ISSN: 0067-2904

An Evolutionary Algorithm for Improving the Quantity and Quality of the Detected Complexes from Protein Interaction Networks

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

Received: 18/2/2023 Accepted: 5/5/2023 Published: 30/5/2024

Abstract

One of the recent significant but challenging research studies in computational biology and bioinformatics is to unveil protein complexes from protein-protein interaction networks (PPINs). However, the development of a reliable algorithm to detect more complexes with high quality is still ongoing in many studies. The main contribution of this paper is to improve the effectiveness of the well-known modularity density (QD) model when used as a single objective optimization function in the framework of the canonical evolutionary algorithm (EA). To this end, the design of the EA is modified with a gene ontology-based mutation operator, where the aim is to make a positive collaboration between the modularity density model and the proposed gene ontology-based mutation operator. The performance of the proposed EA to have a high quantity and quality of the detected complexes is assessed on two yeast PPINs and compared with two benchmarking gold complex sets. The reported results reveal the ability of modularity density to be more productive in detecting more complexes with high quality when teamed up with a gene ontology-based mutation operator.

Keywords: EA, Gene ontology, protein complex, protein interaction networks, modularity density.

خوارزمية تطورية لتحسين كمية ونوعية المجمعات المكتشفة من شبكات تفاعل البروتين

صفا أحمد عبدالصاحب*، براء علي عطية
قسم علوم الحاسوب ، كلية العلوم ، جامعة بغداد ، بغداد ، العراق

الخلاصة:

إحدى الدراسات البحثية الحديثة المهمة، ولكن الصعبة، في البيولوجيا الحاسوبية والمعلوماتية الحيوية هي مشكلة الكشف عن مجمعات البروتين في شبكات تفاعل البروتين والبروتين (PPINs). ومع ذلك، فإن تطوير خوارزمية موثوقة لاكتشاف المزيد من المجمعات ذات الجودة العالية لا يزال مستمراً في العديد من الدراسات. تتمثل المساهمة الرئيسية لهذا البحث في تحسين فعالية النموذج المعروف بالكثافة النمطية (QD) عند استعماله كدالة أمثية الأحادية الموضوعية في قالب الخوارزمية التطورية القانونية (EA). تحقيقا لهذه الغاية، تم تعديل تصميم ال EA باستعمال مشغل طفرة يعتمد على علم الجينات، حيث يكون الهدف هو إقامة تعاون إيجابي بين نموذج الكثافة النمطية ومشغل الطفرة المقترح القائم على علم الجينات. يتم تقييم أداء ال EA المقترح للحصول على كمية وجودة عالية للمجمعات البروتينية المكتشفة على اثنين من شبكات الخمائر ومقارنة المجمعات المكتشفة بمجموعتين من المجمعات القياسية الذهبية. تثبت النتائج عن قدرة الكثافة النمطية

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

على أن تكون أكثر إنتاجية في اكتشاف المزيد من المجمعات البروتينية وبنفس الوقت ذات جودة عالية عند التعاون مع مشغل الطفرات الجينية القائم على علم الوجود.

1. Introduction

A key feature of a networked system is the general tendency toward organizing nodes hierarchically into multiple cohesive modules or communities. However, identifying such communities is a challenging problem in network research, with applications in biological networks, social network modeling, and communication pattern analysis [1–7]. Proteins that control and mediate many biological activities by regulating and supporting one another through their interactions form biological networks [1, 8]. These networks can be represented as protein-protein interaction networks (PPINs), which are powerful modular organizations for understanding protein functional qualities and their future potential as biomarkers of cellular organization. A PPIN holds information on the protein-protein interactome of any organism.

Figure 1 depicts an illustrative example of a *yeast Saccharomyces cerevisiae* PPIN (left) that has 990 different proteins, obtained from the Yeast Protein Database [8], with 4687 interactions. Based on the golden reference set of 81 complexes maintained by *the Munich Information Center for Protein Sequences* (MIPS) database for genome annotation, gene expression analysis, and proteomics, this protein interaction network is decomposed into 78 different-sized complexes [8].

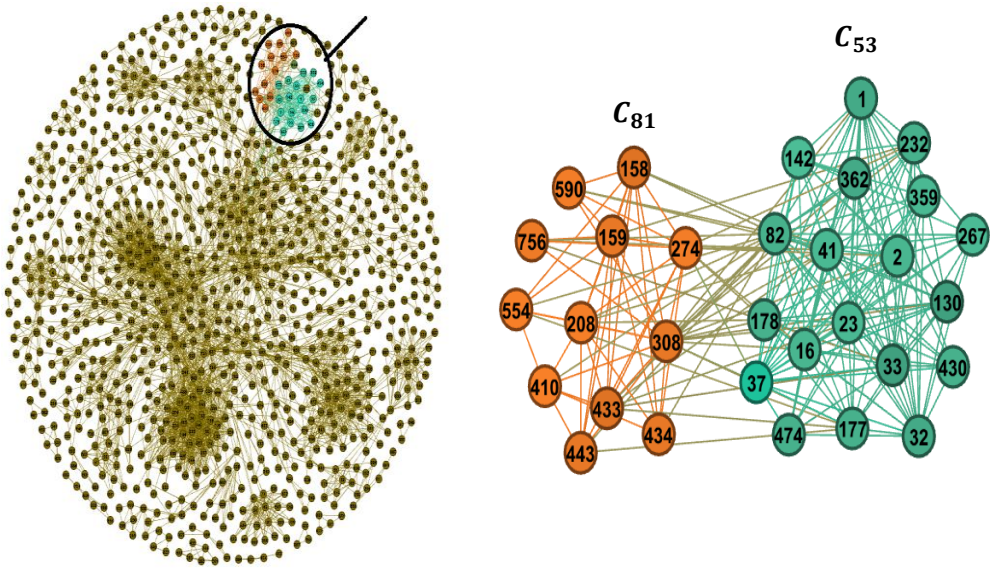


Figure 1: A *yeast Saccharomyces cerevisiae* network (left) and two complex (C_{81} and C_{53}) are zoomed out in the right.

In PPINs, protein interactions can indicate the formation of either stable or transient protein complexes (or functional modules), as well as either physical or functional interactions. A protein complex, then, is defined as a group of proteins that work together to carry out a specific biological process or activity. Figure 2 depicts an illustrative example of the two complexes that are zoomed out in Figure 1. The yeast proteins in Figure 2 are depicted with their names and their intra- and inter-cellular connections.

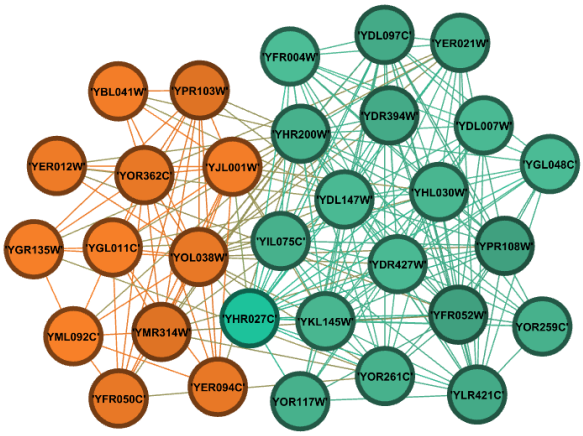


Figure 2: An illustrative example of two complexes ($C_{53}$and C_{81}) of yeast proteins (from *yeast-D1 Saccharomyces cerevisiae* PPIN) with their identity names, intra connections and inter connections.

Protein complex detection in a protein-protein interaction network (PPIN) is a graph clustering problem that involves identifying densely connected regions in the network as genuine protein complexes. This problem is informally defined as an optimization problem and has been proven to be non-deterministic polynomial-time hard (NP-hard) [3]. In other words, it is computationally very difficult to find an optimal solution in a reasonable amount of time.

In this paper, a single objective EA is proposed, and its robustness is evaluated in terms of the quantity and quality of the detected complexes. The definition of the problem is formulated as a single objective function of the modularity density. Further, the performance of the modularity density model is examined using two yeast PPINs under the teamwork of a GO-based mutation operator. The coming sections are outlined as follows: The well-known heuristic and meta-heuristic (i.e., evolutionary-based) complex detection algorithms proposed in the literature are presented next. This is followed by the formal representation of proteins and protein interaction networks in both the topological and biological domains, which are presented in the next three sections. The details of the proposed evolutionary-based complex detection algorithm are given in Section 5. This is followed by the simulation results and a description of the main findings of the research. Finally, a conclusion and recommendation for future work are given in Section 7.

2. Related work

Different meta-heuristic algorithms, mainly evolutionary algorithms (EAs), were proposed in the literature to detect protein complexes from PPINs. The EA-based complex detection methods are proven to be more robust than their counterparts, the heuristic-based complex detection methods. Examples of such heuristic-based methods are 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, 9].

One of the earliest works to identify the importance of evolutionary algorithms for solving complex detection problems is recognized by Pizzuti and Rombo in [10] and [11]. Evolutionary-based complex detection algorithms use evolutionary principles, i.e., natural selection and genetic variation, to search for promising candidates for protein complex

structures. These algorithms typically involve generating a population of candidate solutions (e.g., protein complexes), evaluating their fitness based on one or more criteria (e.g., connectivity, density, and functional coherence), and iteratively evolving the population towards better solutions through selection, recombination, and mutation. They developed a single-objective genetic algorithm (GA) with different single-objective complex detection models to solve the problem. The remaining components of the GA (i.e., selection, crossover, and mutation operators) were designed based on their well-known traditional forms. All their models (i.e., objective functions) 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. On the other hand, traditional modularity explicitly defines the intra-complex structure score only.

Unlike the single-objective models examined by Pizzuti and Rombo in [10] and [11], Bandyopadhyay et al. and Ray et al. in, respectively, [12] and [13], on the other hand, 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. They designed a multi-objective genetic algorithm outlined by the well-known non-dominated sorting algorithm (NSGA-II) for solving the complex detection problem.

In [14], in 2016, a multi-objective evolutionary co-clustering model for social community discovery was proposed. The model identifies disjoint communities using evolutionary algorithms and co-clustering. It describes four types of neighborhood nodes and relations and proposes a heuristic mutation operator to increase the convergence velocity and reliability of the adopted multi-objective optimization model. The heuristic operator lets nodes migrate across communities based on neighborhood relationships.

In [15], two contradictory topological-based structures were formulated to reflect the intra-complex structure and the inter-complex structure as a multi-objective optimization model. The adopted multi-objective evolutionary algorithm was framed by the well-known decomposition-based multi-objective evolutionary algorithm (MOEA/D). In [16] and [17], 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 are often more robust and less sensitive to parameter settings than heuristic algorithms, and they can potentially provide 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 bioontologies were rarely considered in the design of evolutionary-based complex detection algorithms. Only recently, [18] examined the design of the mutation operator in an EA (with a modularity model) based on the biological information inherited 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. Formal representation of PPIN in topological domain

Commonly, a protein-protein interaction network (PPIN), is usually formulated as an undirected graph $G\ (V, E)$. The set of vertices V represents n proteins, i.e. $V = \{v_1, v_2, \ldots, v_n\}$, while the set E of edges embodies the m protein interactions, i.e. $E = \{e_1, e_2, \ldots e_m\}$. Since it is believed that proteins that interact are more likely to perform similar biological functions within a PPIN, there are dense regions (protein complexes) in more than one tightly linked region in the graph. Figure 3 illustrates a graph (small PPIN) example with eight nodes being decomposed into two sub-graphs or complexes, $Complex_1$ and $Complex_2$, respectively.

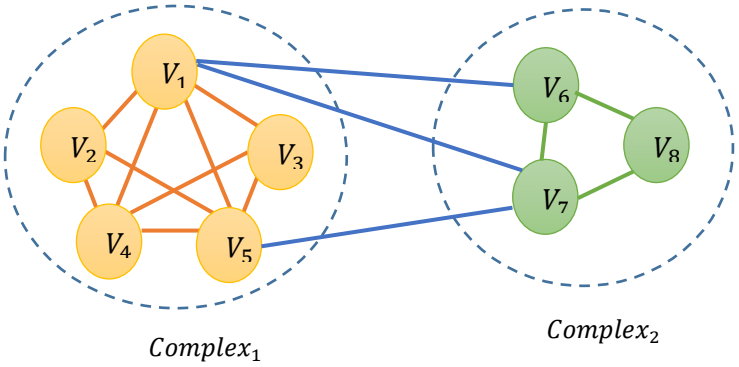


Figure 3: A small PPIN of 8 proteins is decomposed into two complexes. The nodes within a dashed circle form one complex. The edges inside the dashed circle are intra-connections, while those connecting the two complexes are inter-connections.

Mathematically, the graph G of a PPIN can be represented as a square symmetric adjacency matrix, $A = [a_{ij}]^{n \times n}$. If proteins that resemble to vertices v_i and v_j have a biological interaction, it can be interpreted that entry a_{ij} and its counterpart entry a_{ji} of the adjacency matrix A are both set on; otherwise, they are set off. Figure 4 presents the adjacency matrix A for the PPIN depicted in Figure 3. Further, the adjacency matrix can be represented as a set of n adjacency lists $\mathcal{L} = \{\ell_1, \ell_2, \ell_3, \ldots, \ell_n\}$. Using a separate list ℓ_i for each protein $\mathcal{p}_i \in \mathcal{P}$ to aggregate all 1 entries in row i. As a result:

$$|\ell_i| = \sum_{j=1}^{n} (a_{ij}) \quad (1)$$

and

$$|\mathcal{L}| = \sum_{i=1}^{n} \ell_i \quad (2)$$

8×8

$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{array} \begin{array}{c} \begin{array}{cccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

Figure 3: Adjacency matrix for the PPIN in Figure 3, where "1" indicates that the corresponding pair of proteins interacts; otherwise, "0" means no biological interaction. All diagonal entries are set to "0."

4. Formal representation of proteins in biological domain

4.1 Annotation with Gene ontology terms

Gene ontology (GO), as a dynamic ontology, is a popular 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 composed 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 the multiple molecular activities of sets of interacted proteins.

Every GO term has a unique human-readable GO name, e.g., transcription corepressor activity or amino acid binding—and a unique GO seven-digit identifier prefixed by GO:, e.g., 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 the three sub-ontologies. These were downloaded from the Saccharomyces Genome Database (SGD) at http://genome-www.stanford.edu/Saccharomyces/.

4.2 Graph structure of a GO term

Each individual sub-ontology term (t) can be structured hierarchically by an independent directed acyclic graphs (DAG). A directed graph is made up of a set of nodes and a set of edges, 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 '$regulate$'. 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_2. 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.

Table 1: 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:0005634, GO:0008540, GO:0034515,	[GO:0004175, GO:0031625, GO:0030234]

		GO:0050790]	GO:0000502]	
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]

Generally, a GO term may have connections to more than one GO child term (more specific) node, but unlike these GO terms, it can also have more than one parent (broader) node and different relations to its different parents. For example, in Figure 5, the GO term "cytoplasm" (GO:0005737) has two parents: it is_a cellular anatomical entity and it is $part\_of$ the intracellular anatomical structure.

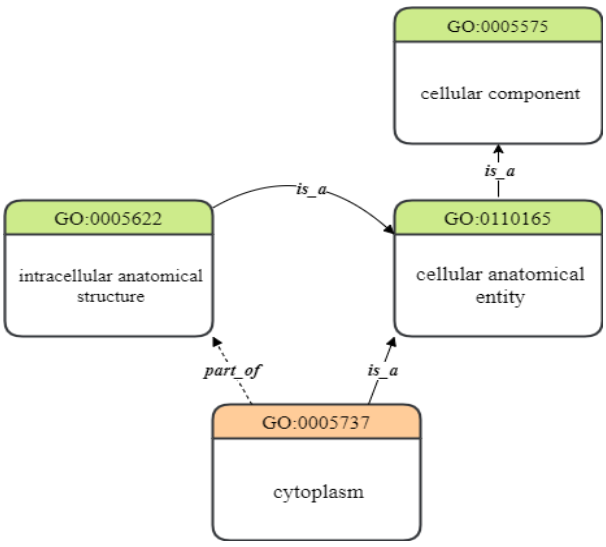


Figure 5: Graph-based representation for GO terms and relations.

Each GO term can be represented as a DAG, in which each term represents a child node of one or more parent nodes. A formal representation of a GO term A is given by $DAG_S(A, \mathcal{TP}_S, \mathbb{E}_S)$, where $\mathcal{TP}_S$ is the set of GO terms in DAG_S that includes term sub-

ontology S and all of its ancestor terms in the GO graph, and $\mathbb{E}_S$ is the set of edges (semantic relations) connecting the GO terms in DAG_S. Then, gene products are annotated with GO terms either directly (i.e. $\mathcal{TP}_S$) or via inheritance or the true path rule, as annotation to a given term, ($t \in \mathcal{TP}_S$), implies annotation to all of its ancestor t_s terms in $DAG(t)$. Then, we may define an ancestor set, $Anc(t)$, for some t as:

$$Anc(t): DAG \rightarrow \{t_s | \exists\ path\ (t_s, t)\} \quad (6)$$

As an illustrative example, consider the three $DAGs$ in Figure 6, of three GO terms for the protein "$YPL139C$ ". The GO terms are: $\mathcal{MF}$ ($GO: 0003714$), $\mathcal{BP}$ ($GO: 0051321$), and $\mathcal{CC}$ ($GO: 005634$). For example, in the figure, the DAG for $GO: 0051321$ (meiotic cell cycle) has six terms connected with six 'is_a' relations and one '$part\_of$' relation. Also, 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).

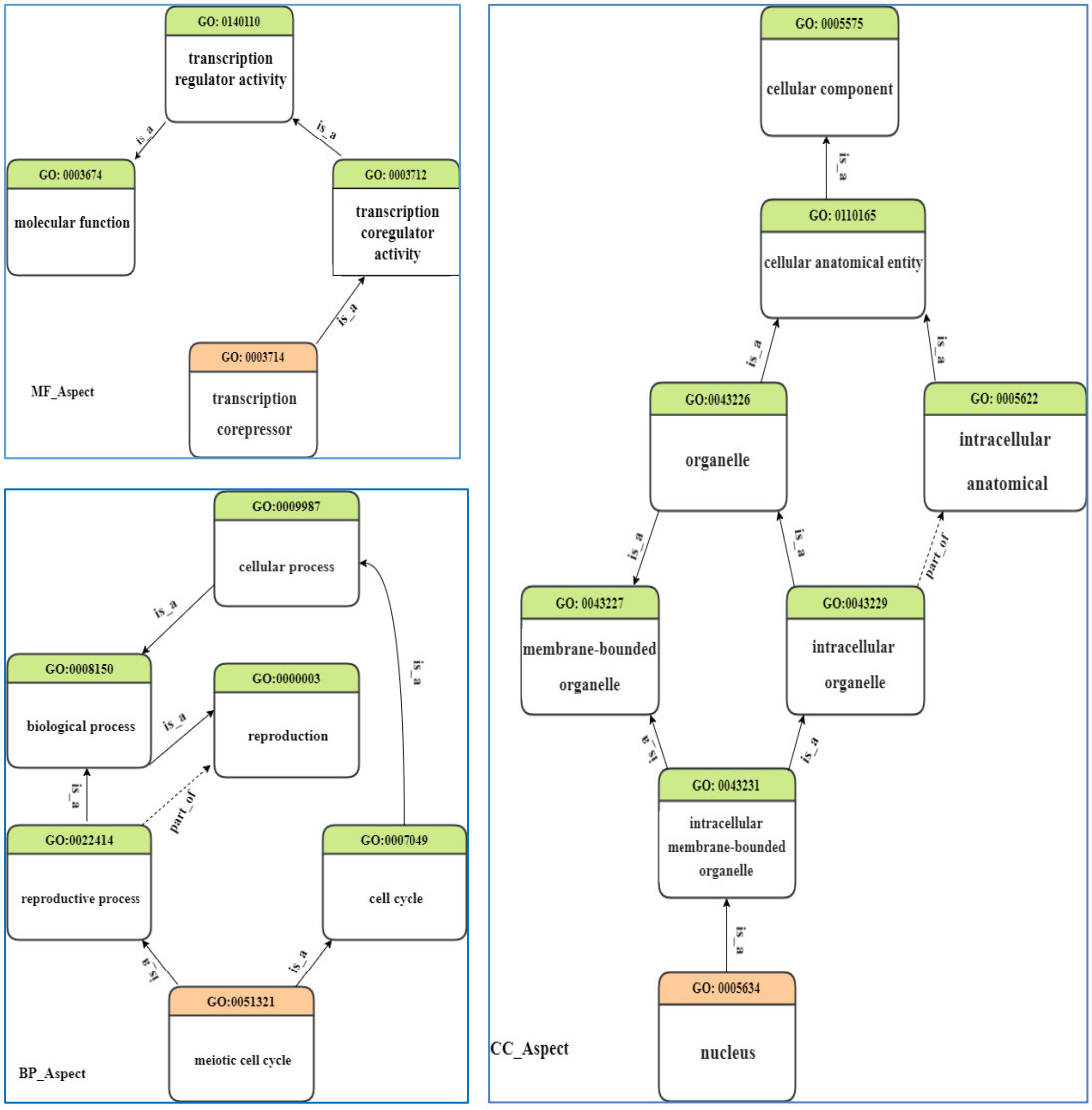


Figure 6: Three DAGs for three different GO terms for the protein " YPL139C." one MF term (GO: 0003714) (top left), one BP term (GO: 0051321), and one CC term (GO:005634) (right). Solid arrows represent 'is_a' relations while dashed arrows represent 'part_of ' relations.

4.3 GO-based semantic similarity

Term semantic similarity in any ontology provides a numerical measure of how closely related and differently defined terms are to one another. 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 $\mathcal{S} = [\mathcal{SS}]^{\mathcal{N}\times\mathcal{N}}$is obtained for $\mathcal{N}$ 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. Wang et al. [19] proposed a semantic similarity based on semantic value and semantic contribution. The semantic value $\mathcal{S}(t)$: $DAG(t) \rightarrow \mathcal{R}^+$ for a GO term t is computed as the sum of the semantic contribution ($\mathcal{SC}$) of all GO terms in $DAG(t)$, $\mathcal{SC}$: $t \times t_s \times DAG(t) \rightarrow \mathcal{R}^+$, along the best (i.e. maximum) weighted paths to t. Here, $\mathcal{SC}\big(t, DAG(t)\big) = 1$. 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. This is formulated in Eq. 7. 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.

$$\mathcal{SC}\big(t_s, DAG(t)\big) = \max\{w \times \mathcal{SC}(t_s^{\square}) | \exists\, e(t_s, t_s^{\square})\} \tag{7}$$

where the directional relation between t_s and $t_s^{\square}$ is denoted by the expression $e(t_s, t_s^{\square})$. Then, the semantic value of the term t in its DAG is defined by the Eq. 3, and the semantic similarity between two GO terms, t_1 and t_2, 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_2 to the semantic values of t_1 and t_2, respectively, in Eq. 4.

$$\mathcal{S}(t) = \sum_{t_i \in DAG(t)} \mathcal{SC}(t_i, DAG(t)) \tag{8}$$

$$\mathcal{SS}(t_1, t_2) = \frac{\sum_{t \in DAG(t_1) \cap DAG(t_2)} \mathcal{SC}\big(t, DAG(t_1)\big) + \mathcal{SC}\big(t, DAG(t_2)\big)}{\mathcal{S}(t_1) + \mathcal{S}(t_2)} \tag{9}$$

4.4 Gene functional similarity

Functional similarity ($\mathcal{FS}$) measures the degree to which two proteins share functional properties. This can be inferred using GO annotations as evidence. 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 of the proteins within a specific category (i.e., $\mathcal{MF}$, $\mathcal{BP}$, or $\mathcal{CC}$) or with all sub-ontology types. 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, $\mathcal{T}_\mathcal{P} = \{\mathcal{MF}, \mathcal{BP}, \mathcal{CC}\}$. 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., $\mathcal{T}_\mathcal{P} \bigcup \mathcal{T}_{t|t\in\mathcal{T}_\mathcal{P}}$, where $\mathcal{T}_t = t \bigcup \{t_s\}$ indicates that the term t and all of its ancestors.

Two major categories for the calculation of $\mathcal{FS}$ can be found in the literature. These are *group-wise* and *pairwise* methods [20]. One of the well-known group-wise methods is Jaccard as defined in Eq. 10.

$$\mathcal{FS}_{Jaccard}(\mathcal{P}_1, \mathcal{P}_2) = \frac{|\mathcal{T}_{\mathcal{P}1} \cap \mathcal{T}_{\mathcal{P}2}|}{|\mathcal{T}_{\mathcal{P}1} \cup \mathcal{T}_{\mathcal{P}2}|} \tag{10}$$

Pair-wise methods (e.g., *average*, *sum*, *maximum*, or *minimum*), on the other hand, 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 (i.e. proteins) $\mathcal{P}_1$ and $\mathcal{P}_2$. For example, maximum functional similarity is defined in Eq. 11.

$$\mathcal{FS}_{Max}(\mathcal{P}_1, \mathcal{P}_2) = \max[\mathcal{SS}(t_1, t_2)] \,|t_1 \in \mathcal{T}_{\mathcal{P}1}, t_2 \in \mathcal{T}_{\mathcal{P}2} \quad (11)$$

5. The proposed evolutionary algorithm

5.1 The general framework

The general framework of the proposed evolutionary-based complex detection algorithm can be formally termed as **ECD**:$\mathbf{I}^{\mu} \longrightarrow \mathbf{I}^{\mu}$. The framework is defined as an iterative composition function that transforms, through a set of three primary operators, an initial population of solutions into an evolved set of solutions. A population can be formally expressed as $\mathbf{I}^{\mu} = \{I_1, I_2, \dots, I_{\mu}\}$ containing μ encoded (i.e., genotype) solutions. The locus-based adjacency representation [21] is adopted in the proposed ECD.

The initial population is generated randomly from all alternative solutions in the search space, Ω of the problem. The evolutionary algorithm evolves the population for more accurate solutions by mating pool selection, $\mathcal{S}$: $\mathbf{I}^{\mu} \longrightarrow \mathbf{I}^{\mu}$, recombination of sets of parents Φ_x: $\mathbf{I} \times \mathbf{I} \longrightarrow \mathbf{I}$ and mutation of individual offspring Φ_m: $\boldsymbol{I} \longrightarrow \boldsymbol{I}$. The evolutionary process continues until a termination criterion $\mathbb{t}$ is satisfied $\mathbb{t}: \mathbf{I}^{\mu} \longrightarrow \{true, false\}$.

5.2 Solution encoding and decoding

The first decision in the design of any evolutionary algorithm is how to represent the solution in genotype space. The solution is encoded as a list of n neighborhood-based representations. For PPIN with n proteins, a genotype solution or individual $I_{1 \le i \le n} \in \mathbf{I}$ is mapped with a list of n parameters:

$$I_i = (I_{i,1}, I_{i,2}, \dots, I_{i,n}) \quad (12)$$

including n loci for n proteins. Each locus is defined by $1 \le j \le n$ and its allele (or value), $I_{i,j} = j'$. Where j refers to protein j in the network, while j' is one of the interacted proteins with protein j, i.e., $a_{jj'} = 1$. The starting set of solutions is encoded randomly from the total search space Ω of the complex detection problem.

The decoding function $\Gamma: I \to C$, then, decodes a genotypic solution I into its corresponding set of complexes (i.e., phenotype). Here, it is a set of complexes $C = \{c_1, c_2, \dots, c_K\}$ of K complexes. For the decoding function, each protein should belong to only one complex, with no overlap between any two complexes.

Figure 7 depicts an illustrative example of a small yeast PPIN with 17 proteins and a total of 97 interactions. Three different individual solutions, with their genotype and phenotype representations, are depicted. According to the allele values in the genotype solutions, different phenotypes are revealed. The first genotype is decoded into two disjoint complexes, while the second and third genotypes are decoded, respectively, into three and four complexes.

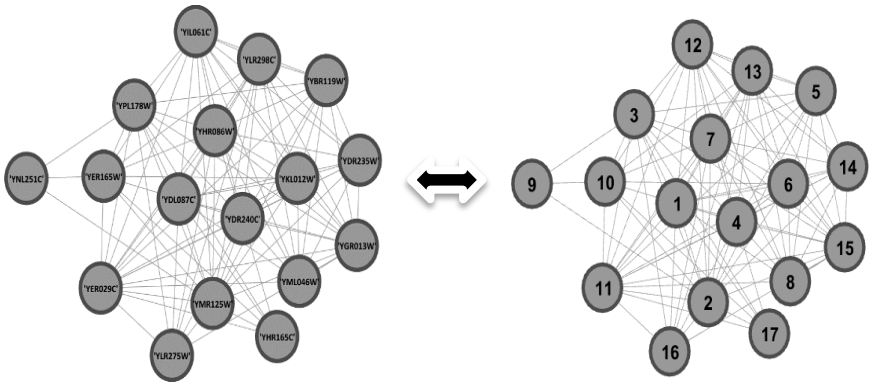
5.3 Objective function

One of the main keys to the success of the evolutionary algorithm is the right selection of the objective function. In this paper, we adopted the well-known modularity density (QD) [22]. For each solution, QD is defined as the sum of the average density of the sub-graphs that constitute the whole solution. In each sub-graph, the density is measured as the difference

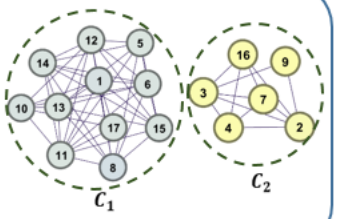
between the intra and inter degrees proportioned to the size of the sub-graph, and it is formulated by:

$$QD = \sum_{k=1}^{K} \frac{L(c_k,c_k) - L(c_k,c_{k'\neq k})}{|c_k|} \quad (12)$$

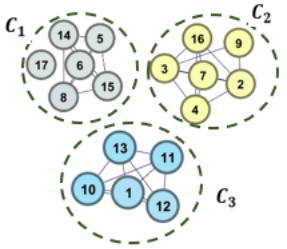
In Eq. 12 and for a solution with K complexes, $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$, the numerator expresses the difference between two terms. The first term is the inner degree of a community c_k, which is twice the number of edges inc_k divided by the number of nodes in the same complex c_k, The 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.



Protein Label	'YDL087C'	'YMR125W'	'YER165W'	'YIL061C'	'YPL178W'	'YDR240C'	'YDR235W'	'YBR119W'	'YKL012W'	'YHR086W'	'YML046W'	'YGR013W'	'YLR298C'	'YLR275W'	'YHR165C'	'YNL251C'	'YER029C'
Protein number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
allele value	5	16	9	3	6	15	4	11	2	14	10	151	12	13	17	7	8
Complex ID	1	2	2	2	1	1	2	1	2	1	1	1	1	1	1	2	1



Protein Label	'YDL087C'	'YMR125W'	'YER165W'	'YIL061C'	'YPL178W'	'YDR240C'	'YDR235W'	'YBR119W'	'YKL012W'	'YHR086W'	'YML046W'	'YGR013W'	'YLR298C'	'YLR275W'	'YHR165C'	'YNL251C'	'YER029C'
Protein number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
allele value	13	16	9	3	6	8	4	14	2	11	1	10	12	15	5	7	10
Complex ID	3	2	2	2	1	1	2	1	2	3	3	3	3	1	1	2	1



Protein Label	'YDL087C'	'YMR125W'	'YER165W'	'YIL061C'	'YPL178W'	'YDR240C'	'YDR235W'	'YBR119W'	'YKL012W'	'YHR086W'	'YML046W'	'YGR013W'	'YLR298C'	'YLR275W'	'YHR165C'	'YNL251C'	'YER029C'
Protein number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
allele value	17	9	16	5	6	14	2	5	3	1	10	13	15	8	12	7	11
Complex ID	3	2	2	1	1	1	2	1	2	3	3	4	4	1	4	2	3

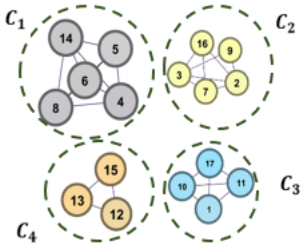


Figure 7: Small Yeast PPIN with 17 proteins and a total of 97 interactions. Three different individual solutions, with their genotype and phenotype representations, are depicted.

5.4 Evolutionary operators

5.4.1 Crossover operator

For the proposed EA-based complex detection algorithm, the uniform crossover operator is used to create a new solution by combining the genotypes of two parent solutions. For two selected parents I_1 and I_2, the crossover operator uniformly mixes their n decision making parameters. The offspring individual uniformly inherits the topological information from the two individual parents, I_1 and I_2. Note that crossover operator crosses the two parents if a random number retains a value less than or equal to the probability of crossover $p_\times$. Here, $p_\times$is set to 0.8.

$\forall i \in \{1,2,\dots,N\} \wedge \forall j \in \{1,2,\dots,n\}$

$$I_{i,j} = \begin{cases} I_{1,j} \; if \; rand \leq 0.5 \\ I_{2,j} \; otherwise \end{cases} \tag{13}$$

where $rand \backsim [0,1]$ is a uniform random value, sampled a new one for every new offspring individual I_i.

5.4.2 Gene ontology-based mutation operator

The traditional mutation operator (Φ_m) proposed by Pizzuti and Rombo [11] in Eq. 14 has been applied with a simple topological-based domain, which operates on the genotype representation. This will eventually change the phenotype structure of the solution.

$\forall i \in (\{1,2,\dots,N\}) \wedge \quad (\forall j \in \{1,2,\dots,n\})$

$$I_{i,j} = \begin{cases} j' \;\; | a_{jj'} = 1 \; if \; rand \leq P_m \\ I_{i,j} \qquad\qquad otherwise \end{cases} \tag{14}$$

where the allele $I_{i,j}$ of the mutated protein j in an individual solution I_i can be swapped with any other direct neighbor j' of j and the $rand$ is a uniform random value, a new one is chosen for every protein $\mathcal{P}_j$.

In this paper, we extend the topological-based mutation proposed in [15] to operate on the functional domain rather than the topological domain. We adopted Jaccard and maximum functional similarity to direct the operator in choosing the candidate complex for the mutated protein to maintain maximum function homogeneity. The mutation operator can be expressed as follows:

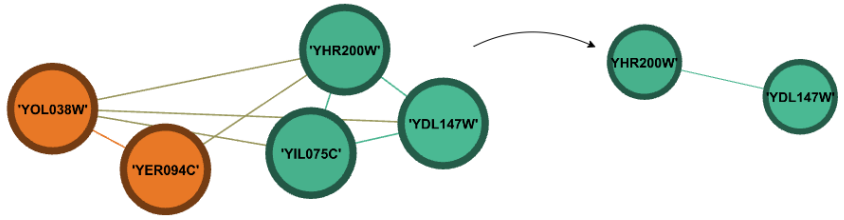
$\forall i \in (\{1,2,\dots,N\}) \wedge \forall j \in (\{1,2,\dots,n\})$

$$I_{i,j} = \begin{cases} j' \;\; | j' \in c_k \wedge argmax_{c_k \in \mathcal{C}} (\sum_{\forall P_{j'} \in \mathcal{C}} \mathcal{FS}(\mathcal{P}_j, \mathcal{P}_{j'})) \;\; if \; rand \leq p_m \\ I_{i,j} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad otherwise \end{cases} \tag{15}$$

where $\mathcal{FS}(\mathcal{P}_j, \mathcal{P}_{j'})$ is the Jaccard or the maximum functional similarity (Eq. 11) between the direct terms, $\mathcal{T}_\mathcal{P}$, of protein pairs $\mathcal{P}_j$ and $\mathcal{P}_{j'}$. In the maximum functional similarity, each GO term of the first protein $\mathcal{P}_j$ is coupled with all GO terms of the second protein $\mathcal{P}_{j'}$ and the maximum similarity is evaluated over all GO pairs of $\mathcal{P}_j$ and $\mathcal{P}_{j'}$. Further, Wang's semantic similarity method is used to evaluate the similarity of GO terms and their DAGs for both protein $\mathcal{P}_j$ and $\mathcal{P}_{j'}$.

Let us consider an example for the yeast protein "YHR200W" in Table 1 and two of its neighboring proteins. The first protein is "YIL075C," which has an intra-connection with "YHR200W." The second protein, however, is "YOL038W," which has an interconnection with "YHR200W." Figures 8 and 9 depict the direct GO terms of these proteins. All GO

categories are reported in the figures. In these figures, the common GO terms for both proteins "YIL075C" and "YOL038W" with protein "YHR200W" are clarified with a checkmark. Topological (i.e., adjacency) and functional similarity (in terms of both Jaccard and maximum functional similarities) are also reported.



'YHR200W'

BP		CC		MF	
✓ GO:0006511	ubiquitin-dependent protein catabolic process	✓ GO:0000502	proteasome complex	GO: 0036435	K48-linked polyubiquitin modification-dependent protein binding
✓ GO:0043248	proteasome assembly	✓ GO:0008540	proteasome regulatory particle, base subcomplex	GO:0031593	polyubiquitin modification-dependent protein binding
GO:0043161	proteasome-mediated ubiquitin-dependent protein catabolic process	GO:0005829	Cytosol		
		✓ GO:0005634	nucleus		

'YIL075C'

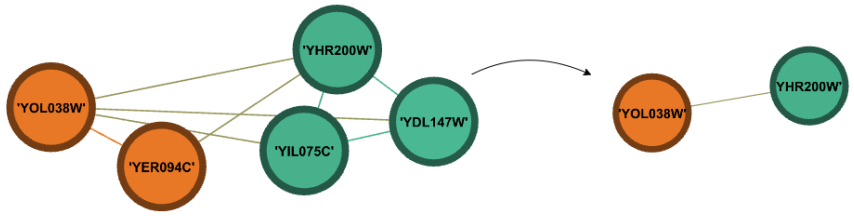
BP		CC		MF	
✓ GO:0006511	ubiquitin-dependent protein catabolic process	✓ GO:0005634	nucleus	GO:0004175	endopeptidase activity
✓ GO:0043248	proteasome assembly	✓ GO:0008540	proteasome regulatory particle, base subcomplex	GO:0031625	ubiquitin protein ligase binding
GO:0042176	proteasome assembly	GO:0034515	proteasome storage granule	GO:0030234	enzyme regulator activity
GO:0050790	regulation of catalytic activity	✓ GO:0000502	proteasome complex		

Figure 8: Direct GO terms (in terms of three categories) for proteins 'YHR200W' and 'YIL075C'.

$a_{ij} = a_{82\ 178} = a_{\text{'YHR200W'},\ \text{'YIL075C'}} = 1$

$\mathcal{FS}_{Jaccard}(\text{'YHR200W'}, \text{'YIL075C'}) = \frac{|\mathcal{T}_{\text{'YHR200W'}} \cap \mathcal{T}_{\text{'YIL075C'}}|}{|\mathcal{T}_{\text{'YHR200W'}} \cup \mathcal{T}_{\text{'YIL075C'}}|} = \frac{5}{14} = 0.33$

$\mathcal{FS}_{Max}(\text{'YHR200W'}, \text{'YIL075C'}) = \mathcal{FS}_{Max(BP)} + \mathcal{FS}_{Max(CC)} + \mathcal{FS}_{Max(MF)} = 1 + 1 + 0.3987 = 2.3987$



'YHR200W'

BP		CC		MF	
✓	GO:0006511	ubiquitin-dependent protein catabolic process	✓ GO:0000502	proteasome complex	GO: 0036435	K48-linked polyubiquitin modification-dependent protein binding
	GO:0043248	proteasome assembly	GO:0008540	proteasome regulatory particle, base subcomplex	GO:0031593	polyubiquitin modification-dependent protein binding
	GO:0043161	proteasome-mediated ubiquitin-dependent protein catabolic process	GO:0005829	Cytosol		
			✓ GO:0005634	Nucleus		

'YOL038W'

BP		CC		MF	
	GO:0010499	proteasomal ubiquitin-independent protein catabolic process	✓ GO:0005634	nucleus	GO:0003674	molecular function
✓	GO:0006511	ubiquitin-dependent protein catabolic process	GO:0005739	mitochondrion	GO:0004298	threonine-type endopeptidase activity
	GO:0005737	cytoplasm	GO:0019773	proteasome core complex, alpha-subunit complex	GO:0004175	endopeptidase activity
			GO:0034515	proteasome storage granule		
			GO:0042175	nuclear outer membrane-endoplasmic reticulum membrane network		
			GO:0005737	cytoplasm		
			✓ GO:0000502	proteasome complex		
			GO:0005839	proteasome core complex		

Figure 9: Direct GO terms (in terms of three categories) for proteins 'YHR200W 'and 'YOL038W'.

$a_{ij} = a_{82\ 308} = a_{\text{'YHR200W'},\ \text{'YOL038W'}} = 1$

$\mathcal{FS}_{Jaccard}(\text{'YHR200W'}, \text{'YOL038W'}) = \frac{|\mathcal{T}_{\text{'YHR200W'}} \cap \mathcal{T}_{\text{'YOL038W'}}|}{|\mathcal{T}_{\text{'YHR200W'}} \cup \mathcal{T}_{\text{'YOL038W'}}|} = \frac{3}{20} = 0.15$

$\mathcal{FS}_{Max}(\text{'YHR200W'}, \text{'YOL038W'}) = \mathcal{FS}_{Max(BP)} + \mathcal{FS}_{Max(CC)} + \mathcal{FS}_{Max(MF)} = 1 + 1 + 0.3232 =$

2.3232

6. Results and discussions

6.1 Dataset for PPIN and benchmark protein complexes

The performance evaluation of the tested evolutionary-based complex detection methods is reported using *Saccharomyces cerevisiae* PPINs. The first PPIN (Yeast-D1) dataset was arranged by Gavin et al. [23] and filtered by Zaki et al. [24]. It contains 990 yeast proteins coupled in pairs, with 4687 different interactions. In this network, all but 28 proteins have more than one interaction, for an average of 9.4687 interactions per protein. The maximum number of interactions is owned by protein 'YCR057C' (protein #170) to reach 52 different interactions. The second PPIN (Yeast-D2) has 1443 different yeast proteins with a total of 6993 interactions. All but 92 proteins have more than one interaction, with an average of 9.6923 interactions. In this network, protein 'YHR052W' (protein #339) holds the maximum number of interactions, which is 59 different interactions.

Further, each protein is annotated with both direct and indirect GO terms. The direct GO terms (in BP, CC, and MF categories) together with their DAG ancestor terms are downloaded from the Saccharomyces Genome Database (SGD) at url: http://genome-www.stanford.edu/Saccharomyces/ in November 2022. For Yeast-D2, on the other hand, all 1443 proteins are annotated with a total of 1552 BP terms, 558 CC terms, and 663 MF terms. Two different sets of true yeast complexes, $Cmplx\_D1$ and $Cmplx\ \_D2$, are used to assess the detection reliability of the algorithms over, respectively, Yeast-D1 and Yeast-D2. Both benchmark sets are supported by the Munich Information Center for Protein Sequence (MIPS) genome and protein sequence databases. The first complex set, i.e. $Cmplx\_D1$ dataset covers 81 different true or reference complexes scopes from 6 to 38 different proteins. The second set of complexes, i.e. $Cmplx\ \_D2$ dataset, on the other hand, encompasses more reference complexes and reaches up to 162 complexes with sizes ranging from 4 to 266 yeast proteins. However, in this set, there are 12 true complexes with completely unknown proteins to Yeast-D2, while 680 yeast proteins known to Yeast-D2 are spread over 150 true complexes.

6.2 Parameter setting

The parameters of the proposed EA with GO-based mutation and the counterpart algorithms are set to the following, more or less, standard settings: The population size, μ, is set to 100. The maximum number of generations required to stop the evolution of the algorithms is set at 100. The probability of uniform crossover, P_x, is set to 0.8. The probability of the canonical and the topological-based mutation operators, P_m, is set to 0.2; and the probability of the GO-based mutation operator, P_m, is likewise set to 0.2. The results of the algorithms are presented for the average of 30 simulation runs for the best solutions obtained (in terms of QD). To easily follow the competitive performance of the effective algorithm, the effective values are designated in boldface.

Validation measures are used to measure the accuracy of the predicted complexes. If a predicted complex $\mathcal{C}_i$ matches one of the golden complexes from the benchmark set in $\mathcal{S}^*$ (i.e. complex $\mathcal{S}_j$), we can say that the proteins of both complexes are overlapped or intersected with a neighborhood affinity score (Eq. 24). If the score is found to be equal to or greater than an overlapping threshold or overlapping score σ_{OS}, then complex $\mathcal{C}_i$ matches $\mathcal{S}_j$ [25].

$$NA(C_i, S_j) = \frac{|C_i \cap S_j|}{|C_i||S_j|} \quad (24)$$

where $|.|$ denotes the overlap between the predicted complex and the golden complex in terms of the number of proteins they share.

$$match\,(C_i, S_j) = \begin{cases} 1 \;\; if \;\; NA(C_i, S_j) \geq \sigma_{OS} \\ 0 \;\; otherwisex \end{cases} \quad (25)$$

6.3 Quantity of the detected complexes

The ratio of the number of reference complexes that overlap with any of the detected complexes (given an overlapping score) is defined as $recall$. It is used to evaluate the quantity of matched true complexes. Also, the ratio of the number of detected complexes that overlap any of the true complexes is defined as $precision$. Finally, F score, is defined as the harmonic mean of both $recall$ and $precision$. Thus, F score is used to evaluate the overall quantity of matched complexes.

$$recall = \frac{|S_i|S_i \in S^* \wedge \exists C_j \in C \rightarrow match(S_i, C_j)}{K^*} \quad (26)$$

$$precision = \frac{|C_i|C_i \in C \wedge \exists S_j \in S^* \rightarrow match(C_i, S_j)}{K} \quad (27)$$

$$F = \frac{2 \times recall \times precision}{recall + precision} \quad (28)$$

The results reported in Tables 1-4 and in Tables 5 and 8 for, respectively, Yeast-D1 and Yeast-D2.

Table 1: Performance comparison for **Yeast-D1** in terms of $Recall$, $Precision$, and F score for an average of 30 different runs. The results are reported for the proposed EA with GO-based mutation ($\boldsymbol{EA_{GOm1}}$) and Jaccard functional similarity against the canonical EA ($\boldsymbol{EA}$).

σ_{OS}	$Recall$		$Precision$		F	
	EA	EA_{GOm1}	EA	EA_{GOm1}	EA	EA_{GOm1}
0.10	0.9287	0.9513	0.7813	0.7993	0.8484	0.8686
0.15	0.8765	0.9184	0.7446	0.7726	0.8050	0.8391
0.20	0.8361	0.8838	0.7374	0.7680	0.7814	0.8217
0.25	0.8056	0.8551	0.7299	0.7671	0.7656	0.8086
0.30	0.7761	0.8188	0.7080	0.7622	0.7402	0.7893
0.35	0.7406	0.7944	0.6818	0.7534	0.7098	0.7732
0.40	0.7098	0.7739	0.6652	0.7492	0.6866	0.7613
0.45	0.6821	0.7547	0.6400	0.7367	0.6602	0.7455
0.50	0.6624	0.7333	0.6247	0.7174	0.6429	0.7252
0.55	0.6201	0.7111	0.5824	0.6957	0.6005	0.7032
0.60	0.5979	0.6932	0.5616	0.6780	0.5790	0.6854
0.65	0.5774	0.6752	0.5425	0.6605	0.5592	0.6677
0.70	0.5483	0.6556	0.5152	0.6413	0.5311	0.6482
0.75	0.5111	0.6111	0.4801	0.5978	0.4950	0.6043
0.80	0.4782	0.5812	0.4494	0.5686	0.4632	0.5747

Table 2: Performance comparison for **Yeast-D1** in terms of *Recall*, *Precision,* and *F* score for an average of 30 different runs. The results are reported for the proposed EA with GO-based mutation ($\boldsymbol{EA_{GOm2}}$) and Maximum functional similarity against the canonical EA ($\boldsymbol{EA}$).

σ_{OS}	*Recall*		*Precision*		*F*	
	EA	EA_{GOm2}	EA	EA_{GOm2}	EA	EA_{GOm2}
0.10	0.9287	0.9483	0.7813	0.7967	0.8484	0.8655
0.15	0.8765	0.9047	0.7446	0.7707	0.8050	0.8474
0.20	0.8361	0.8547	0.7374	0.7703	0.7814	0.8316
0.25	0.8056	0.8325	0.7299	0.7703	0.7656	0.8198
0.30	0.7761	0.8060	0.7080	0.7685	0.7402	0.8088
0.35	0.7406	0.7850	0.6818	0.7630	0.7098	0.8036
0.40	0.7098	0.7671	0.6652	0.7595	0.6866	0.7890
0.45	0.6821	0.7474	0.6400	0.7446	0.6602	0.7568
0.50	0.6624	0.7333	0.6247	0.7313	0.6429	0.7282
0.55	0.6201	0.7073	0.5824	0.7053	0.6005	0.6808
0.60	0.5979	0.6957	0.5616	0.6939	0.5790	0.6663
0.65	0.5774	0.6816	0.5425	0.6797	0.5592	0.6387
0.70	0.5483	0.6667	0.5152	0.6648	0.5311	0.6230
0.75	0.5111	0.6132	0.4801	0.6115	0.4950	0.5450
0.80	0.4782	0.5769	0.4494	0.5752	0.4632	0.5214

Table 3: Performance comparison for **Yeast-D1** in terms of *Recall*, *Precision,* and *F* score for an average of 30 different runs. The results are reported for the proposed EA with GO-based mutation ($\boldsymbol{EA_{GOm1}}$) and Jaccard functional similarity against the topological-based EA ($\boldsymbol{EA_{Top}}$).

σ_{OS}	*Recall*		*Precision*		*F*	
	EA_{Top}	EA_{GOm1}	EA_{Top}	EA_{GOm1}	EA_{Top}	EA_{GOm1}
0.10	0.9496	0.9513	0.7764	0.7993	0.8542	0.8686
0.15	0.9064	0.9184	0.7433	0.7726	0.8167	0.8391
0.20	0.8650	0.8838	0.7425	0.7680	0.7989	0.8217
0.25	0.8355	0.8551	0.7413	0.7671	0.7854	0.8086
0.30	0.8021	0.8188	0.7388	0.7622	0.7690	0.7893
0.35	0.7812	0.7944	0.7313	0.7534	0.7553	0.7732
0.40	0.7620	0.7739	0.7293	0.7492	0.7451	0.7613
0.45	0.7496	0.7547	0.7242	0.7367	0.7366	0.7455
0.50	0.7231	0.7333	0.6999	0.7174	0.7112	0.7252
0.55	0.7077	0.7111	0.65850	0.6957	0.6960	0.7032
0.60	0.6987	0.6932	0.6762	0.6780	0.6872	0.6854
0.65	0.6795	0.6752	0.6576	0.6605	0.6682	0.6677
0.70	0.6675	0.6556	0.6459	0.6413	0.6564	0.6482
0.75	0.6222	0.6111	0.6022	0.5978	0.6119	0.6043
0.80	0.5893	0.5812	0.5704	0.5686	0.5796	0.5747

Table 4: Performance comparison for **Yeast-D1** in terms of *Recall*, *Precision,* and *F* score for an average of 30 different runs. The results are reported for the proposed EA with GO-based mutation ($\boldsymbol{EA_{GOm2}}$) and Maximum functional similarity against the topological-based EA ($\boldsymbol{EA_{Top}}$).

σ_{OS}	***Recall***		***Precision***		***F***	
	$\boldsymbol{EA_{Top}}$	$\boldsymbol{EA_{GOm2}}$	$\boldsymbol{EA_{Top}}$	$\boldsymbol{EA_{GOm2}}$	$\boldsymbol{EA_{Top}}$	$\boldsymbol{EA_{GOm2}}$
0.10	0.9496	0.9483	0.7764	0.7967	0.8542	0.8655
0.15	0.9064	0.9047	0.7433	0.7707	0.8167	0.8474
0.20	0.8650	0.8547	0.7425	0.7703	0.7989	0.8316
0.25	0.8355	0.8325	0.7413	0.7703	0.7854	0.8198
0.30	0.8021	0.8060	0.7388	0.7685	0.7690	0.8088
0.35	0.7812	0.7850	0.7313	0.7630	0.7553	0.8036
0.40	0.7620	0.7671	0.7293	0.7595	0.7451	0.7890
0.45	0.7496	0.7474	0.7242	0.7446	0.7366	0.7568
0.50	0.7231	0.7333	0.6999	0.7313	0.7112	0.7282
0.55	0.7077	0.7073	0.65850	0.7053	0.6960	0.6808
0.60	0.6987	0.6957	0.6762	0.6939	0.6872	0.6663
0.65	0.6795	0.6816	0.6576	0.6797	0.6682	0.6387
0.70	0.6675	0.6667	0.6459	0.6648	0.6564	0.6230
0.75	0.6222	0.6132	0.6022	0.6115	0.6119	0.5450
0.80	0.5893	0.5769	0.5704	0.5752	0.5796	0.5214

Table 5: Performance comparison for **Yeast-D2** in terms of *Recall*, *Precision,* and *F* score for an average of 30 different runs. The results are reported for the proposed EA with a GO-based mutation ($\boldsymbol{EA_{GOm1}}$) and Jaccard functional similarity against the canonical EA ($\boldsymbol{EA}$).

σ_{OS}	*Recall*		*Precision*		*F*	
	EA	EA_{GOm1}	EA	EA_{GOm1}	EA	EA_{GOm1}
0.10	0.9709	0.9840	0.6175	0.7993	0.7547	0.6591
0.15	0.9113	0.9347	0.5822	0.7726	0.7103	0.6493
0.20	0.8449	0.8602	0.5521	0.7680	0.6676	0.6254
0.25	0.7736	0.8016	0.5003	0.7671	0.6074	0.5959
0.30	0.7009	0.7413	0.4760	0.7622	0.5667	0.5754
0.35	0.6353	0.6631	0.4443	0.7534	0.5227	0.5433
0.40	0.5909	0.6280	0.4279	0.7492	0.4961	0.5107
0.45	0.5236	0.5567	0.3969	0.7367	0.4513	0.4649
0.50	0.4991	0.5311	0.3886	0.7174	0.4368	0.4290
0.55	0.4240	0.4587	0.3406	0.6957	0.3775	0.3946
0.60	0.3853	0.4222	0.3196	0.6780	0.3492	0.3657
0.65	0.3413	0.3933	0.2990	0.6605	0.3185	0.3381
0.70	0.2904	0.3136	0.2632	0.6413	0.2760	0.2690
0.75	0.2584	0.2976	0.2369	0.5978	0.2470	0.2434
0.80	0.2211	0.2704	0.2124	0.5686	0.2165	0.2154

Table 6: Performance comparison for **Yeast-D2** in terms of *Recall*, *Precision*, and *F* score for an average of 30 different runs. The results are reported for the proposed EA with GO-based mutation ($\boldsymbol{EA_{GOm2}}$) and Maximum functional similarity against the canonical EA ($\boldsymbol{EA}$).

σ_{OS}	*Recall*		*Precision*		*F*	
	EA	EA_{GOm2}	EA	EA_{GO2}	EA	EA_{GOm2}
0.10	0.9709	0.9804	0.6175	0.6085	0.7547	0.7509
0.15	0.9113	0.9198	0.5822	0.5931	0.7103	0.7210
0.20	0.8449	0.8404	0.5521	0.5708	0.6676	0.6797
0.25	0.7736	0.7851	0.5003	0.5273	0.6074	0.6307
0.30	0.7009	0.7269	0.4760	0.5181	0.5667	0.6049
0.35	0.6353	0.6533	0.4443	0.4902	0.5227	0.5599
0.40	0.5909	0.6091	0.4279	0.4716	0.4961	0.5315
0.45	0.5236	0.5433	0.3969	0.4425	0.4513	0.4876
0.50	0.4991	0.5158	0.3886	0.4296	0.4368	0.4686
0.55	0.4240	0.4591	0.3406	0.4039	0.3775	0.4296
0.60	0.3853	0.4282	0.3196	0.3905	0.3492	0.4084
0.65	0.3413	0.3987	0.2990	0.3695	0.3185	0.3834
0.70	0.2904	0.3180	0.2632	0.3305	0.2760	0.3240
0.75	0.2584	0.3011	0.2369	0.3134	0.2470	0.3071
0.80	0.2211	0.2733	0.2124	0.2882	0.2165	0.2805

Table 7: Performance comparison for **Yeast-D2** in terms of *Recall*, *Precision*, and *F* score for an average of 30 different runs. The results are reported for the proposed EA with GO-based mutation ($\boldsymbol{EA_{GOm1}}$) and Jaccard functional similarity against the topological-based EA ($\boldsymbol{EA_{Top}}$).

σ_{OS}	*Recall*		*Precision*		*F*	
	EA_{Top}	EA_{GOm1}	EA_{Top}	EA_{GO1}	EA_{Top}	EA_{GOm1}
0.10	0.9853	0.9840	0.6183	0.7993	0.7597	0.6591
0.15	0.9318	0.9347	0.5906	0.7726	0.7229	0.6493
0.20	0.8624	0.8602	0.5674	0.7680	0.6843	0.6254
0.25	0.7987	0.8016	0.5098	0.7671	0.6222	0.5959
0.30	0.7327	0.7413	0.4997	0.7622	0.5940	0.5754
0.35	0.6684	0.6631	0.4764	0.7534	0.5562	0.5433
0.40	0.6216	0.6280	0.4593	0.7492	0.5281	0.5107
0.45	0.5520	0.5567	0.4289	0.7367	0.4825	0.4649
0.50	0.5271	0.5311	0.4212	0.7174	0.4681	0.4290
0.55	0.4569	0.4587	0.3827	0.6957	0.4163	0.3946
0.60	0.4222	0.4222	0.3646	0.6780	0.3911	0.3657
0.65	0.3807	0.3933	0.3446	0.6605	0.3616	0.3381
0.70	0.3062	0.3136	0.2958	0.6413	0.3008	0.2690
0.75	0.2909	0.2976	0.2840	0.5978	0.2873	0.2434
0.80	0.2618	0.2704	0.2618	0.5686	0.2617	0.2154

Table 8: Performance comparison for **Yeast-D2** in terms of $Recall$, $Precision$, and F score for an average of 30 different runs. The results are reported for the proposed EA with GO-based mutation ($\boldsymbol{EA_{GOm2}}$) and Maximum functional similarity against the topological-based EA ($\boldsymbol{EA_{Top}}$).

σ_{OS}	$Recall$		$Precision$		F	
	EA_{Top}	EA_{GOm2}	EA_{Top}	EA_{GOm2}	EA_{Top}	EA_{GOm2}
0.10	0.9853	0.9804	0.6183	0.6085	0.7597	0.7509
0.15	0.9318	0.9198	0.5906	0.5931	0.7229	0.7210
0.20	0.8624	0.8404	0.5674	0.5708	0.6843	0.6797
0.25	0.7987	0.7851	0.5098	0.5273	0.6222	0.6307
0.30	0.7327	0.7269	0.4997	0.5181	0.5940	0.6049
0.35	0.6684	0.6533	0.4764	0.4902	0.5562	0.5599
0.40	0.6216	0.6091	0.4593	0.4716	0.5281	0.5315
0.45	0.5520	0.5433	0.4289	0.4425	0.4825	0.4876
0.50	0.5271	0.5158	0.4212	0.4296	0.4681	0.4686
0.55	0.4569	0.4591	0.3827	0.4039	0.4163	0.4296
0.60	0.4222	0.4282	0.3646	0.3905	0.3911	0.4084
0.65	0.3807	0.3987	0.3446	0.3695	0.3616	0.3834
0.70	0.3062	0.3180	0.2958	0.3305	0.3008	0.3240
0.75	0.2909	0.3011	0.2840	0.3134	0.2873	0.3071
0.80	0.2618	0.2733	0.2618	0.2882	0.2617	0.2805

Tables 1–8 reveal that the added information being encapsulated by the GO terms in the design of the proposed algorithms obviously affects their performance to outperform the counterpart canonical and topological-based EAs in all evaluation metrics and in almost all overlapping scores. In other words, this can reflect that the modularity density with the added GO information has the ability to detect a greater number of accurate protein complexes in both yeast-D1 and D2. This indeed implies that the structure of the detected complexes has a high percentage of correct proteins being aggregated into the correct complex set. Further, the results reveal the ability of the proposed algorithms to reach a compromise or tradeoff between the contradictory objectives of *recall* and *precision*. This can be clearly seen from the higher values of F score achieved by the proposed algorithms.

6.4 Quality of the detected complexes

$Recall_N$ $Precision_N$ and F_N are used in evaluating the quality of the detected complexes [15]. Further, positive predictive value (PPV), sensitivity ($sensitivity$), and geometric accuracy ($accuracy$) [15] are also used to measure the quality of the detected complexes.

$$recall_N = \frac{\sum_{i=1}^{K_S} |m_i|}{\sum_{j=1}^{K_S} |S_i|} \text{ where } |m_i| = max_{|C_i \cap S_j|} \{\forall\, S_j \in\ S^* \wedge match(S_i, C_j) \geq \sigma_{OS}\} \quad (29)$$

$$precision_N = \frac{\sum_{i=1}^{K_C} |m_i|}{\sum_{i=1}^{K_C} |C_i|} \text{ where } |m_i| = max_{|C_j \cap S_i|} \{\forall\, C_j \in\ C^* \wedge match(C_j, S_i) \geq \sigma_{OS}\} \quad (30)$$

$$F_N = \frac{2 \times recall_N \times precision_N}{recall_n + precision_N} \quad (31)$$

$$PPV = \frac{\sum_{j=1}^{K_C} max_{i=1}^{K_S} t_{ij}}{\sum_{j=1}^{K_C} \sum_{i=1}^{K_S} t_{ij}} \quad (32)$$

$$sensitivity = \frac{\sum_{i=1}^{K_S} max_{j=1}^{K_C} t_{ij}}{\sum_{i=1}^{K_S} |S_i|} \quad (33)$$

where t_{ij} is the number of proteins common to the golden standard complex i and the predicted complex j. The $accuracy$ is used to evaluate the overall performance based on the quality of the matched complexes.

$$accuracy = \sqrt{sensitivity * PPV} \quad (34)$$

The results in Tables 9–16 prove the ability of the proposed GO-based EAs to outperform (in all terms of quantity metrics, $Recall_N$, $Precision_N$ and F_N) the canonical and topological EAs. Table 17 shows that the addition of gene ontology information can yield better results. However, further development may be needed.

Table 9: Performance comparison for **Yeast-D1** in terms of $Recall_N$, $Precision_N$, and F_N for an average of 30 different runs. The results are reported for the proposed EA with a GO-based mutation ($\boldsymbol{EA_{GOm1}}$) and Jaccard functional similarity against the canonical EA ($\boldsymbol{EA}$).

σ_{OS}	$Recall_N$		$Precision_N$		F_N	
	EA	EA_{GOm1}	EA	EA_{GOm1}	EA	EA_{GOm1}
0.10	0.8540	0.9447	0.7343	0.8000	0.7894	0.8663
0.15	0.8229	0.9152	0.7297	0.7969	0.7731	0.8519
0.20	0.7942	0.8881	0.7269	0.7959	0.7587	0.8394
0.25	0.7692	0.8624	0.7217	0.7952	0.7445	0.8274
0.30	0.7391	0.8266	0.7039	0.7935	0.7209	0.8096
0.35	0.7097	0.8127	0.6844	0.7887	0.6967	0.8005
0.40	0.6697	0.7903	0.6624	0.7846	0.6659	0.7874
0.45	0.6332	0.7642	0.6295	0.7629	0.6313	0.7636
0.50	0.5998	0.7161	0.6013	0.7161	0.6005	0.7161
0.55	0.5497	0.6953	0.5497	0.6953	0.5497	0.6953
0.60	0.5189	0.6728	0.5189	0.6728	0.5189	0.6728
0.65	0.4893	0.6308	0.4893	0.6308	0.4893	0.6308
0.70	0.4547	0.6096	0.4547	0.6096	0.4547	0.6096
0.75	0.4222	0.5405	0.4222	0.5405	0.4222	0.5405
0.80	0.3992	0.5229	0.3992	0.5229	0.3992	0.5229

Table 10: Performance comparison for **Yeast-D1** in terms of $Recall_N$, $Precision_N$, and F_N for an average of 30 different runs. The results are reported for the proposed EA with GO-based mutation ($\boldsymbol{EA_{GOm2}}$) and Maximum functional similarity against the canonical EA ($\boldsymbol{EA}$).

σ_{OS}	$Recall_N$		$Precision_N$		F_N	
	EA	EA_{GOm2}	EA	EA_{GOm2}	EA	EA_{GOm2}
0.10	0.8540	0.9464	0.7343	0.7974	0.7894	0.8655
0.15	0.8229	0.9081	0.7297	0.7945	0.7731	0.8474
0.20	0.7942	0.8727	0.7269	0.7944	0.7587	0.8316
0.25	0.7692	0.8470	0.7217	0.7944	0.7445	0.8198
0.30	0.7391	0.8250	0.7039	0.7934	0.7209	0.8088
0.35	0.7097	0.8154	0.6844	0.7922	0.6967	0.8036
0.40	0.6697	0.7913	0.6624	0.7867	0.6659	0.7890
0.45	0.6332	0.7573	0.6295	0.7564	0.6313	0.7568
0.50	0.5998	0.7282	0.6013	0.7282	0.6005	0.7282
0.55	0.5497	0.6808	0.5497	0.6808	0.5497	0.6808
0.60	0.5189	0.6663	0.5189	0.6663	0.5189	0.6663
0.65	0.4893	0.6387	0.4893	0.6387	0.4893	0.6387
0.70	0.4547	0.6230	0.4547	0.6230	0.4547	0.6230
0.75	0.4222	0.5450	0.4222	0.5450	0.4222	0.5450
0.80	0.3992	0.5214	0.3992	0.5214	0.3992	0.5214

Table 11: Performance comparison for **Yeast-D1** in terms of $Recall_N$, $Precision_N$, and F_N for an average of 30 different runs. The results are reported for the proposed EA with GO-based mutation ($\boldsymbol{EA_{GOm1}}$) and Jaccard functional similarity against the topological-based EA ($\boldsymbol{EA_{Top}}$).

σ_{OS}	$Recall_N$		$Precision_N$		F_N	
	EA_{Top}	EA_{GOm1}	EA_{Top}	EA_{GOm1}	EA_{Top}	EA_{GOm1}
0.10	0.9441	0.9447	0.7893	0.8000	0.8597	0.8663
0.15	0.9042	0.9152	0.7854	0.7969	0.8406	0.8519
0.20	0.8786	0.8881	0.7853	0.7959	0.8293	0.8394
0.25	0.8523	0.8624	0.7844	0.7952	0.8169	0.8274
0.30	0.8174	0.8266	0.7825	0.7935	0.7995	0.8096
0.35	0.8054	0.8127	0.7787	0.7887	0.7918	0.8005
0.40	0.7804	0.7903	0.7749	0.7846	0.7777	0.7874
0.45	0.7629	0.7642	0.7620	0.7629	0.7624	0.7636
0.50	0.6927	0.7161	0.6927	0.7161	0.6927	0.7161
0.55	0.6741	0.6953	0.6741	0.6953	0.6741	0.6953
0.60	0.6595	0.6728	0.6595	0.6728	0.6595	0.6728
0.65	0.6229	0.6308	0.6229	0.6308	0.6229	0.6308
0.70	0.6062	0.6096	0.6062	0.6096	0.6062	0.6096
0.75	0.5403	0.5405	0.5403	0.5405	0.5403	0.5405
0.80	0.5201	0.5229	0.5201	0.5229	0.5201	0.5229

Table 12: Performance comparison for **Yeast-D1** in terms of $Recall_N$, $Precision_N$, and F_N for an average of 30 different runs. The results are reported for the proposed EA with GO-based mutation ($\boldsymbol{EA_{GOm2}}$) and Maximum functional similarity against the topological-based EA ($\boldsymbol{EA_{Top}}$).

σ_{OS}	$Recall_N$		$Precision_N$		F_N	
	EA_{Top}	EA_{GOm2}	EA_{Top}	EA_{GOm2}	EA_{Top}	EA_{GOm2}
0.10	0.9441	0.9464	0.7893	0.7974	0.8597	0.8655
0.15	0.9042	0.9081	0.7854	0.7945	0.8406	0.8474
0.20	0.8786	0.8727	0.7853	0.7944	0.8293	0.8316
0.25	0.8523	0.8470	0.7844	0.7944	0.8169	0.8198
0.30	0.8174	0.8250	0.7825	0.7934	0.7995	0.8088
0.35	0.8054	0.8154	0.7787	0.7922	0.7918	0.8036
0.40	0.7804	0.7913	0.7749	0.7867	0.7777	0.7890
0.45	0.7629	0.7573	0.7620	0.7564	0.7624	0.7568
0.50	0.6927	0.7282	0.6927	0.7282	0.6927	0.7282
0.55	0.6741	0.6808	0.6741	0.6808	0.6741	0.6808
0.60	0.6595	0.6663	0.6595	0.6663	0.6595	0.6663
0.65	0.6229	0.6387	0.6229	0.6387	0.6229	0.6387
0.70	0.6062	0.6230	0.6062	0.6230	0.6062	0.6230
0.75	0.5403	0.5450	0.5403	0.5450	0.5403	0.5450
0.80	0.5201	0.5214	0.5201	0.5214	0.5201	0.5214

Table 13: Performance comparison for **Yeast-D2** in terms of $Recall_N$, $Precision_N$, and F_N for an average of 30 different runs. The results are reported for the proposed EA with a GO-based mutation ($\boldsymbol{EA_{GOm1}}$) and Jaccard functional similarity against the canonical EA ($\boldsymbol{EA}$).

σ_{OS} /.,mn	$Recall_N$		$Precision_N$		F_N	
	EA	EA_{GOm1}	EA	EA_{GOm1}	EA	EA_{GOm1}
0.10	0.5538	0.5931	0.7217	0.7594	0.6265	0.6659
0.15	0.5395	0.5822	0.7113	0.7551	0.6134	0.6574
0.20	0.5051	0.5503	0.6955	0.7458	0.5849	0.6332
0.25	0.4728	0.5218	0.6684	0.7287	0.5536	0.6080
0.30	0.4386	0.4964	0.6415	0.7218	0.5207	0.5885
0.35	0.4103	0.4601	0.6184	0.6925	0.4930	0.5529
0.40	0.3703	0.4344	0.5864	0.6710	0.4537	0.5273
0.45	0.3314	0.3774	0.5542	0.6444	0.4146	0.4759
0.50	0.3005	0.3456	0.5323	0.6190	0.3839	0.4435
0.55	0.2505	0.3021	0.4873	0.5887	0.3305	0.3992
0.60	0.2247	0.2779	0.4571	0.5630	0.3009	0.3721
0.65	0.1980	0.2528	0.4215	0.5283	0.2689	0.3418
0.70	0.1618	0.1915	0.3752	0.4876	0.2256	0.2749
0.75	0.1445	0.1730	0.3401	0.4369	0.2025	0.2478
0.80	0.1188	0.1529	0.2880	0.3832	0.1679	0.2185

Table 14: Performance comparison for **Yeast-D2** in terms of $Recall_N$, $Precision_N$, and F_N for an average of 30 different runs. The results are reported for the proposed EA with GO-based mutation ($\boldsymbol{EA_{GOm2}}$) and Maximum functional similarity against the canonical EA ($\boldsymbol{EA}$).

σ_{OS}	$Recall_N$		$Precision_N$		F_N	
	EA	EA_{GOm2}	EA	EA_{GOm2}	EA	EA_{GOm2}
0.10	0.5538	0.5909	0.7217	0.7453	0.6265	0.6591
0.15	0.5395	0.5769	0.7113	0.7424	0.6134	0.6493
0.20	0.5051	0.5454	0.6955	0.7333	0.5849	0.6254
0.25	0.4728	0.5120	0.6684	0.7127	0.5536	0.5959
0.30	0.4386	0.4885	0.6415	0.6999	0.5207	0.5754
0.35	0.4103	0.4541	0.6184	0.6764	0.4930	0.5433
0.40	0.3703	0.4189	0.5864	0.6542	0.4537	0.5107
0.45	0.3314	0.3687	0.5542	0.6291	0.4146	0.4649
0.50	0.3005	0.3341	0.5323	0.5998	0.3839	0.4290
0.55	0.2505	0.2984	0.4873	0.5823	0.3305	0.3946
0.60	0.2247	0.2709	0.4571	0.5631	0.3009	0.3657
0.65	0.1980	0.2501	0.4215	0.5218	0.2689	0.3381
0.70	0.1618	0.1877	0.3752	0.4741	0.2256	0.2690
0.75	0.1445	0.1702	0.3401	0.4276	0.2025	0.2434
0.80	0.1188	0.1506	0.2880	0.3787	0.1679	0.2154

Table 15: Performance comparison for **Yeast-D2** in terms of $Recall_N$, $Precision_N$, and F_N for an average of 30 different runs. The results are reported for the proposed EA with GO-based mutation ($\boldsymbol{EA_{GOm1}}$) and Jaccard functional similarity against the topological-based EA ($\boldsymbol{EA_{Top}}$).

σ_{OS}	$Recall_N$		$Precision_N$		F_N	
	EA_{Top}	EA_{GOm1}	EA_{Top}	EA_{GOm1}	EA_{Top}	EA_{GOm1}
0.10	0.5878	0.5931	0.7506	0.7453	0.6592	0.6659
0.15	0.5764	0.5822	0.7448	0.7424	0.6498	0.6574
0.20	0.5427	0.5503	0.7364	0.7333	0.6247	0.6332
0.25	0.5125	0.5218	0.7151	0.7127	0.5970	0.6080
0.30	0.4855	0.4964	0.7051	0.6999	0.5750	0.5885
0.35	0.4598	0.4601	0.6817	0.6764	0.5491	0.5529
0.40	0.4183	0.4344	0.6535	0.6542	0.5100	0.5273
0.45	0.3710	0.3774	0.6255	0.6291	0.4657	0.4759
0.50	0.3390	0.3456	0.6068	0.5998	0.4349	0.4435
0.55	0.2968	0.3021	0.5786	0.5823	0.3922	0.3992
0.60	0.2679	0.2779	0.5568	0.5631	0.3616	0.3721
0.65	0.2430	0.2528	0.5272	0.5218	0.3325	0.3418
0.70	0.1842	0.1915	0.4608	0.4741	0.2631	0.2749
0.75	0.1730	0.1730	0.4347	0.4276	0.2474	0.2478
0.80	0.1490	0.1529	0.3756	0.3787	0.2133	0.2185

Table 16: Performance comparison for **Yeast-D2** in terms of $Recall_N$, $Precision_N$, and F_N for an average of 30 different runs. The results are reported for the proposed EA with GO-based mutation ($\boldsymbol{EA_{GOm1}}$) and Maximum functional similarity against the topological-based EA ($\boldsymbol{EA_{Top}}$).

σ_{OS}	$Recall_N$		$Precision_N$		F_N	
	EA_{Top}	EA_{GOm2}	EA_{Top}	EA_{GOm2}	EA_{Top}	EA_{GOm2}
0.10	0.5878	0.5909	0.7506	0.7453	0.6592	0.6591
0.15	0.5764	0.5769	0.7448	0.7424	0.6498	0.6493
0.20	0.5427	0.5454	0.7364	0.7333	0.6247	0.6254
0.25	0.5125	0.5120	0.7151	0.7127	0.5970	0.5959
0.30	0.4855	0.4885	0.7051	0.6999	0.5750	0.5754
0.35	0.4598	0.4541	0.6817	0.6764	0.5491	0.5433
0.40	0.4183	0.4189	0.6535	0.6542	0.5100	0.5107
0.45	0.3710	0.3687	0.6255	0.6291	0.4657	0.4649
0.50	0.3390	0.3341	0.6068	0.5998	0.4349	0.4290
0.55	0.2968	0.2984	0.5786	0.5823	0.3922	0.3946
0.60	0.2679	0.2709	0.5568	0.5631	0.3616	0.3657
0.65	0.2430	0.2501	0.5272	0.5218	0.3325	0.3381
0.70	0.1842	0.1877	0.4608	0.4741	0.2631	0.2690
0.75	0.1730	0.1702	0.4347	0.4276	0.2474	0.2434
0.80	0.1490	0.1506	0.3756	0.3787	0.2133	0.2154

Table 17: Performance comparison for **Yeast-D1** and **Yeast-D2** in terms of $Sensitivity$, PPV, and $accuracy$ for an average of 30 different runs. The results are reported for the proposed EAs with GO-based mutation ($\boldsymbol{EA_{GOm1}}$) and Jaccard functional similarity and EAs with GO-based mutation ($\boldsymbol{EA_{GOm2}}$) and Maximum functional similarity against the canonical EA ($\boldsymbol{EA}$) and the topological-based EA ($\boldsymbol{EA_{Top}}$).

	$Sensitivity$	PPV	$accuracy$
		Yeast-D1	
EA	0.8937	0.7363	0.8109
EA_{Top}	0.9646	0.7904	0.8732
EA_{GOm1}	0.9625	**0.8004**	**0.8777**
EA_{GOm2}	**0.9662**	**0.7977**	**0.8779**
		Yeast-D2	
EA	0.5648	0.2912	0.4229
EA_{Top}	0.5933	**0.4269**	**0.4269**
EA_{GOm1}	**0.5978**	0.3025	0.4253
EA_{GOm2}	**0.5972**	0.3000	0.4233

7. Conclusions

The main contribution of this paper is to improve the detection reliability of the well-known modularity density model when used as the optimization model in the framework of an evolutionary-based complex detection algorithm. To this end, the design of the EA is extended by adding a gene ontology-based mutation operator. With Jaccard and maximum functional similarity, the GO information of gene products is injected into the mechanism of the mutation operator. On two yeast PPINs and two benchmark sets of gold complexes, the proposed EA is proven to produce more accurate complexes with more accurate quality than the counterpart canonical and topological-based EAs. According to the results, a gene ontology-based mutation operator complements modularity density well, allowing for the

discovery of additional complexes. Further investigation and future work are recommended to improve the quality of the detected complexes in terms of $Sensitivity$, PPV, and $accuracy$. This would open the door for redefining the modularity density model to cope with the biological domain rather than the topological domain. Also, more research investigations are required for detecting disease-related (e.g., bone, cancer, endocrine, and cardiovascular) complexes.

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] 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.

[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] 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.

[12] 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.

[13] 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.

[14] B. a. A. Attea, W. A. Hariz, M. F. Abdulhalim, "Improving the performance of evolutionary multi-objective co-clustering models for community detection in complex social networks," *Swarm and Evolutionary Computation*, vol. 26, pp. 137-156, 2016.

[15] B. a. 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.

[16] A. H. Abdulateef, A. A. Bara'a, A. N. Rashid, and M. Al-Ani, "A new evolutionary algorithm with locally assisted heuristic for complex detection in protein interaction networks," *Applied Soft Computing*, vol. 73, pp. 1004-1025, 2018.

[17] A. H. Abdulateef, A. A. Bara'a, 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.

[18] I. H. Abdulateef, D. A. J. Alzubaydi, and A. A. Bara'a, "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.

[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] J. Handl and J. Knowles, "An evolutionary approach to multiobjective clustering," *IEEE transactions on Evolutionary Computation*, vol. 11, no. 1, pp. 56-76, 2007.

[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] A.-C. Gavin *et al.*, "Proteome survey reveals modularity of the yeast cell machinery," *Nature*, vol. 440, no. 7084, pp. 631-636, 2006.

[24] 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.

[25] S. Brohee and J. Van Helden, "Evaluation of clustering algorithms for protein-protein interaction networks," *BMC bioinformatics*, vol. 7, no. 1, pp. 1-19, 2006.