECD of Pizzuti and Rombo in [9, 10] and the topological-based ECD of Attea and Abdullah [11]. The findings in this paper show how important it is to include gene ontology information when designing ECD because it makes detection much more reliable compared to the canonical and topological frameworks of ECD.

The remaining sections of this paper are organized as follows: The main ECD approaches proposed in the literature are mentioned in the next section. A brief presentation of the foundational ideas relating to this study follows. Problem formulation and algorithm design are then presented in detail. Results and performance evaluations are reported in Section 5. Finally, the paper is closed with a conclusion in Section 6.

2. Related works

The literature encompasses different complex detection methods based on meta-heuristic algorithms, mainly evolutionary algorithms (EAs). The EA-based complex detection methods are proved to be more reliable than their counterpart local-based complex detection methods such as Molecular Complex Detection (MCODE) [3], Purification of the bait proteins [4], Dense-neighborhood Extraction using Connectivity and conFidence Features (DECAFF) [5], Repeated Random Walk (RRW) [6], Clustering-based on maximal cliques (CMC) [7], and Hierarchical Link Clustering [7, 12].

Evolutionary-based complex detection algorithms use evolutionary principles, i.e., natural selection and genetic variation, to search for promising candidates for protein complex structures. Pizzuti and Rombo proposed one of the earliest evolutionary-based complex detection algorithms in [9] and [10]. They developed a single-objective genetic algorithm (GA) with different single-objective models to solve the problem. The remaining components of their algorithm (i.e., selection, crossover, and mutation operators) were designed based on their canonical forms. All their objective function models were defined based on different topological characteristics of the proteins and their interactions in the networks. The formulation of the objective functions includes the well-known modularity (Q) function, community score (CS) function, conductance (CO) function, normalized cut (NC) function, internal density (ID) function, expansion (EX)

function, and cut ratio (CR) function. Unlike the modularity (Q) function, all the remaining models explicitly define both the intra-complex structure and the inter-complex structure with different maximization or minimization scores. Traditional modularity, on the other hand, explicitly defines the intra-complex structure score only.

Bandyopadhyay et al. [13] and Ray et al. [14] were the first to formulate the problem as a multi-objective optimization (MOO) problem. Both intra-complex structure and inter-complex structure are reflected in their MOO model. To solve the issue, they created a multi-objective genetic algorithm according to the well-known non-dominated sorting algorithm (NSGA-II).

In [11], two contradictory topological-based intra- and inter-structures were formulated as a multi-objective optimization model. The well-known decomposition-based multi-objective evolutionary algorithm (MOEA/D) served as the frame for the adopted multi-objective evolutionary algorithm.

In [15], a locally-assisted migration operator is proposed based on the topological properties of the tested PPINs. The operator has the ability to improve the performance of both single-objective and multi-objective evolutionary-based complex detection algorithms. These evolutionary-based algorithms have proven to be more robust than heuristic algorithms, potentially providing better accuracy and scalability for complex detection in large biological networks.

Significant exploitation of domain knowledge of the optimization problems can support the use of EAs to the fullest. Unfortunately, there is a lack of research investigating these evolutionary-based algorithms to examine the impact of domain knowledge on their design. In bioinformatics, the utilization of ontologies for genome annotation has brought significant advances to the field of molecular biology. These bio-ontologies were rarely considered in the design of evolutionary-based complex detection algorithms. A few months ago, Abdulateef et al. [16] looked at how to design the mutation operator in the EA (with modularity model) using biological information from three different gene sub-ontology types. They designed the mutation operator based on protein pair similarity in four versions: molecular function (MF), cellular component (CC), biological process (BP), and their combinations.

3. Background

3.1 Interactome and interaction graph

The interactome refers to the set of all the molecular interactions within cells, especially protein-protein physical interactions. It's a global description obtained through various methods to estimate the entire biological network of protein interactions in an organism [17]. For example, the interactome of Saccharomyces cerevisiae was estimated to be on the order of 20,000 interactions. However, larger estimates include indirect or predicted interactions from affinity purification/mass spectrometry (AP/MS) studies.

Mathematically, PPIN is represented as an undirected interaction graph, $\mathcal{N}(P, E)$, where $P = \{p_1, p_2, \dots, p_n\}$ represents a set of n proteins and $E = \{e_1, e_2, \dots, e_m\}$ represents a set of m pairwise interactions. To represent the finite graph of $\mathcal{N}$, a square binary symmetric matrix, $A = \left[a_{ij}\right]^{n \times n}$ is normally used. If proteins p_i and p_j interact (i.e., adjacent), both entries a_{ij} and a_{ji} of A are non-zeros; otherwise, both entries are assigned zeros. Further, the diagonal entries of

the adjacency matrix are assigned zeros. Also, for each protein p_i, $\sum_{j=1}^{n}(a_{ij})$ is said to be the degree d_i of p_i, while $\sum_{i=1}^{n}\sum_{j=1}^{n}(a_{ij})$ is the whole volume of the network. Figure 4 depicts an illustrative example of ten PPIs from the yeast *Saccharomyces cerevisiae* PPIN (depicted in Figure 3). In Figure 3, a total of 16 interactions out of 4687 interactions are mapped to their corresponding adjacency matrix. In other words, 32 entries in the adjacency matrix are set to 1. Thus, for the whole yeast PPIN network, there should be 4687×2 entries set to 1 in the counterpart adjacency matrix.

Mathematically, a complex detection problem means to decompose the adjacency matrix *A* into a priori unknown number (K) of varying sized sub-matrices. The space Ω of all possible decomposition solutions determines the complexity of the problem. There is no deterministic rule to decompose the adjacency matrix A, however, any complex detection algorithm attempts to figure out the structure of the complex set $C = \{c_1, c_2, \dots, c_K\}$ following the general rule of dense and sparse connectivity features. It is widely assumed that a protein $p_i \in c_k$ should have more internal connections $in(p_i)$ than external connections $out(p_i)$. Formally speaking, $in(p_i) = \sum_{p_j \in c_k} a_{ij}$ and $out(p_i) = \sum_{p_j \notin c_k} a_{ij}$ express, respectively, the number of intra-connections and inter-connections of node p_i belongs to cluster c_k. In other words, $d_i = in(p_i) + out(p_i)$.

Table 1: Adjacency matrix for a small PPIN of 10 proteins from the whole PPIN in Figure 3. "1" indicates that the corresponding pair of proteins physically interacts, otherwise, "0" means no biological interaction. All diagonal entries are set to "0"

$\boldsymbol{A} =$

	$\boldsymbol{v_1}$	$\boldsymbol{v_2}$	$\boldsymbol{v_3}$	$\boldsymbol{v_4}$	$\boldsymbol{v_5}$	$\boldsymbol{v_6}$	$\boldsymbol{v_7}$	$\boldsymbol{v_8}$	$\boldsymbol{v_9}$	$\boldsymbol{v_{10}}$
v_1	0	1	1	1	0	0	0	0	0	0
v_2	1	0	0	0	1	1	0	1	0	1
v_3	1	0	0	1	0	0	0	0	0	0
v_4	1	0	1	0	0	0	0	0	0	0
v_5	0	1	0	0	0	0	0	1	0	0
v_6	0	1	0	0	0	0	1	1	1	1
v_7	0	0	0	0	0	1	0	0	0	1
v_8	0	1	0	0	1	1	0	0	1	0
v_9	0	0	0	0	0	1	0	1	0	1
v_{10}	0	1	0	0	0	1	1	0	1	0

3.2 Annotation of proteins with gene ontology

Gene ontology (GO) is an active species-agnostic ontology used in biology to describe the semantics or context of gene and gene product attributes in single and multicellular organisms. As the activity or function of a protein is defined at different levels, the GO domain has been decomposed into three orthogonal categories or aspects: *molecular function* (MF), *biological process* (BP), and *cellular component* (CC). Each protein performs elementary molecular-level activities that are normally independent of the environment and occur at the molecular level, such as catalytic, transport, or binding activities. Larger cellular processes or biological programs are accomplished by multiple molecular activities of sets of interacted proteins [18].

Every GO term has a distinct seven-digit identifier that begins with the letters GO (for example, GO: 0003714). As an illustrative example, consider Table 1, where the annotations of five different proteins with their direct GO terms are reported. The annotations are reported in their MF, BP, and CC sub-ontology terms. The GO terms were downloaded online from the Saccharomyces Genome Database (SGD) at the following URL: http://genome-www.stanford.edu/Saccharomyces/.

Table 2: A sample of yeast proteins with their identity numbers, identity names, and direct GO annotation with MF, BP, and CC sub-ontology terms

Protein		GO term		
#	**name**	**BP**	**CC**	**MF**
82	**'YHR200W'**	[GO:0006511, GO:00 43248, GO:00 43161]	[GO:0000502, GO:0008540, GO:0005829, GO:0005634]	[GO: 0036435, GO:00 31593]
41	**'YDL147W'**	[GO:0000338]	[GO:0000502, GO:0008180, GO:0008541, GO:0034515, GO:0005737, GO:0008541, GO:0031595]	[GO:0005515]
178	**'YIL075C'**	[GO:0006511, GO:0043248, GO:0042176, GO:0050790]	[GO:0005634, GO:0008540, GO:0034515, GO:0000502]	[GO:0004175, GO:0031625, GO:0030234]
434	**'YER094C'**	[GO:0010498, GO:0010499, GO:0043161, GO:0006508, GO:0051603]	[GO:0019774, GO:0005634, GO:0005789, GO:0019774, GO:0034515, GO:0005634, GO:0005737, GO:0000502, GO:0005839, GO:0019774]	[GO:0061133]
274	**'YJL001W'**	[GO:0010498, GO:0010499, GO:0043161, GO:0006508, GO:0051603]	[GO:0019774, GO:0005634, GO:0005789, GO:0034515, GO:0005737, GO:0000502, GO:0005634, GO:0005839]	[GO:0004175, GO:0004298, GO:0016787, GO:0008233, GO:0004298]
308	**'YOL038W'**	[GO:0010499, GO:0043161, GO:0006511, GO:0051603, GO:0005737]	[GO:0005634, GO:0005739, GO:0019773, GO:0034515, GO:0042175, GO:0005737, GO:0000502, GO:0005839]	[GO:0003674, GO:0004298, GO:0004175]

Each GO term (t) can be structured hierarchically by a directed acyclic graph (DAG), where each GO term is a node, and the relationships between the terms are edges between the nodes. Child GO terms are more specialized than their parent GO terms, and a GO term may

have more than one parent GO term. A relation between two terms (t_1, t_2) is represented as a directed edge pointing from t_2 to t_1. There are three main types of directed relationships between GO terms. These are 'is_a', '$part\_of$', and '$regulate$' [18]. A straightforward class-subclass relation is called is_a, where t_1 is_a t_2 denotes that GO term t_1 is a subclass of GO term t_1. A partial ownership relation is a $part\_of$ where t_3 $part\_of$ t_4 means that whenever t_3 is present, it is always a part of t_4, but t_3 is not required to be present. The relation '$regulate$' describes a case in which one process directly affects the manifestation of another process or quality, i.e., the former *regulates* the latter. Figure 4, depicts the DAG for cytoplasm (GO:0005737). This GO term has two parents: it is_a cellular anatomical entity, and it is $part\_of$ the intracellular anatomical structure.

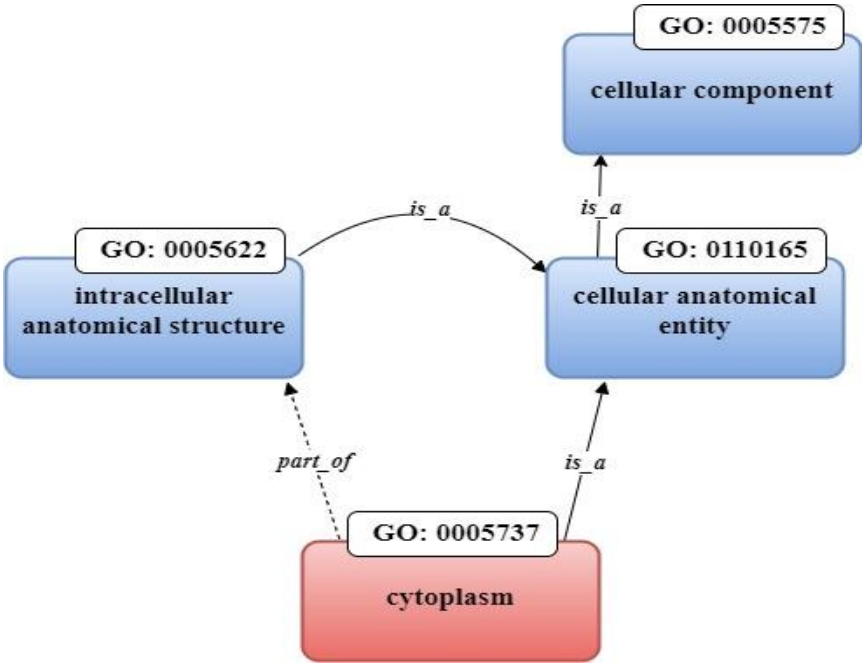


Figure 4: Graph-based representation for GO terms and relations

Proteins or gene products are, then, annotated with GO terms either directly or via inheritance, which implies annotation to all of their ancestor t_s terms in $DAG(t)$. An ancestor set, $Anc(t)$, for some t is defined as:

$$Anc(t): DAG \rightarrow \{t_s | \exists\ path\ (t_s, t)\} \quad (1)$$

As an illustrative example, consider the three DAGs for three GO terms for protein "$YPL139C$ " in Figure 5. The GO terms are: $\mathcal{MF}$ ($GO: 0003714$), $\mathcal{BP}$ ($GO: 0051321$), and $\mathcal{CC}$ ($GO: 005634$). For example, the DAG for $GO: 0051321$ (meiotic cell cycle) has six terms connected with six 'is_a' relations and one '$part\_of$' relation. The term $GO: 0022414$ (reproductive process) is considered as is_a subclass of $GO: 0008150$ (biological process) and also a $part\_of$ $GO: 0000003$ (reproduction).

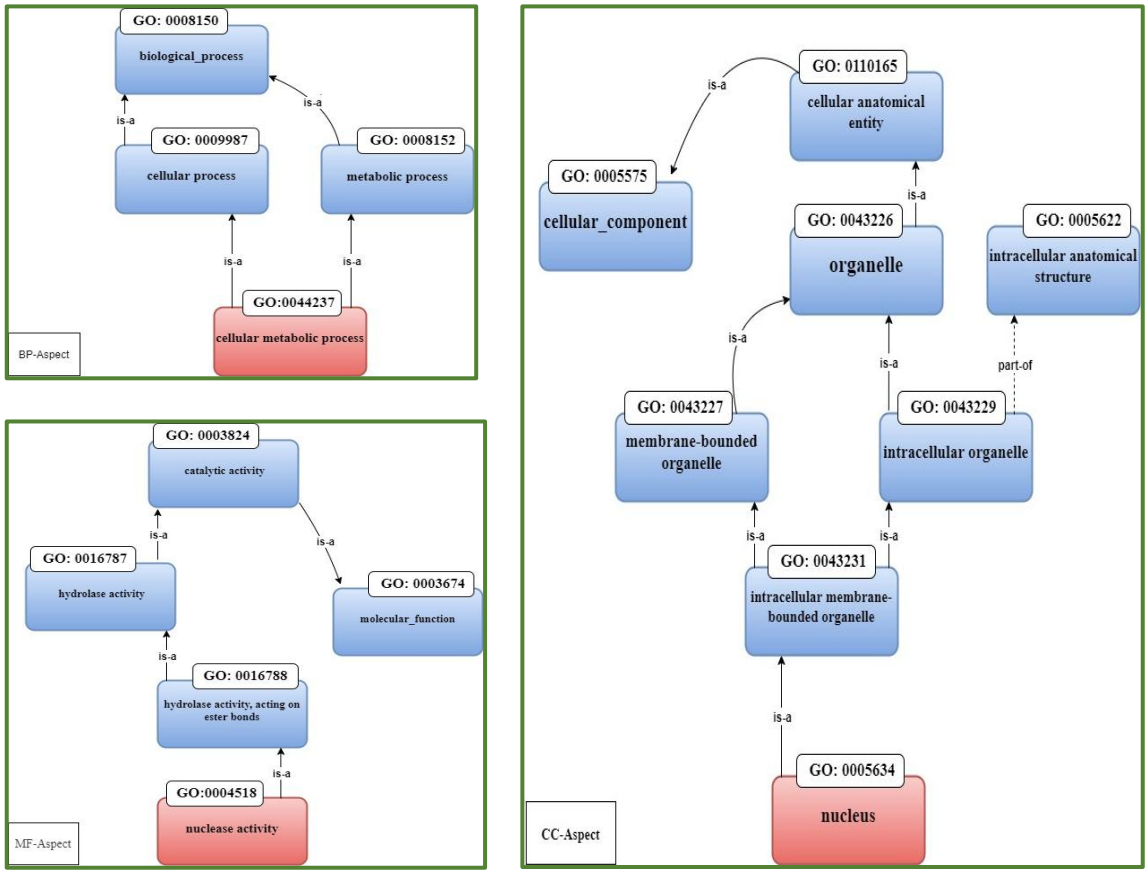


Figure 5: Three DAGs for three different GO terms for the protein "YPL139C". One MF term (GO: 0003714) (top left), one BP term (GO: 0051321), and one CC term (GO: 005634) (right). Solid arrows represent 'is_a' relations, while dashed arrows represent 'part_of ' relations

3.2.1 GO-based semantic similarity

Gene-ontology-based semantic similarity ($\mathcal{SS}$) gives the opportunity to compare GO terms or entities annotated with GO terms based on their semantic properties, normally acquired from corpora. From $\mathcal{SS}$, a semantic similarity matrix $\mathcal{S} = [\mathcal{SS}]^{N\times N}$is obtained for N GO terms that annotate n different proteins, where $\mathcal{SS}_{ij} = \mathcal{SS}_{ji} \in \mathcal{R}^+$ is the semantic similarity between terms t_i and t_j.

Based on the meaning of semantic value and semantic contribution, Wang et al. [19] proposed one of the well-known semantic similarity measures. The semantic value $\mathcal{S}(t)$: $DAG(t) \rightarrow \mathcal{R}^+$ for a GO term t is computed as the sum of the semantic contributions ($\mathcal{SC}$) of all GO terms in $DAG(t)$ to term t, $\mathcal{SC}$: $t \times t_s \times DAG(t) \rightarrow \mathcal{R}^+$, along the best (i.e., maximum) weighted paths to t. Note that the semantic contribution of the term t in its DAG to itself is 1, i.e., $\mathcal{SC}\big(t, DAG(t)\big) = 1$. The best weighted path for each ancestor is the path that has the maximum product of the weights on its edges. Wang et al. [19] set $w = 0.8$ and $w = 0.6$ for 'is_a' and 'part_of,' respectively. The formulation in Eq. 2 expresses the semantic contribution of term t_s to term t in $DAG(t)$. The formulation reveals that terms t_s that are closer to t in $DAG(t)$

contribute more to its semantics, whereas terms t_s that are farther from t in $DAG(t)$ contribute less as they are more general terms [19].

$$\mathcal{SC}\big(t_s, DAG(t)\big) = \max\{w \times \mathcal{SC}(t'_s) | \exists\ e(t_s, t'_s)\} \quad (2)$$

where the directed relation between t_s and t'_s is denoted by the expression $e(t_s, t'_s)$. Then, the semantic value of the term t in its DAG is expressed in Eq. 3.

$$\mathcal{S}(t) = \sum_{t_i \in DAG(t)} \mathcal{SC}(t_i, DAG(t)) \quad (3)$$

Then, the semantic similarity between two GO terms, t and t (as expressed in Eq. 4), is defined as the ratio of the semantic contributions of all common terms (also known as intersecting terms) in the $DAGs$ of, t_1 and t to the semantic values of t_1 and t_2, respectively .

$$\mathcal{SS}(t_1, t_2) = \frac{\sum_{t \in DAG(t_1) \cap DAG(t_2)} \mathcal{SC}\big(t, DAG(t_1)\big) + \mathcal{SC}\big(t, DAG(t_2)\big)}{\mathcal{S}(t_1) + \mathcal{S}(t_2)} \quad (4)$$

3.2.2 Gene functional similarity

Functional similarity ($\mathcal{FS}$) measures the degree to which two proteins share functional properties [20]. For n different proteins, then, a functional-based similarity matrix $\mathcal{FS} = [\mathcal{FS}_{ij}]^{n \times n}$ can be derived. For a pair of proteins, $\mathcal{FS}$ requires two sets of protein-level annotation, i.e., GO terms. Protein-term ($\mathcal{T}_\mathcal{P}$) representation can be established at two different levels: 1) *the direct annotation scheme* and 2) *the indirect annotation scheme.* In the direct annotation scheme, proteins are annotated using their direct GO terms across all three sub-ontology types. In other words, $\mathcal{T}_\mathcal{P} = \{\mathcal{MF}, \mathcal{BP}, \mathcal{CC}\}$. One of the well-known methods is Jaccard [20], as defined in Eq. 5.

$$\mathcal{FS}_{Jaccard}(\mathcal{P}_1, \mathcal{P}_2) = \frac{|\mathcal{T}_{\mathcal{P}1} \cap \mathcal{T}_{\mathcal{P}2}|}{|\mathcal{T}_{\mathcal{P}1} \cup \mathcal{T}_{\mathcal{P}2}|} \quad (5)$$

For indirect annotation, each protein is annotated according to its direct GO terms ($\mathcal{T}_\mathcal{P}$) and their ancestors in their corresponding DAG structures, i.e., $\mathcal{T}_\mathcal{P} \cup \mathcal{T}_{t|t \in \mathcal{T}_\mathcal{P}}$, where $\mathcal{T}_t = t \cup \{t_s\}$ indicates that the term t and all of its ancestors. They statistically consider a combination of the semantic similarities between the terms $\mathcal{T}_{\mathcal{P}1}$ and $\mathcal{T}_{\mathcal{P}2}$ to determine $\mathcal{FS}$ between two gene products $\mathcal{P}_1$ and $\mathcal{P}_2$. For example, maximum functional similarity [20] is defined in Eq. 6.

$$\mathcal{FS}_{Max}(\mathcal{P}_1, \mathcal{P}_2) = \max[\mathcal{SS}(t_1, t_2)]\ | t_1 \in \mathcal{T}_{\mathcal{P}1}, t_2 \in \mathcal{T}_{\mathcal{P}2} \quad (6)$$

4. Problem formulation and algorithm design

4.1 Synthesizing noisy PPINs

The success of accurate protein complex detection depends on the availability of high-quality benchmarks. High-throughput experimental techniques typically produce a rich source of experimentally detected PPI datasets; however, these PPIs are susceptible to noise (i.e., spurious interactions that do not exist) and incompleteness (i.e., missing interactions). This can arise due to a variety of factors, such as experimental limitations, technical errors, or even intentional attempts to manipulate the data.

In this paper, to simulate negative controls, different noisy PPINs were generated by perturbing the yeast Saccharomyces cerevisiae (S. cerevisiae) dataset. These synthesized PPINs were generated with varying percentages of misinformed PPIs by randomly adding or removing

an increasing percentage of interactions from the original PPIN. To follow different noise levels in the synthesized PPINs, five increasing percentages (from 10% to 50%) of interactions were randomly added or removed from the network. The interactions are divided into two types with respect to the interacted proteins: those interactions connecting proteins with the highest degrees (weighty proteins) and those interactions connecting proteins with the lowest degrees (light proteins). Proteins were classified first by Zaki et al. [21] according to the number of interactions they contribute. They categorize hub proteins into two main categories: genuine hub proteins and noisy proteins. Hub proteins are those highly connected proteins within complexes and those essential proteins with few connections but inside too small complexes. Genuine hub proteins have critical roles in mediating cellular processes. They showed that understanding the protein complex structure correctly is heavily augmented by differentiating genuine hub proteins from noisy proteins.

In this paper, three different forms of perturbation are used to add fake interactions or remove valid interactions. These are perturbing weighty proteins, perturbing light proteins, and perturbing random proteins. Here, a weighty protein is defined as having a degree greater than average in the PPIN, whereas a light protein is defined as having a degree less than average in the PPIN. Thus, a noisy PPIN was generated by adding a percentage (10%–50%) of fake interactions to either weighty proteins, light proteins, or random proteins. Similarly, a noisy PPIN can be obtained by removing 10%–50% interactions from either weighty proteins, light proteins, or random proteins.

Two illustrative examples demonstrate how two light proteins from the yeast Saccharomyces cerevisiae network (shown in Figure 1) are perturbed with noisy information. The PPIN in Figure 1 is perturbed with 10% noise. Protein #844 (YDR311') has only three interactions, as depicted in Figure 6 (left), which are perturbed after adding 10% fake interactions to the whole PPIN. Adding fake interactions leaves protein #844 (YDR311W) with 4 additional interactions, as depicted in Figure 6 (right). Figure 7, on the other hand, depicts another illustrative example while deleting true interactions from protein #618 (YOL090W). Protein #618 (YOL090W) has 6 true interactions (left). The 10% noise perturbation leaves this protein with only 4 interactions, as depicted in Figure 7 (right).

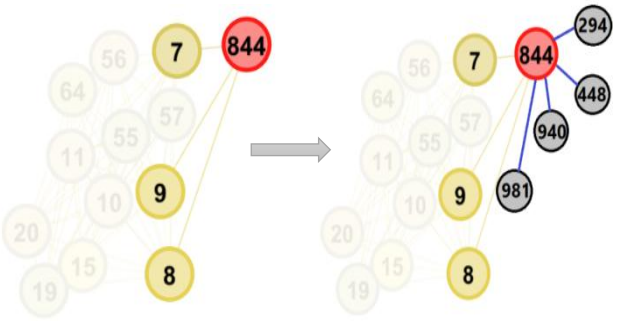


Figure 6: Adding fake interactions to protein #844 (YDR311W)

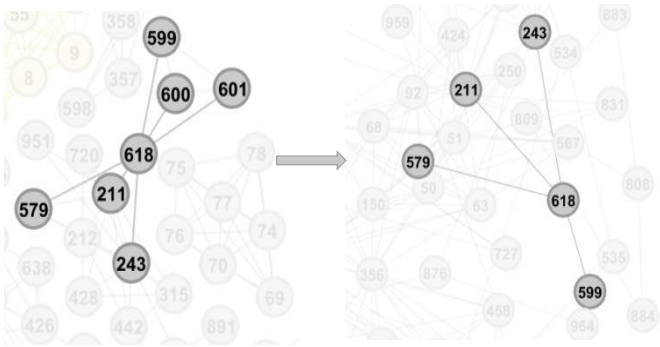


Figure 7: Deleting true interactions from protein #618 (YOL090W)

Also, Figure 8 shows how adding or removing 50% noise from the whole yeast Saccharomyces cerevisiae PPIN (in Figure 1) changes three different types of proteins. The selected proteins from the PPIN are one weighty protein #38 (YLR033W), one light protein #503 (YFL049W), and one random protein #179 (YPL016W).

Protein type	Selected yeast protein	50% spurious interactions	50% missing interactions
Weighty protein			
Light protein			
Random protein			

Figure 8: Adding fake interactions to or removing actual interactions from protein #38 (YLR033W), protein #503 (YFL049W), and protein #179 (YPL016W)

4.2 Algorithm design

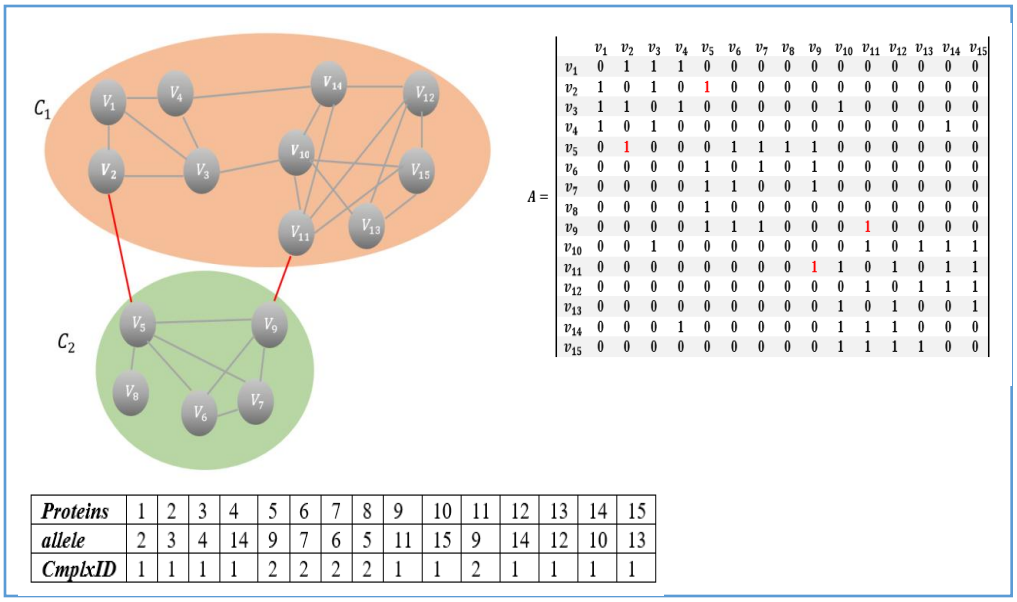
To make a strong evolutionary algorithm (EA) that can solve a certain real-life optimization problem, one of the most important things is to create a heuristic-based evolutionary operator that works with the EA framework [23]. Several studies followed this rule in designing competitive EAs for solving different NP-hard problems [24–27].

The general framework for the proposed evolutionary-based complex detection (ECD) algorithm $ECD: \mathbf{I} \rightarrow \mathbf{I}$ is an iterated transformation function that starts with an initial population $\mathbf{I} = \{I_1, I_2, \dots, I_{Psize}\}$ of genotype/phenotype solutions. The genotype of these solutions is generated randomly from the whole search space of the problem Ω. The encoding scheme is a locus-based adjacency representation, where each locus of an individual $I_{1 \leq i \leq n} \in \mathbf{I}$ corresponds to a protein j and its allele value represents a neighbor protein with which it can coexist in the same complex. The global locus-based initialization process designates the direct neighbors of j in A, i.e., $a_{jj'} = 1$, to be the possible allele values at locus j. The phenotype solution corresponding to a set of K complexes is mapped by a decoding function $\Gamma: I \rightarrow C$ (where $C = \{c_1, c_2, \dots, c_K\}$) applied to each individual. By this, Γ assigns the intra- and inter-relationships among the whole set of connections in A. Note that for any two solutions I_i and I_j in the population **I,** K_i and K_j do not necessitate to be equivalent. In other words, their phenotype solutions $C_i = \{c_1, c_2, \dots, c_{K_i}\}$ and $C_j = \{c_1, c_2, \dots, c_{K_j}\}$ could be dissimilar. Figure 9 depicts an illustrative example of two different genotype solutions and their corresponding phenotype solutions for a small PPIN with 15 yeast proteins from the yeast Saccharomyces cerevisiae PPIN (in Figure 9). The genotype is depicted as three vectors. The 1st vector lists protein labels, while the 2nd assigns the neighbors as alleles for the corresponding protein labels, and the 3rd vector maps proteins with their neighbors to complexes. Note that the two genotype solutions in Figure 9 are decoded into two different solutions with, respectively, two and three complexes. This also revises where the intra-connections and the inter-connections are to be in the adjacency matrix (as clarified in Figure 9 with black ones and red ones for, respectively, the intra-connections and the inter-connections of the two solutions).

The quantitative function modularity density (QD in Eq. 7) is adopted as an objective function to quantitatively measure the quality of the generated solutions. The model of QD [22] is defined as the sum of the averaged density of the sub-graphs that constitute the whole graph structure. In each sub-graph, the density is measured as the difference between the intra- and inter-degrees proportioned to the size of the sub-graph, and it is formulated by:

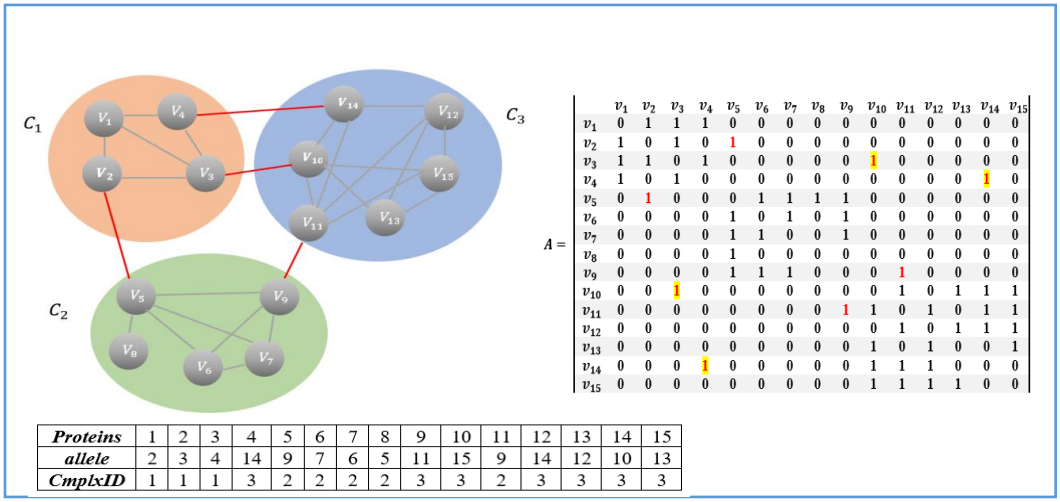
$$QD = \sum_{k=1}^{K} \frac{L(c_k, c_k) - L(c_k, c_{k' \neq k})}{|c_k|} \quad (7)$$

where for a set of K complexes $\mathcal{C} = \{c_1, c_2, \dots, c_K\}$, the numerator expresses the difference between two terms. The first term is the inner degree of a community c_k, which is twice the number of edges in c_k divided by the number of nodes in the complex c_k, The second term is the outer degree of c_k, which is the number of edges between nodes in c_k and other nodes in $c_{k' \neq k}$. The denominator expresses the number of nodes in c_k.



v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}
v_1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
v_2	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
v_3	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0
v_4	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0
v_5	0	1	0	0	0	1	1	1	1	0	0	0	0	0	0
v_6	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
v_7	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0
v_8	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
v_9	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0
v_{10}	0	0	1	0	0	0	0	0	0	0	1	0	1	1	1
v_{11}	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1
v_{12}	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
v_{13}	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1
v_{14}	0	0	0	1	0	0	0	0	0	1	1	1	0	0	0
v_{15}	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0

Proteins	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
allele	2	3	4	14	9	7	6	5	11	15	9	14	12	10	13
CmplxID	1	1	1	1	2	2	2	2	1	1	2	1	1	1	1



v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}
v_1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
v_2	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
v_3	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0
v_4	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0
v_5	0	1	0	0	0	1	1	1	1	0	0	0	0	0	0
v_6	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
v_7	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0
v_8	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
v_9	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0
v_{10}	0	0	1	0	0	0	0	0	0	0	1	0	1	1	1
v_{11}	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1
v_{12}	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
v_{13}	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1
v_{14}	0	0	0	1	0	0	0	0	0	1	1	1	0	0	0
v_{15}	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0

Proteins	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
allele	2	3	4	14	9	7	6	5	11	15	9	14	12	10	13
CmplxID	1	1	1	3	2	2	2	2	3	3	2	3	3	3	3

Figure 9: Two genotypes and their corresponding phenotype solutions for a small PPIN with 15 proteins from the *yeast Saccharomyces cerevisiae*

Based on the quality values of the solutions, parent solutions are then selected using binary tournament selection $\Phi_s: (I_1, \Theta_1) \times (I_2, \Theta_2) \rightarrow I$, where Θ is the quality value of the solution computed by QD. The selection operator prepares a pool of $Psize$ parents. For a pair of parent solutions I_1 and I_2, uniform crossover operator ($\Phi_\times: I_1 \times I_2 \times p_\times \rightarrow I$) is adopted to evenly mix their n decision making parameters. Crossover occurs to the parent's pair if the probability of crossing p is greater than the probability of crossover, $p_\times$. Here, $p_\times$is set to 0.8.
$\forall i \in \{1,2,\dots,N\} \wedge \forall j \in \{1,2,\dots,n\}$

$$I_{i,j} = \begin{cases} I_{1,j}\ if\ rand \leq 0.5 \\ I_{2,j}\ otherwise \end{cases} \quad (8)$$

where 0.5 refers to uniformly mix the n parameters from the two parents.

Mutation operator, on the other hand, is named as migration operator ($\Phi_{GO-m}: \mathrm{I} \times p_m \rightarrow I$) and it directly operates on the phenotype (or the topological) representation of the PPIN. When mutation operator is activated on protein j for an individual I_i, it will change the complex of this protein to a new complex, say c_k, where it could maintain there the maximum functional similarity (i.e., $\sum_{l \in c_k} \mathcal{FS}_{jl}$ has its maximum value).
$\forall i \in \{1,2,\dots,N\} \wedge \forall j \in \{1,2,\dots,n\}$

$$I_{i,j} = \begin{cases} j' \mid j' \in c_k \wedge argmax_{c_k \in \mathcal{C}}(\sum_{l \in c_k} \mathcal{FS}_{jl}) \; if \; rand \leq p_m \\ I_{i,j} \qquad\qquad otherwise \end{cases} \tag{9}$$

This will imply a modification to the genotype representation. The general framework for the proposed GO-based EA (noted as noted as EA_{GOm}) with modularity density and GO-based mutation operator is sketched out in Algorithm 1.

5. Results and discussions

A yeast called Saccharomyces cerevisiae PPIN (see Figure 1) is used in the experiments [24] to test how well the proposed GO-based EA works. This PPIN has 4687 interactions over 990 proteins. Only 28 proteins have single interactions, while the remaining proteins have two or more interactions, with an average degree of 9.4687 interactions per protein. The protein "YCR057C" (#170) has the highest number of interactions, 52. Thus, weighty proteins are defined as those with more than 9.4687 interactions, while light proteins are those with degrees less than 9.4687. From this PPIN, 30 different noisy PPINs are generated using 5 increasing levels of noise percentage (10%–50%).

Algorithm 1: General framework for the proposed GO-based EA

Input: $\mathcal{N}$,A, $\mathcal{SS}$, $\mathcal{FS}$ //PPIN, topological, Semantic similarity and Functional similarity for n proteins
Input: $Psize$, Φ_s, $\Phi_\times$, Φ_m, $p_\times$, p_m
Output: $C = \{c_1, c_2, \dots, c_K\}$ with maximum QD
begin
$t \leftarrow 0$; // initial generation
$MaxGen \leftarrow 100$; // maximum number of generations
initialize $\mathbf{I}^t \leftarrow \{I_1^t, I_2^t, \dots, I_{Psize}^t\}$;
decode: $\Gamma\left(I_{1 \leq i \leq Psize}^t\right): \{C_1^t, C_2^t, \dots, C_{Psize}^t\}$; //Phenotype as a set of complexes for each individual where

$$C_i^t = \{c_1, c_2, \dots, c_{K_i}\}$$

evaluate : $\mathbf{I}^t: \{QD(C_1^t), QD(C_2^t), \dots, QD(C_{Psize}^t)\}$; // QD for each phenotype
while $(t \neq MaxGen)$ **do**
  select Φ_s: $\mathbf{I}^{t+1} \leftarrow \{(I_1, QD_1)_{1 \leq i \leq Psize} \times (I_2, QD_2)_{1 \leq i \leq Psize}\}$;
  recombine $\Phi_\times$: $\mathbf{I}^{t+1} \leftarrow \{(I_1 \times I_2 \times p_\times)_{1 \leq i \leq Psize}\}$;
  GO-based mutate Φ_m: $\mathbf{I}^{t+1} \leftarrow \{I_{1 \leq i \leq N}^{t+1}, p_m\}$;
  decode: $\Gamma\left(I_{1 \leq i \leq Psize}^{t+1}\right): \{C_1^{t+1}, C_2^{t+1}, \dots, C_{Psize}^{t+1}\}$; // where $C_i^{t+1} = \{c_1, c_2, \dots, c_{K_i}\}$
  evaluate: $\mathbf{I}^{t+1}: \{QD(C_1^{t+1}), QD(C_2^{t+1}), \dots, QD(C_{Psize}^{t+!})\}$;
  $t \leftarrow t + 1$;
end while;
return $I_{1 \leq i \leq Psize}^t$ with **best** $C = \{c_1, c_2, \dots, c_K\}$ with maximum QD**;**
end

For each noise percentage, fake interactions are added to the proteins, or true interactions are deleted from the proteins. Further, for each noise percentage, the perturbation is performed on either random proteins, weighty proteins, or light proteins. Recall that m for the *yeast Saccharomyces cerevisiae* PPIN in Figure 1 equals to 4687, then let m_{fake}, m_{del}, m', $\max(m)$, $n_{\max(m)}$, and $n_{\min(m)}$ to denote, respectively, number of added fake interactions, number of deleted true interactions, total number of interactions in the noisy PPIN, maximum number of interactions per protein, number of proteins with maximum number of interactions, and number of proteins with minimum number of interactions. Table 2 and Table 3 report the characteristics of the synthesized PPINs used in the evaluation.

The reference set, as identified by $Cmplx\_D1$, is used to validate the quality of the detected complexes over yeast PPIN. This truly complex dataset was created from the *Munich Information Center for Protein Sequences* (MIPS) genomes and protein sequences database. It contains 81 true complexes with different sizes ranging from 6 up to 38 proteins.

Table 3: Characteristics of noisy PPINs generated from the yeast Saccharomyces cerevisiae PPIN by adding different percentages of fake interactions to random proteins, weighty proteins, and light proteins

Perturbing random proteins					
Noise%	m_{fake}	m'	$\max(m)$	$n_{\max(m)}$	$n_{\min(m)}$
0.10	469	5156	54	1	11
0.20	937	5624	53	1	2
0.30	1406	6093	55	1	13
0.40	1875	6562	54	1	1
0.50	2344	7031	58	1	1
Perturbing weighty proteins					
Noise%	m_{fake}	m'	$\max(m)$	$n_{\max(m)}$	$n_{\min(m)}$
0.10	378	5065	52	1	28
0.20	757	5444	56	1	28
0.30	1135	5822	55	1	28
0.40	1514	6201	58	1	28
0.50	1892	6579	60	1	28
Perturbing light proteins					
Noise%	m_{fake}	m'	$\max(m)$	$n_{\max(m)}$	$n_{\min(m)}$
0.10	91	4778	52	1	17
0.20	181	4868	52	1	11
0.30	272	4959	52	1	5
0.40	362	5049	52	1	4
0.50	453	5140	52	1	7

Table 4: Characteristics of noisy PPINs generated from yeast Saccharomyces cerevisiae PPIN by deleting different percentage of true interactions from random proteins, weighty proteins, and light proteins

Perturbing random proteins					
Noise%	m_{del}	m'	$\max(m)$	$n_{\max(m)}$	$n_{\min(m)}$
0.10	469	4218	49	1	42
0.20	937	3750	42	1	76
0.30	1406	3281	39	1	116
0.40	1875	2812	32	1	170
0.50	2344	2343	28	1	217
Perturbing weighty proteins					
Noise%	m_{del}	m'	$\max(m)$	$n_{\max(m)}$	$n_{\min(m)}$
0.10	378	4309	45	1	28
0.20	757	3930	41	1	28
0.30	1135	3552	37	1	28
0.40	1514	3173	30	1	30
0.50	1892	2795	31	1	31
Perturbing light proteins					
Noise%	m_{del}	m'	$\max(m)$	$n_{\max(m)}$	$n_{\min(m)}$
0.10	89	4598	52	1	57
0.20	178	4509	52	1	98
0.30	268	4419	52	1	152
0.40	357	4330	52	1	126
0.50	446	4241	52	1	143

Three measures are used in the evaluation. These are complex-wise sensitivity ($sensitivity$), complex-wise positive predictive value (PPV), and geometric accuracy ($accuracy$). Both $sensitivity$ and PPV are based on the size of the intersection between the detected complexes and the true benchmark complexes [23].

$$sensitivity = \frac{\sum_{i=1}^{K_S} max_{j=1}^{K_C} t_{ij}}{\sum_{i=1}^{K_S} |s_i|} \quad (10)$$

$$PPV = \frac{\sum_{j=1}^{K_C} max_{i=1}^{K_S} t_{ij}}{\sum_{j=1}^{K_C} \sum_{i=1}^{K_S} t_{ij}} \quad (11)$$

where t_{ij} acts as the number of proteins shared by both the golden standard complex i and the predicted complex j, while K_S and K_C refer, respectively, to the number of true complexes and the number of the predicted complexes. The geometric accuracy can be utilized to indicate the trade-of between $sensitvity$ and PPV.

$$accuracy = \sqrt{sensitivity * PPV} \quad (12)$$

The performance of the proposed algorithm is compared with the EA of Pizzuti and Rombo [9] with a canonical mutation operator (noted as noted as EA_{Canm}) and the single-objective EA of Attea and Abdullah [11, 30] with a topological-based mutation operator (noted as

noted as EA_{Topm}). The tested algorithms were endorsed for evaluation under a simulation of 30 different runs. Each run is initialized with a random population of 100 individual solutions. The evolutionary process of each algorithm is allowed to continue for 100 generations. The average of the 30 different runs (in terms of the best solution obtained) is reported for each algorithm. The best solution for each algorithm is recognized by its objective function (QD). The results are reported in Tables 4–9. The best result in each test case is given in bold.

Table 5: Performance evaluation for noisy PPINs generated from the yeast Saccharomyces cerevisiae PPIN by adding different percentages of fake interactions to random proteins

Noise%	Sensitivity			PPV			Accuracy		
	EA_{Canm}	EA_{Topm}	EA_{GOm}	EA_{Canm}	EA_{Topm}	EA_{GOm}	EA_{Canm}	EA_{Topm}	EA_{GOm}
0.1	0.8218	0.9565	**0.9587**	0.5789	0.7445	**0.7552**	0.6892	0.8438	**0.8508**
0.2	0.7425	0.8781	**0.9248**	0.4241	0.6363	**0.6412**	0.5606	0.6412	**0.7696**
0.3	0.6918	0.7143	**0.8141**	0.2873	0.3872	**0.4484**	0.4443	0.4484	**0.6027**
0.4	0.6575	0.6398	**0.7091**	0.2240	0.2444	**0.2705**	0.3823	0.2705	**0.4350**
0.5	0.6962	0.6365	**0.6330**	0.1591	0.1818	**0.1949**	0.3309	0.3368	**0.3497**

Table 6: Performance evaluation for noisy PPINs generated from yeast Saccharomyces cerevisiae PPIN by deleting different percentage of true interactions from random proteins

Noise%	Sensitivity			PPV			Accuracy		
	EA_{Canm}	EA_{Topm}	EA_{GOm}	EA_{Canm}	EA_{Topm}	EA_{GOm}	EA_{Canm}	EA_{Topm}	EA_{GOm}
0.1	0.8893	0.9569	**0.9621**	0.7496	0.7882	**0.7985**	0.8162	0.8684	**0.8765**
0.2	0.8795	0.9466	**0.9550**	0.7503	0.7849	**0.7999**	0.8122	0.8619	**0.8740**
0.3	0.8699	0.9281	**0.9417**	0.7632	0.7888	**0.7932**	0.8146	0.8556	**0.8642**
0.4	0.8413	0.9049	**0.9175**	0.7684	0.7951	**0.7996**	0.8038	0.8482	**0.8564**
0.5	0.8239	0.8735	**0.8924**	0.7951	0.8145	**0.8184**	0.8093	0.8434	**0.8545**

Table 7: Performance evaluation for noisy PPINs generated from the yeast Saccharomyces cerevisiae PPIN by adding different percentages of fake interactions to weighty proteins

Noise%	Sensitivity			PPV			Accuracy		
	EA_{Canm}	EA_{Topm}	EA_{GOm}	EA_{Canm}	EA_{Topm}	EA_{GOm}	EA_{Canm}	EA_{Topm}	EA_{GOm}
0.1	0.8425	0.9347	**0.9572**	0.6066	0.7522	**0.7575**	0.7144	0.8384	**0.8514**
0.2	0.8052	0.8973	**0.9320**	0.4858	0.6769	**0.6859**	0.6246	0.7790	**0.7990**
0.3	0.8106	0.8266	**0.8878**	0.4019	0.4983	**0.5302**	0.5697	0.6399	**0.6844**
0.4	0.8192	0.8448	**0.8782**	0.3435	0.4186	**0.4286**	0.5288	0.5920	**0.6108**
0.5	0.8420	0.9148	**0.9402**	0.3136	0.3359	**0.3563**	0.5125	0.5519	**0.5776**

Table 8: Performance evaluation for noisy PPINs generated from the yeast Saccharomyces cerevisiae PPIN by deleting different percentages of true interactions from weighty proteins

Noise%	Sensitivity			PPV			Accuracy		
	EA_{Canm}	EA_{Topm}	EA_{GOm}	EA_{Canm}	EA_{Topm}	EA_{GOm}	EA_{Canm}	EA_{Topm}	EA_{GOm}
0.1	0.8900	0.9544	**0.9654**	0.7493	0.7859	**0.7967**	0.8164	0.8660	**0.8769**
0.2	0.8885	0.9591	**0.9642**	0.7447	0.7967	**0.8043**	0.8132	0.8741	**0.8806**
0.3	0.8658	0.9449	**0.9501**	0.7669	0.8112	**0.8206**	0.8147	0.8754	**0.8829**
0.4	0.8541	0.9380	**0.9524**	0.7677	0.8247	**0.8281**	0.8096	0.8795	**0.8880**
0.5	0.8374	0.9238	**0.9365**	0.7800	0.8211	**0.8313**	0.8081	0.8709	**0.8823**

Table 9: Performance evaluation for noisy PPINs generated from the yeast Saccharomyces cerevisiae PPIN by adding different percentages of fake interactions to light proteins

Noise%	Sensitivity			PPV			Accuracy		
	EA_{Canm}	EA_{Topm}	EA_{GOm}	EA_{Canm}	EA_{Topm}	EA_{GOm}	EA_{Canm}	EA_{Topm}	EA_{GOm}
0.1	0.8776	0.9571	**0.9670**	0.7117	0.7747	**0.7883**	0.7900	0.8610	**0.8731**
0.2	0.8676	0.9502	**0.9653**	0.6875	0.7509	**0.7731**	0.7720	0.8446	**0.8638**
0.3	0.8410	0.9489	**0.9631**	0.6532	0.7539	**0.7641**	0.7408	0.8457	**0.8578**
0.4	0.8291	0.9479	**0.9614**	0.6297	0.7324	**0.7441**	0.7223	0.8331	**0.8457**
0.5	0.8131	0.9425	**0.9625**	0.6013	0.7187	**0.7452**	0.6989	0.8229	**0.8468**

Table 10: Performance evaluation for noisy PPINs generated from the yeast Saccharomyces cerevisiae PPIN by deleting different percentages of true interactions from light proteins

Noise%	Sensitivity			PPV			Accuracy		
	EA_{Canm}	EA_{Topm}	EA_{GOm}	EA_{Canm}	EA_{Topm}	EA_{GOm}	EA_{Canm}	EA_{Topm}	EA_{GOm}
0.1	0.8928	0.9652	**0.9652**	0.7396	0.7796	**0.7934**	0.8125	0.8674	**0.8751**
0.2	0.8858	0.9546	**0.9538**	0.7430	0.7831	**0.7943**	0.8111	0.8646	**0.8704**
0.3	0.8546	0.9184	**0.9214**	0.7337	0.7743	**0.7778**	0.7917	0.8432	**0.8465**
0.4	0.8792	0.9513	**0.9548**	0.7397	0.7838	**0.7881**	0.8062	0.8634	**0.8674**
0.5	0.8791	0.9486	**0.9511**	0.7427	0.7814	**0.7855**	0.8079	0.8609	**0.8643**

The results reported in the tables prove the ability of the proposed EA with GO-based mutation operators to outperform the EA with canonical mutation and the EA with topological-based mutation operators in all evaluation measures and in all noisy PPINs. This performance suggests that the design of the EA should disclose the significance of injecting biological information (i.e., GO semantic similarity and protein functional similarity) into its framework and that acting on this concept can be satisfying and lead to interesting results.

6. Conclusions

Identifying protein complexes in protein-protein interaction networks (PPINs) is a challenging task with broad applications in biological networks, social network modeling, and communication pattern analysis. PPINs can help to understand the mechanisms that control cell life, predict the biological functions of unknown proteins, and have important therapeutic applications. However, the high rate of false positives and false negatives in large-scale experiments can greatly increase the complexity of the networks and lead to unreliable results for topological-based detection algorithms. To address this issue, we proposed an evolutionary algorithm that incorporates information from the functional domain of proteins to detect protein complexes in PPINs. The algorithm's reliability was tested using noisy PPINs being synthesized from a yeast PPIN dataset. The results clarify the ability of the proposed algorithm to outperform both canonical and topological-based EAs in all evaluation measures. In other words, the results prove the effectiveness of the proposed algorithm to handle noisy PPINs.

References

[1] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821-7826, 2002.

[2] G. D. Bader and C. W. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," *BMC Bioinformatics*, vol. 4, no. 1, pp. 1-27, 2003.

[3] J. Gagneur, R. Krause, T. Bouwmeester, and G. Casari, "Modular decomposition of protein-protein interaction networks," *Genome Biology*, vol. 5, pp. 1-12, 2004.

[4] X.-L. Li, C.-S. Foo, and S.-K. Ng, "Discovering protein complexes in dense reliable neighborhoods of protein interaction networks," in *Computational Systems Bioinformatics: (Volume 6)*: World Scientific, 2007, pp. 157-168.

[5] K. Macropol, T. Can, and A. K. Singh, "RRW: repeated random walks on genome-scale protein networks for local cluster discovery," *BMC Bioinformatics*, vol. 10, pp. 1-10, 2009.

[6] G. Liu, L. Wong, and H. N. Chua, "Complex discovery from weighted PPI networks," *Bioinformatics*, vol. 25, no. 15, pp. 1891-1897, 2009.

[7] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761-764, 2010.

[8] M. C. Costanzo *et al.*, "The yeast proteome database (YPD) and Caenorhabditis elegans proteome database (WormPD): "Comprehensive resources for the organization and comparison of model organism protein information," *Nucleic Acids Research*, vol. 28, no. 1, pp. 73-76, 2000.

[9] C. Pizzuti and S. E. Rombo, "Algorithms and tools for protein–protein interaction networks clustering, with a special focus on population-based stochastic methods," *Bioinformatics*, vol. 30, no. 10, pp. 1343-1352, 2014.

[10] C. Pizzuti and S. Rombo, "Experimental evaluation of topological-based fitness functions to detect complexes in PPI networks," in *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, 2012, pp. 193-200.

[11] B. A. Attea and Q. Z. Abdullah, "Improving the performance of evolutionary-based complex detection models in protein–protein interaction networks," *Soft Computing*, vol. 22, pp. 3721-3744, 2018.

[12] R. W. Solava, R. P. Michaels, and T. Milenković, "Graphlet-based edge clustering reveals pathogen-interacting proteins," *Bioinformatics*, vol. 28, no. 18, pp. i480-i486, 2012.

[13] S. Bandyopadhyay, S. Ray, A. Mukhopadhyay, and U. Maulik, "A multiobjective approach for identifying protein complexes and studying their association in multiple disorders," *Algorithms for Molecular Biology*, vol. 10, pp. 1-15, 2015.

[14] S. Ray, A. Hossain, and U. Maulik, "Disease associated protein complex detection: a multi-objective evolutionary approach," in *2016 International conference on microelectronics, computing and communications (MicroCom)*, 2016: IEEE, pp. 1-6.

[15] A. H. Abdulateef, B. A. Attea, and A. N. Rashid, "Heuristic Modularity for Complex Identification in Protein-Protein Interaction Networks," *Iraqi Journal of Science*, vol. 60, no. 8, pp. 1846–1859, Aug. 2019.

[16] I. H. Abdulateef, D. A. J. Alzubaydi, and B. A. Attea, "A Tri-Gene Ontology Migration Operator for Improving the Performance of Meta-heuristics in Complex Detection Problems," *Iraqi Journal of Science*, vol. 64, no. 3, pp. 1426–1441, Mar. 2023.

[17] D. Plewczyński and K. Ginalski, "The interactome: predicting the protein-protein interactions in cells," *Cellular and Molecular Biology Letters*, vol. 14, no. 1, pp. 1-22, 2009.

[18] Hill, D.P., Smith, B., McAndrews-Hill, M.S. and Blake, J.A., "Gene Ontology annotations: what they mean and where they come from," *BMC bioinformatics*, vol. 9, no. 5, pp. 1-9, Apr. 2008.

[19] J. Z. Wang, Z. Du, R. Payattakool, P. S. Yu, and C.-F. Chen, "A new method to measure the semantic similarity of GO terms," *Bioinformatics*, vol. 23, no. 10, pp. 1274-1281, 2007.

[20] C. Pesquita, D. Faria, A. O. Falcao, P. Lord, and F. M. Couto, "Semantic similarity in biomedical ontologies," *PLoS computational Biology*, vol. 5, no. 7, p. e1000443, 2009.

[21] N. Zaki, J. Berengueres, and D. Efimov, "Detection of protein complexes using a protein ranking algorithm," *Proteins: Structure, Function, and Bioinformatics*, vol. 80, no. 10, pp. 2459-2468, 2012.

[22] Z. Li, S. Zhang, R.-S. Wang, X.-S. Zhang, and L. Chen, "Quantitative function for community detection," *Physical review E*, vol. 77, no. 3, pp. 036109, 2008.

[23] S. Brohee and J. Van Helden, "Evaluation of clustering algorithms for protein-protein interaction networks," *BMC bioinformatics*, vol. 7, pp. 1-19, 2006.

[24] Q. Z. Abdullah and A. A. Bara'a, "A Heuristic Strategy for Improving the Performance of Evolutionary Based Complex Detection in Protein-Protein Interaction Networks," *Iraqi Journal of Science*, vol. 57, no. 4A, pp. 2513-2528, 2016.