**ISSN: 0067-2904**

# Heuristic Modularity for Complex Identification in Protein-Protein Interaction Networks

**Amenah H. Abdulateef[1,3], Bara'a A. Attea*[2], Ahmed N. Rashid[3]**
[1]Department of Computer Science, College of Education for Pure Science (Ibn al-Haitham), University of Baghdad, Iraq
[2] Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq
[3] Computers and Information Technology, University of Anbar, Anbar, Iraq

**Abstract**

Due to the significant role in understanding cellular processes, the decomposition of Protein-Protein Interaction (PPI) networks into essential building blocks, or complexes, has received much attention for functional bioinformatics research in recent years. One of the well-known bi-clustering descriptors for identifying communities and complexes in complex networks, such as PPI networks, is modularity function. The contribution of this paper is to introduce heuristic optimization models that can collaborate with the modularity function to improve its detection ability. The definitions of the formulated heuristics are based on nodes and different levels of their neighbor properties. The modularity function and the formulated heuristics are then injected into the mechanism of a single objective Evolutionary Algorithm (EA) tailored specifically to tackle the problem, and thus, to identify possible complexes from PPI networks. In the experiments, different overlapping scores are used to evaluate the detection accuracy in both complex and protein levels. According to the evaluation metrics, the results reveal that the introduced heuristics have the ability to harness the accuracy of the existing modularity while identifying protein complexes in the tested PPI networks.

**Keywords:** complex detection; graph partitioning; heuristic; modularity; Protein-Protein Interaction network.

# دالة النمطية الموجهه لتحديد المركبات في شبكات التفاعل بين البروتين والبروتين

**امنة هيثم[1,3]،براء علي عطية*[2]،احمد رشيد[3]**
[1]قسم علوم الحاسوب ، كلية التربية للعلوم الصرفة (ابن الهيثم) ، جامعة بغداد ، العراق
[2]قسم علوم الحاسوب ، كلية العلوم ، جامعة بغداد ، بغداد ، العراق
[3]أجهزة الكمبيوتر وتكنولوجيا المعلومات ، جامعة الأنبار ، الأنبار ، العراق

**الخلاصة**

نظرًا للدور المهم في فهم العمليات الخلوية ، فإن تحليل شبكات التفاعل بين البروتين والبروتين (PPI) إلى لبنات أساسية أو مجتمعات (مركبات) ، قد حظي في السنوات الأخيرة الى الكثير من الاهتمام في بحوث المعلوماتية الحيوية. أحدى الدوال الرياضية المعروفة للكشف عن المجتمعات في الشبكات المعقدة، مثل شبكات ال PPI، هي دالة النمطية. تتمثل مساهمة هذا البحث في تقديم نماذج تحسين موجهه أستكشافية التي يمكن أن تتعاون مع دالة النمطية التقليدية لتحسين قدرتها على تحديد المركبات. تعتمد تعريفات نماذج التحسين الموجهه المقترحة في هذا البحث على العقد (أي البروتينات) ومستويات مختلفة من خصائص

*Email: baraaali@yahoo.com

الجيران لهذه البروتينات. يتم بعد ذلك حقن دالة النمطية و نماذج التحسين الموجهه المصاغة في آلية خوارزمية تطورية (EA) مصممة خصيصًا لمعالجة المشكلة ، وبالتالي ، لتحديد المركبات المحتملة من شبكات ال PPI. في التجارب، يتم استخدام درجات تداخل مختلفة لتقييم دقة الكشف في كل من المستويات المعقدة والبروتين. وفقًا لمقاييس التقييم، تكشف النتائج أن نماذج التحسين الموجهه التي تم إدخالهم الى خوارزمية ال (EA) لديهم القدرة على تسخير دقة دالة النمطية التقليدية مع تحديد مركبات البروتين في شبكات ال PPI المختبرة.

## 1. Introduction

Many complex systems in all areas of science, including social science, politics, biology and medicine, can be represented as networks. Topological analyses of such complex networks are universal and provide insights in many science studies. Complex systems are usually organized in compartments, which have their own role and / or function. In the network representation, such compartments appear as sets of nodes with a high density of internal links, whereas links between compartments have a lower density. These subgraphs are called communities, or modules, and can occur in a wide variety of networked systems. Finding compartments may shed light on the organization of complex systems and on their function. Therefore, detecting communities in networks has become a fundamental problem in network science. Many methods have been developed, using tools and techniques from different disciplines like physics, applied mathematics, biology, computer and social sciences. However, it is still not clear which algorithms are reliable and shall to be used in applications [1].

As an increasing amount of protein–protein interaction (PPI) data becomes available, its computational interpretation has become an important problem in bioinformatics. Observations show that PPI networks possess invaluable evolutionary insights and information to understand various biological processes and cellular functions. However, prediction of protein complexes, like many other practical optimization problems, falls into the category of strongly NP-hard combinatorial optimization problems that can easily bewilder exact optimization algorithms [2], [3].

Complex network clustering is data clustering in dividing the interested entities into clusters or modules. However, clusters in complex networks are based on both the inter and intra connections densities, while clusters in data clustering are groups of points close to each other in a way forming several local optima. In the literature, modularity [3] and a series of follow-up models have been proposed to measure the quality of a set of predicted intra-dense and inter-sparse subgraphs in a graph. The majority of these works have been applied to community detection in social networks and to complex identification in PPI networks.

The main contribution of this paper is to develop a heuristic ground definitions for modularity that can improve its detection ability. Here we are motivated by the logical consequence of protein neighboring properties and how to exploit and couple such properties with modularity function. In this paper, a single objective Evolutionary Algorithm (EA) [4] is adopted to combine modularity (as an objective function) and the proposed heuristic approaches. The algorithm attempts, with aid of modularity, to identify the *global structure* of the complexes and with the aid of heuristic functions, to fine tune such complexes. In the experimental results, we show that coupling the proposed heuristic operator as exploiter to capture local structures of the solutions provided by modularity can significantly improve the detection performance of EA.

In the remainder of this paper, preliminary concepts relating to complex detection problem in PPI networks and the main interest in the literature towards solving complex detection problem in PPI networks are presented first. These are followed by a closer look into the formal development of the proposed evolutionary based complex detection framework together with the proposed heuristic operators. Experimental results are then provided to support the positive impact of the proposed heuristic definitions to further correct the complex structures of the well-known modularity function. The final section of this paper presents major conclusions and further directions of this work.

## 2. Related background

A complex network such as PPI network can be denoted by $\mathcal{N}$, of $n$ proteins and $m$ interactions. In other terms $\mathcal{N}$ is said to be with cardinality $n$ and volume $m$. Generally, $\mathcal{N}$ can be modeled as undirected graph $G = (V, E)$ of a set $V = \{v_1, v_2, \dots, v_n\}$ of $n$ vertices, and a set $E \subseteq V \times V = \{(v_i, v_j) \big| v_i, v_j \in V \; and \; i \neq j\}$ of $m$ edges. Note that throughout this paper, the terms: tie, edge, link,

connection, relation, and interaction are used interchangeably to denote any vertex pair $(v_i, v_j)$ in $E$. Also, let $\Omega$ be the space of all possible partitioning solutions for $\mathcal{N}$ and let $\mathcal{C} \in \Omega = \{C_1, \dots, C_K\}$ be a network partitioning solution belongs to the space $\Omega$ with $K$ partitions or divisions. Normally, any unsigned graph $G$ can be represented by a symmetric $n \times n$ adjacency matrix denoted by $A$. Rows and columns of $A$ are labeled with the vertices of $V$ and assigned with 1 in entry $(i, j)$ if vertex pair $(v_i, v_j)$ is in $E$, and set to 0 if $(v_i, v_j) \notin E$.

From the adjacency matrix $A$ a set of $n$ direct neighboring lists, $L = \{l_1, l_2, \dots, l_n\}$, can be formed. Each list $l_i$ in the set $L$ aggregates connections of all vertices with vertex $v_i \in V$. Thus, $|l_i| = \sum_{j=1}^{n}(i, j)$ and $|L| = \sum_{i=1}^{n}|l_i|$. Mathematically noted, $|l_i| = m_i$ is said to be the degree of $v_i$, while $m = |L|/2$ is said to be the volume of $G$. Furthermore, the strength of each node $i$ can be specified in more details by $|l_i| = \left|l_{i,intra}\right|_{i\in C} + \left|l_{i,inter}\right|_{i\in C}$, where $\left|l_{i,intra}\right|_{i\in C}$ and $\left|l_{i,inter}\right|_{i\in C}$ be the intra-strength and inter-strength of node $i$, respectively. Generalizing this to all nodes, implies $|L| = |L_{intra}| + |L_{inter}| = \sum_{i=1}^{n}\left|l_{i,intra}\right|_{i\in C} + \sum_{i=1}^{n}\left|l_{i,inter}\right|_{i\in C}$.

## 3 Modularity based co-clustering model

Co-clustering or simultaneous matrix partitioning (in contrast to clustering, as depicted in Figure-1) needs a quality function that can capture the embedded *distinct* sub-matrix structures. The *modularity* (normally noted as $Q$) model defined after Newman and Girvan, lays the foundation of many existing successful graph clustering algorithms [5], [6]. The purpose of $Q$ is to capture the hidden structure of sub-graphs or community sets in complex networks by maximizing intra-cluster links while minimizing inter-cluster ones.

Consider partitioning $V$ of $G$ into a co-clustering solution $\mathcal{C} = \{C_1, \dots, C_K\}$ such that each vertex $v_i, 1 \le i \le n$ is exactly assigned to one cluster $C_j, 1 \le j \le K$. The impact of $E$ in $\mathcal{C}$ is quantified in two distinct terms. The set of edges between vertices existing in two distinct clusters: $E(C_i, C_j), 1 \le i, j \le K \; and \; i \ne j$ and the set of edges found inside one cluster: $E(C_i, C_i), 1 \le i \le K$. Then, modularity will award $\mathcal{C}$ according to the fraction of connections inside its communities as formulated in Eq. 1. The left term in Eq. 1 biases towards a solution $\mathcal{C}$ that is covered with a set of densely intra-connected modules, i.e. many edges fall within each sub-graph $\{C_1, \dots, C_K\}$. On the other hand, the right term in Eq. 1 expresses that the expected value of the same edge density in $\mathcal{C}$ with the same community structure $\{C_1, \dots, C_K\}$ but fall at random between the vertices should be small. $Q$ will approach its minimum at 0 if the number of within-community edges is no better than random. On the other hand, values approaching $Q = 1$, which is the maximum, indicate strong community structure.

$$Q(\mathcal{C}) = \sum_{i=1}^{K}\left[\frac{|E(C_i, C_i)|}{m(\mathcal{C})} - \left(\frac{\sum_{v\in C_i} m(v)}{2m(\mathcal{C})}\right)^2\right] \quad (1)$$

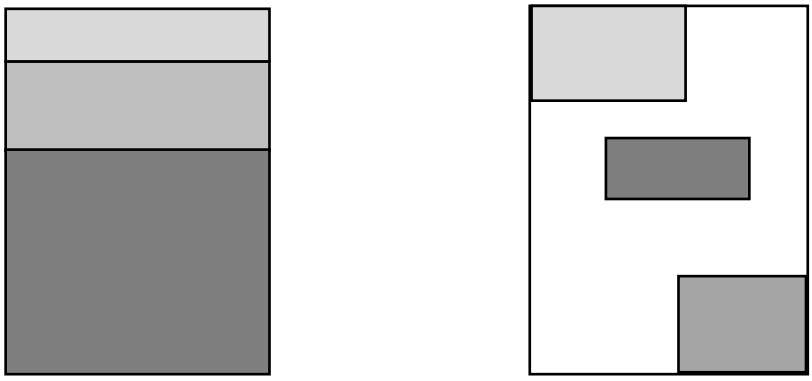

**Figure 1**-Clustering against co-clustering. Left: clustering means partitioning all data vectors with all their features into $k$ (sometimes unknown) disjoint groups. Right: co-clustering, or bi-clustering, means partitioning into a set of $k$ (sometimes unknown) blocks each containing a consistent local feature pattern (Note that it is not generally possible to display several bi-clusters at the same time as contiguous blocks).

## 4 The proposed heuristic based modularity

This section introduces a heuristic based approach for modularity with three different optimization models. A set of protein-neighborhood related functions is proposed to extend, accordingly, the unveil ability of a single objective EA. First, the main components that characterize the evolution process of EA (solution representation and perturbation operators) are formulated towards solving the problem. Then, the optimization models and the heuristic operator are introduced and formulated to improve the quality of generated complexes in the search space. Finally, the main steps of the proposed EA is outlined.

### 4.1 The proposed EA

Any Evolutionary Algorithm (EA) searches for appropriate solutions from the set Ω of all possible solutions of the problem at hand. Generally, the search for good solutions is performed through individual evaluations, selection, crossover, and mutation operators. The design of such operators would then determine the characteristic of the adopted EA. In this section, the definitions of all components are relaxed for the purpose of complex detection problem in PPI networks.

First, the construction of several, but unknown, number of complexes among the interacted proteins of a given PPI network, is an important issue that the individual representation (i.e. chromosome genotype encoding) should take quite seriously. In the proposed EA, the locus-based representation used in [7] is adopted. A chromosome $P$ of the population $\mathbb{P}$ is defined as a collection of node-node neighbor genes. A single gene in the chromosome $P$ is defined by its locus and its allele. Consider a PPI network $\mathcal{N}$ with $n$ proteins. Then, $P$ will consist of $n$ genes, where locus $i$ identifies protein $i$ in the network, while its allele value $j$ corresponds to the neighbor $j$ that has an actual interaction with node $i$ in the network, i.e. $(v_i, v_j) \in E$. This in turn implies that both proteins $i$ and $j$ will be in the same complex. The decoding function $\delta$ of a chromosome $P$ (chromosome phenotype) outlines one of the possible partitioning of the network $\mathcal{N}$ into complexes, i.e. $\delta(P)$: $\mathcal{C} = \{C\}_{i=1}^{K}$. However, $K$ could vary from one chromosome to another.

Once the population is created and their individuals are evaluated (according to the modularity in Eq. 1), a set of good population of parents is selected and processed by perturbation operators to create better child individuals. Two main perturbation operators are used. These are crossover $\Psi_{\times}$ and mutation $\Psi_m$.

Uniform crossover is used and achieved with a specified *chromosome*-wise crossover probability, $p_{\times}$. Consider two chromosomes $P_1$: $(I_{1,1}, I_{1,2}, \dots, I_{1,n})$ and $P_2$: $(I_{2,1}, I_{2,2}, \dots, I_{2,n})$ to be the two participating parents in the crossover.

With probability $p_{\times}$, a child $P'$: $(I'_1, I'_2, \dots, I'_n)$ can be generated from the two parents by uniformly mixing their alleles (i.e. performing *protein*-wise *fair* combination). This can be formally defined by:

$\Psi_{\times}$: $(P_1, P_2, p_{\times}) \rightarrow P'$

$\forall i, 1 \leq i \leq n$:

$$I'_i = \begin{cases} I_{1,i} & if\ r \leq 0.5 \\ I_{2,i} & otherwise \end{cases}$$

(2)

where $r \sim [0,1]$ is a uniform random number.

The mutation operator $\Psi_m$ imitates the traditional *allelic* mutation operator which works on allele values and alters, with a specified mutation probability $p_m$, the allele (i.e. neighbor) of a selected locus (i.e. node). This can formally be specified by:. $\Psi_m$: $(P, p_m) \rightarrow P'$

$\forall i, 1 \leq i \leq n \wedge r \leq p_m$:

$I'_i = j | (i,j) \in E$

(3)

where $r \sim [0,1]$ is a uniform random number.

### 4.2 Formulation of the heuristic optimization models and operator

Generally, heuristic operator or search heuristic is defined to be a rule that decides which solution, given the current solution, to generate or to visit next based on some heuristic criterion. In evolutionary computation community, the search for designing appropriate heuristic for a given problem is essential and can harness the performance of the algorithm. In the following discussion, we introduce a heuristic operator with three optimization models tailored, here, specifically for complex detection problem.

The general characteristic of complexes in PPI networks expresses dense interactions within complexes while more sparse interactions among different complexes. The main purpose of the proposed heuristic operator $\Psi_h: P \rightarrow P'$ is to move proteins between the complexes of an individual solution $P$. The movement of the selected proteins should reduce the problems of both sparse intra-connections and dense inter-ties. Thus, the proposed $\Psi_h$ operator works, with a specified probability $p_h$, on those nodes maintaining un-reliable template in their chosen complexes and move them to other complexes that can participate within their proteins more reliably. Here, we propose three different optimization models on how to define reliability assignment of a node to a given complex.

Consider an individual chromosome $P: (I_1, I_2, \dots, I_n)$ corresponding to a candidate partitioning solution $\mathcal{C} = \{C_1, \dots, C_K\}$ with $K$ complexes. Let node $i$ that corresponds to gene $I_i$ being located in complex $k$, where $1 \leq k \leq K$. Then, node $i$ inside complex $k$ has a possible reliable interaction assignment that can be expressed as the difference between the impact of the intra-connections and inter-connections:

$$ReliableAssign(i, C_k) = IntraImpact_{i \in C_k} - InerImpact_{i \in C_k} \quad (4)$$

For $IntraImpact$ and $InterImpact$ in Eq. 4, three models are proposed to define them. Let us first consider an extended version of the adjacency matrix $A$ (discussed in Section 2). A weighted adjacency matrix $wA$ is constructed using Eq. 5.

$$wA(i,j) = \begin{cases} \sum_{k=1}^{n} \left(\frac{A(i,k)}{|l_i|-1}\right) \times \left(\frac{A(k,j)}{|l_k|}\right) & if\ A(i,j) = 1 \\ 0\ otherwise \end{cases} \quad (5)$$

The proposed heuristic operator, $\Psi_h$, (see Algorithm 1), then, moves node $i$ to another complex $k'$, $1 \leq k' \leq K$ and $k' \neq k$ where node $i$ could maintain, there, the highest reliability assignment, i.e. with the highest difference between intra-connections and inter-connections impact. The proposed heuristic operator ($\Psi_h$), then, can be stated formally as in Eq. 5. Note that when more than one complex can receive node $i$ with equal $ReliableAssign$ value, then $\Psi_h$ randomly selects any one of these complexes.

$$\Psi_h(i \in C_k, p_h) = \max_{C_{k'} \in \mathcal{C}} ReliableAssign(i, C_{k'}) \quad (6)$$

Now, three different heuristic models are proposed to reflect $IntraImpact_{i \in C_k}$ and $InterImpact_{i \in C_k}$ terms of $ReliableAssign$. The first model ($heuristic\#1$) considers the difference between the accumulated impact of the weighted intra-connections of all proteins $i \in C_k$ and their inter-connections, i.e.:

$$IntraImpact_{i \in C_k} = \sum_{i \in C_k, j \in C_k} wA(i,j) \quad (7)$$

$$InterImpact_{i \in C_k} = \sum_{i \in C_k, j \notin C_k} wA(i,j) \quad (8)$$

Equation 7 and 8 are also maintained in the next two heuristic models, however, to be combined with additional terms in an alternating fashion with equal probability. For the second heuristic model ($heuristic\#2$), $IntraImpact_{i \in C_k}$ and $InterImpact_{i \in C_k}$, are computed with equal probability, to be either: 1) Eq.7 and Eq.8, respectively, or 2) to return accumulated impact of the intra-connections of neighbors of all proteins $i \in C_k$ and their inter-connections, i.e.:

$$IntraImpact_{i \in C_k} = \sum_{i \in C_k, j \in C_k \wedge (i,j) \in E} A(i,j) \quad (9)$$

$$InterImpact_{i \in C_k} = \sum_{i \in C_k, j \notin C_k \wedge (i,j) \in E} A(i,j) \quad (10)$$

Finally, the third model ($heuristic\#3$) computes $IntraImpact_{i \in C_k}$ and $InterImpact_{i \in C_k}$, with equal probability, to be either: 1) equal to Eq.7 and Eq.8, respectively, or 2) to reflect the accumulated impact of the intra-connections and inter-connections of all proteins $i \in C_k$, i.e.:

$$IntraImpact_{i \in C_k} = \sum_{i \in C_k, j \in C_k} A(i,j) \quad (11)$$

$$InterImpact_{i \in C_k} = \sum_{i \in C_k, j \notin C_k} A(i,j) \quad (12)$$

**Algorithm 1: $\Psi_h$**

**Input:** 1) chromosome phenotype $\mathcal{C} = \{C_1, \dots, C_K\}$ ,
2) number of proteins $n$,
3) probability of heuristic operator $p_h$

**Output:** modified chromosome phenotype $\mathcal{C}' = \{C_1, \dots, C_K\}$

**for** $i = 1$ to $n$ **do**
  **if**( $rand \leq p_h$) // apply heuristic movement to protein $i$
    **set** $C_i \leftarrow Complex(i)$; // return the current complex of protein $i$
    **compute** $cur\_intra\_impact\_i$; // according to the heuristic model
    **compute** $cur\_inter\_impact\_i$; // according to the heuristic model
    **set** $[C_j,\ ReliableAssign] \leftarrow argmax_{C_j \in \mathcal{C}} \left(ReliableAssign(i, C_j)\right)$; //return complex $C_j$ with maximum reliability assignment for protein $i$
    **if**($ReliableAssign > (cur\_intra\_impact\_i - cur\_inter\_impact\_i)$)
      **set** $Complex(i) \leftarrow C_j$;
    **end if**
  **end if**
**end for**

### 4.3 General EA layout

The overall component of the proposed EA with the proposed heuristic model is then presented in Algorithm 2.

**Algorithm 2: heuristic EA for complex detection problem in PPI networks**

**Input:** 1) PPI network: $\mathcal{N}(n, E)$,
2) population size: $\mu$, and maximum number of generations $max_t$
3) EA operators and their probabilities: $s$, $\Psi_\times$, $\Psi_m$, $\Psi_h$ $p_\times$, $p_m$, $p_h$,

**Output:** Best individual solution $P$

initialize $\mathbb{P} \leftarrow \{\mathrm{P}_1, \mathrm{P}_2, \dots, \mathrm{P}_\mu\}$;
$t \leftarrow 0$;
**evaluate** modularity for each individual in the population $\mathbb{P}(t) = \{Q_1, Q_2, \dots, Q_\mu\}$;
**while** $(t \leq max_t)$ **do**
  **for** $i \leftarrow 1$ to $\mu$ **do**
    $P_{i,1}(t) \leftarrow select(\mathbb{P}_i(t))$; // select parent 1 for $P_i$
    $P_{i,2}(t) \leftarrow select(\mathbb{P}_i(t))$; // select parent 2 for $P_i$
    $P_i(t) \leftarrow \Psi_\times(P_{i,1}(t), P_{i,2}(t), p_\times)$;
    $P_i(t) \leftarrow \Psi_m(P_i(t), p_m)$;
    $P_i(t) \leftarrow \Psi_h(P_i(t), p_h)$;
    evaluate $Q(P_i(t))$;
  **end for**
  $t \leftarrow t + 1$;
**end while**
**return** $P \in \mathbb{P}(t)$ with maximum $Q$;

**5 Results**

The experiments include two commonly PPI networks, denoted by $\mathcal{N}1$ and $\mathcal{N}2$. $\mathcal{N}1$ has $n = 990$ proteins with $E = 4687$ interactions, while $\mathcal{N}2$ has $n = 1443$ proteins and $E = 6993$ interactions. To validate the quality of the predicted complexes generated by the tested EA without heuristic against with heuristic, two sets of golden standard complexes ($Cmplx\_D1$ and $Cmplx\_D2$) drawn from the Munich Information Center for Protein Sequence (MIPS) catalog [8] are used in the experiments. $Cmplx\_D1$ contains 81 complexes, while $Cmplx\_D2$ is made of 162 hand-curated complexes To evaluate the quality of the detected complexes obtained by the EA, several metrics are used. The predicted set of complexes $\mathcal{C} = \{C_1, C_2, \dots, C_K\}$ obtained by EA is compared with the golden standard complexes $\mathcal{C}^* = \{C_1^*, C_2^*, \dots, C_{K^*}^*\}$ of $K^*$ complexes. A predicted complex $C_i$ in the solution $\mathcal{C}$ overlaps a golden standard complex $C_j^*$ by an overlapping score ($OS$). Then, the predicted complex $C_i$ *matches* the golden standard complex $C_j^*$ if $OS$ is equal or larger than a specified threshold, $\sigma_{OS}$, [9].

$$OS(C_i, C_j^*) = \frac{|C_i \cap C_j^*|^2}{|C_i||C_j^*|} \tag{13}$$

where $|C_i \cap C_j^*|$ is the number of proteins common to both a predicted complex and a golden standard complex.

$$match(C_i, C_j^*) = \begin{cases} 1 \; if \; OS(C_i, C_j^*) \geq \sigma_{OS} \\ 0 \; otherwise \end{cases} \tag{14}$$

At both complex and protein levels, the three standard metrics of recall, precision, and F measure are evaluated. The complex/protein levels, $recall/recall_N$, $precision/precision_N$, and cumulative $F\text{-}measure/F_N$ are defined. In $recall$, the fraction of golden complexes/proteins that are matched to any predicted complex is determined. On the other hand, $precision$ refers to the fraction of predicted complexes/proteins that are matched to any golden standard complex. A harmonic mean of both $recall$,and $precision$ is reflected by $F\text{-}measure$.

$$recall = \frac{|C_i^* : C_i^* \in \mathcal{C}^* \wedge \quad \exists C_j \in \mathcal{C} \rightarrow match(C_i^*, C_j)|}{K^*} \tag{15}$$

$$precision = \frac{|C_i : C_i \in \mathcal{C} \wedge \exists C_j^* \in \mathcal{C}^* \rightarrow match(C_i, C_j^*)|}{K_{\mathcal{C}}} \tag{16}$$

$$F\text{-}measure = \frac{2 * recall * precision}{recall + precision} \tag{17}$$

$$recall_N = \frac{\sum_{i=1}^{K^*} |m_i|}{\sum_{i=1}^{K^*} |C_i^*|} \tag{18}$$

where $|m_i| = max_{C_j \in \mathcal{C}} |match(C_i^*, C_j)$.

$$precision_N = \frac{\sum_{i=1}^{K_{\mathcal{C}}} |m_i|}{\sum_{i=1}^{K_{\mathcal{C}}} |C_i|} \tag{19}$$

where $|m_i| = max_{C_j^* \in \mathcal{C}^*} |match(C_i, C_j^*)$.

$$F_N = \frac{2 * recall_N * precision_N}{recall_N + precision_N} \tag{20}$$

Other measures can be computed with no dependency to the overlapping score ($\sigma_{OS}$). These are general sensitivity ($sensitivity$), general positive predictive value ($PPV$), and $accuracy$ [9]. General sensitivity ($sensitivity$) between the set of complexes $\mathcal{C}^* = \{C_1^*, C_2^*, \dots, C_{K^*}^*\}$ and the set of detected partitioning solution $\mathcal{C} = \{C_1, C_2, \dots, C_K\}$ is the weighted average of complex-wise sensitivity of all reference complexes (Eq. 21). Similarly, general $PPV$, with respect to the detected complexes (Eq. 22).

$$sensitivity = \frac{\sum_{i=1}^{K^*} max_{j=1}^{K} \frac{T(i,j)}{n_i^*}}{\sum_{i=1}^{K^*} n_i^*} \tag{21}$$

$$PPV = \frac{\sum_{j=1}^{K} max_{i=1}^{K^*} \frac{T(i,j)}{T_{\cdot j}}}{\sum_{j=1}^{K} \sum_{i=1}^{K^*} T(i,j)} \tag{22}$$

where $T_{.j}$ represents the marginal sum of column $j$. The tradeoff between $sensitivity$ and $PPV$ can be represented by the geometric $accuracy$. High accuracy (Eq. 23) value requires a high performance for both $sensitivity$ and $PPV$.

$$accuracy = \sqrt{sensitivity * PPV} \quad (23)$$

Results for all mentioned metrics are reported in Tables-(1 – 7) and Figures-(2, 3). The reported results (given in bold) clearly reveal the positive impact of the proposed heuristic operator with the three different versions of models. The proposed heuristic operator extends the applicability of the well-known modularity function ($Q$) to partition a given PPI network.

**Table 1**-Performance in terms of $\boldsymbol{recall}$ for $\mathcal{N}1$ and $\mathcal{N}2$ when $Q$ is adopted without heuristic, in one hand, and with the proposed heuristics, on the other hand. Threshold of overlapping score $\sigma_{OS}$ is varied from 0.1 to 0.5 in step of 0.05.

| $\sigma_{OS}$ | $\mathcal{N}1$ | | | | $\mathcal{N}2$ | | | |
|---|---|---|---|---|---|---|---|---|
| | **Q without *heuristic*** | **Q with** | | | **Q without *heuristic*** | **Q with** | | |
| | | **heuristic #1** | **heuristic #2** | **heuristic #3** | | **heuristic #1** | **heuristic #** | **heuristic #3** |
| **0.1** | 0.8500 | **0.8949** | **0.8538** | **0.8744** | 0.9287 | 0.9267 | 0.9167 | 0.9193 |
| **0.15** | 0.7910 | **0.8359** | 0.7795 | **0.8141** | 0.8493 | **0.8493** | 0.8307 | 0.8420 |
| **0.2** | 0.7244 | **0.7872** | **0.7256** | **0.7538** | 0.7860 | **0.7860** | 0.7613 | 0.7653 |
| **0.25** | 0.6872 | **0.7449** | **0.6782** | **0.7077** | 0.7260 | **0.7307** | 0.7013 | 0.7147 |
| **0.3** | 0.6462 | **0.7090** | **0.6551** | **0.6731** | 0.6580 | 0.6573 | 0.6307 | 0.6467 |
| **0.35** | 0.6269 | **0.7038** | **0.6359** | **0.6603** | 0.5767 | **0.5813** | 0.5660 | **0.5787** |
| **0.4** | 0.5936 | **0.6872** | **0.6064** | **0.6244** | 0.5333 | **0.5367** | **0.5340** | **0.5413** |
| **0.45** | 0.5577 | **0.6538** | **0.5808** | **0.6000** | 0.4633 | **0.4673** | **0.4660** | **0.4773** |
| **0.5** | 0.5423 | **0.6346** | **0.5603** | **0.5872** | 0.4440 | **0.4467** | **0.4467** | **0.4520** |

**Table 2**-Performance in terms of $\boldsymbol{recall_N}$ for $\mathcal{N}1$ and $\mathcal{N}2$ when $Q$ is adopted without heuristic, in one hand, and with the proposed heuristics, on the other hand. Threshold of overlapping score $\sigma_{OS}$ is varied from 0.1 to 0.5 in step of 0.05.

| $\sigma_{OS}$ | $\mathcal{N}1$ | | | | $\mathcal{N}2$ | | | |
|---|---|---|---|---|---|---|---|---|
| | **Q without *heuristic*** | **Q with** | | | **Q without *heuristic*** | **Q with** | | |
| | | **heuristic #1** | **heuristic #2** | **heuristic #3** | | **heuristic #** | **heuristic #** | **heuristic #3** |
| **0.1** | 0.8582 | **0.8936** | **0.8835** | **0.8970** | 0.6062 | **0.6070** | **0.6121** | **0.6116** |
| **0.15** | 0.8180 | **0.8498** | **0.8300** | **0.8505** | 0.5833 | 0.5790 | 0.5800 | **0.5856** |
| **0.2** | 0.7575 | **0.8137** | **0.7866** | **0.8016** | 0.5571 | 0.5507 | 0.5459 | 0.5518 |
| **0.25** | 0.6983 | **0.7635** | **0.7100** | **0.7466** | 0.5220 | 0.5155 | 0.5101 | 0.5126 |
| **0.3** | 0.6163 | **0.7091** | **0.6779** | **0.6765** | 0.4717 | 0.4707 | 0.4542 | 0.4650 |
| **0.35** | 0.6011 | **0.7049** | **0.6535** | **0.6668** | 0.4026 | **0.4113** | **0.4138** | **0.4301** |
| **0.4** | 0.5478 | **0.6860** | **0.6029** | **0.6157** | 0.3576 | 0.3740 | **0.3783** | **0.3916** |
| **0.45** | 0.4964 | **0.6288** | **0.5524** | **0.5601** | 0.3040 | **0.3211** | **0.3215** | **0.3403** |
| **0.5** | 0.4718 | **0.5845** | **0.5180** | **0.5382** | 0.2783 | **0.2881** | **0.2904** | **0.3042** |

**Table 3** Performance in terms of ***precision*** for $\mathcal{N}1$ and $\mathcal{N}2$ when $Q$ is adopted without heuristic, in one hand, and with the proposed heuristics, on the other hand. Threshold of overlapping score $\sigma_{OS}$ is varied from 0.1 to 0.5 in step of 0.05.

| $\sigma_{OS}$ | $\mathcal{N}1$ | | | | $\mathcal{N}2$ | | | |
|---|---|---|---|---|---|---|---|---|
| | **Q without *heuristic*** | **Q with** | | | **Q without *heuristic*** | **Q with** | | |
| | | **heuristic #1** | **heuristic #2** | **heuristic #3** | | **heuristic #** | **heuristic #** | **heuristic #** |
| **0.1** | 0.7735 | 0.7588 | **0.7859** | **0.7799** | 0.5932 | **0.5964** | **0.6125** | **0.6049** |
| **0.15** | 0.7361 | 0.7212 | **0.7533** | **0.7431** | 0.5656 | **0.5698** | **0.5897** | **0.5872** |
| **0.2** | 0.7332 | 0.7175 | **0.7533** | **0.7402** | 0.5403 | **0.5458** | **0.5643** | **0.5634** |
| **0.25** | 0.7226 | 0.7097 | **0.7485** | **0.7343** | 0.4964 | 0.4945 | **0.5169** | **0.5235** |
| **0.3** | 0.7002 | 0.6931 | **0.7421** | **0.7224** | 0.4913 | 0.4814 | **0.5068** | **0.5199** |
| **0.35** | 0.6866 | **0.6880** | **0.7343** | **0.7163** | 0.4694 | 0.4548 | **0.4878** | **0.4924** |
| **0.4** | 0.6718 | **0.6844** | **0.7172** | **0.7059** | 0.4559 | 0.4331 | **0.4751** | **0.4764** |
| **0.45** | 0.6452 | **0.6653** | **0.7016** | **0.6869** | 0.4178 | 0.4039 | **0.4441** | **0.4428** |
| **0.5** | 0.6288 | **0.6477** | **0.6802** | **0.6735** | 0.4103 | 0.3956 | **0.4378** | **0.4339** |

**Table 4**-Performance in terms of $\boldsymbol{precision_N}$ for $\mathcal{N}1$ and $\mathcal{N}2$ when $Q$ is adopted without heuristic, in one hand, and with the proposed heuristics, on the other hand. Threshold of overlapping score $\sigma_{OS}$ is varied from 0.1 to 0.5 in step of 0.05.

| $\sigma_{OS}$ | $\mathcal{N}1$ | | | | $\mathcal{N}2$ | | | |
|---|---|---|---|---|---|---|---|---|
| | **Q without *heuristic*** | **Q with** | | | **Q without *heuristic*** | **Q with** | | |
| | | **heuristic #1** | **heuristic #2** | **heuristic #3** | | **heuristic #** | **heuristic #** | **heuristic #** |
| **0.1** | 0.6338 | **0.7185** | **0.6683** | **0.6779** | 0.6829 | **0.6976** | **0.6890** | **0.6974** |
| **0.15** | 0.6302 | **0.7138** | **0.6653** | **0.6743** | 0.6768 | **0.6929** | **0.6853** | **0.6944** |
| **0.2** | 0.6300 | **0.7107** | **0.6653** | **0.6715** | 0.6672 | **0.6841** | **0.6735** | **0.6829** |
| **0.25** | 0.6097 | **0.7037** | **0.6511** | **0.6653** | 0.6510 | **0.6607** | 0.6509 | **0.6687** |
| **0.3** | 0.5692 | **0.6779** | **0.6359** | **0.6325** | 0.6429 | 0.6328 | 0.6241 | **0.6465** |
| **0.35** | 0.5596 | **0.6743** | **0.6194** | **0.6260** | 0.6106 | 0.6082 | **0.6146** | **0.6246** |
| **0.4** | 0.5297 | **0.6719** | **0.5867** | **0.6067** | 0.5826 | 0.5710 | **0.5910** | **0.5894** |
| **0.45** | 0.4906 | **0.6282** | **0.5501** | **0.5588** | 0.5259 | **0.5312** | **0.5331** | **0.5510** |
| **0.5** | 0.4688 | **0.5845** | **0.5180** | **0.5382** | 0.5078 | **0.5100** | **0.5138** | **0.5294** |

**Table 5-**Performance in terms of $\boldsymbol{Fmeasure}$ for $\mathcal{N}1$ and $\mathcal{N}2$ when $Q$ is adopted without heuristic, in one hand, and with the proposed heuristics, on the other hand. Threshold of overlapping score $\sigma_{OS}$ is varied from 0.1 to 0.5 in step of 0.05.

| $\sigma_{OS}$ | $\mathcal{N}1$ | | | | $\mathcal{N}2$ | | | |
|---|---|---|---|---|---|---|---|---|
| | **Q without *heuristic*** | **Q with** | | | **Q without *heuristic*** | **Q with** | | |
| | | **heuristic #1** | **heuristic #2** | **heuristic #3** | | **heuristic #** | **heuristic #** | **heuristic #** |
| **0.1** | 0.8095 | **0.8209** | **0.8178** | **0.8241** | 0.7238 | **0.7255** | **0.7342** | **0.7296** |
| **0.15** | 0.7621 | **0.7737** | **0.7655** | **0.7765** | 0.6787 | **0.6818** | **0.6895** | **0.6918** |
| **0.2** | 0.7281 | **0.7498** | **0.7385** | **0.7467** | 0.6401 | **0.6440** | **0.6478** | **0.6489** |
| **0.25** | 0.7041 | **0.7260** | **0.7110** | **0.7204** | 0.5894 | **0.5895** | **0.5948** | **0.6042** |
| **0.3** | 0.6716 | **0.7002** | **0.6954** | **0.6965** | 0.5624 | 0.5554 | 0.5615 | **0.5762** |
| **0.35** | 0.6550 | **0.6950** | **0.6811** | **0.6866** | 0.5172 | 0.5098 | **0.5234** | **0.5319** |
| **0.4** | 0.6298 | **0.6849** | **0.6568** | **0.6622** | 0.4912 | 0.4789 | **0.5022** | **0.5066** |
| **0.45** | 0.5980 | **0.6587** | **0.6351** | **0.6401** | 0.4391 | 0.4329 | **0.4543** | **0.4592** |
| **0.5** | 0.5821 | **0.6400** | **0.6140** | **0.6269** | 0.4262 | 0.4191 | **0.4418** | **0.4426** |

**Table 6-**Performance in terms of $\boldsymbol{F_N}$ for $\mathcal{N}1$ and $\mathcal{N}2$ when $Q$ is adopted without heuristic, in one hand, and with the proposed heuristics, on the other hand. Threshold of overlapping score $\sigma_{OS}$ is varied from 0.1 to 0.5 in step of 0.05.

| $\sigma_{OS}$ | $\mathcal{N}1$ | | | | $\mathcal{N}2$ | | | |
|---|---|---|---|---|---|---|---|---|
| | **Q without *heuristic*** | **Q with** | | | **Q without *heuristic*** | **Q with** | | |
| | | **heuristic #1** | **heuristic #2** | **heuristic #3** | | **heuristic #** | **heuristic #** | **heuristic #** |
| **0.1** | 0.7286 | **0.7962** | **0.7609** | **0.7721** | 0.6422 | **0.6489** | **0.6480** | **0.6516** |
| **0.15** | 0.7115 | **0.7757** | **0.7383** | **0.7517** | 0.6264 | **0.6305** | **0.6281** | **0.6353** |
| **0.2** | 0.6873 | **0.7584** | **0.7205** | **0.7302** | 0.6071 | **0.6099** | 0.6027 | **0.6104** |
| **0.25** | 0.6506 | **0.7322** | **0.6792** | **0.7031** | 0.5791 | 0.5789 | 0.5716 | **0.5802** |
| **0.3** | 0.5917 | **0.6930** | **0.6562** | **0.6536** | 0.5438 | 0.5396 | 0.5255 | 0.5408 |
| **0.35** | 0.5796 | **0.6891** | **0.6359** | **0.6456** | 0.4847 | **0.4904** | **0.4940** | **0.5093** |
| **0.4** | 0.5385 | **0.6788** | **0.5946** | **0.6111** | 0.4426 | **0.4517** | **0.4607** | **0.4704** |
| **0.45** | 0.4934 | **0.6285** | **0.5512** | **0.5594** | 0.3848 | **0.3999** | **0.4005** | **0.4206** |
| **0.5** | 0.4702 | **0.5845** | **0.5180** | **0.5382** | 0.3591 | **0.3677** | **0.3705** | **0.3860** |

**Table 7** Performance in terms of ***Sensitivity***, ***PPV***, ***Accuracy***, ***CCF*** and ***Strength*** for $\mathcal{N}1$ and $\mathcal{N}2$ when Q is adopted without heuristic, in one hand, and with the proposed heuristics.

| | $\mathcal{N}1$ | | | | $\mathcal{N}2$ | | | |
|---|---|---|---|---|---|---|---|---|
| | **Q without *heuristic*** | **Q with** | | | **Q without *heuristic*** | **Q with** | | |
| | | **heuristic #1** | **heuristic #2** | **heuristic #3** | | **heuristic #1** | **heuristic #2** | **heuristic #3** |
| ***Sensitivity*** | 0.9621 | 0.9613 | **0.9772** | **0.9752** | 0.6420 | 0.6411 | **0.6531** | **0.6527** |
| ***PPV*** | 0.6345 | **0.7201** | **0.6688** | **0.6790** | 0.2785 | **0.2810** | 0.2778 | **0.2832** |
| ***Accuracy*** | 0.7810 | **0.8317** | **0.8081** | **0.8134** | 0.4229 | **0.4244** | **0.4259** | **0.4299** |
| ***CCF*** | 559.6000 | **589.3500** | **576.9000** | **579.8000** | 875.4500 | **876.9000** | **885.2500** | **890.1000** |
| ***Strength*** | 0.6647 | 0.6635 | **0.7435** | **0.7312** | 0.5647 | 0.5636 | **0.6098** | **0.6311** |

Two additional metrics are also included in Table 7. These are Cross Common Fraction ($CCF$) and Strength of complex structure. $CCF$ compares each pair of complexes, in which one comes from the golden data ($C_i$) and the second comes from the detected result ($C_j$), to find the maximal shared parts.$Strength$ measures the intensity of the detected complexes ($C_j$). This measure comes after Radicchi et al. [10] well-known community definition. They showed that a community structure usually seems correct, strong, and valid if most members with their neighbors are inside one community.

$$CCF = \frac{1}{2}\sum_{i=1}^{K^*} \max_j |C_i \cap C_j| + \frac{1}{2}\sum_{j=1}^{K} \max_i |C_i \cap C_j| \quad (24)$$

$$Strength = \frac{1}{K}\sum_{i=1}^{K} Score(C_i) \quad (25)$$

where:

$$Score(C_i) = \begin{cases} 1 & if\ \forall v \in C_i \rightarrow |l_{v,intra}|_{v\in C_i} + |l_{v,inter}|_{v\in C_i} \\ 0.5 & if\ \sum_{\forall v\in C_i} |l_{v,intra}|_{v\in C_i} > \sum_{\forall v\in C_i} |l_{v,inter}|_{v\in C_i} \\ 0 & otherwise \end{cases} \quad (26)$$

Generally, all reported results reveal consistent effectiveness of the proposed heuristic models and the proposed operator when coupled with the modularity function to partition a PPI network into different complexes. The detected complexes have satisfactory aggregation of intra connections where more intra-group connections appear than inter-group connections. This occurs much clearly in the first PPI network than in the second network. This is mainly due to the fact that the second network contains several overlapping proteins.

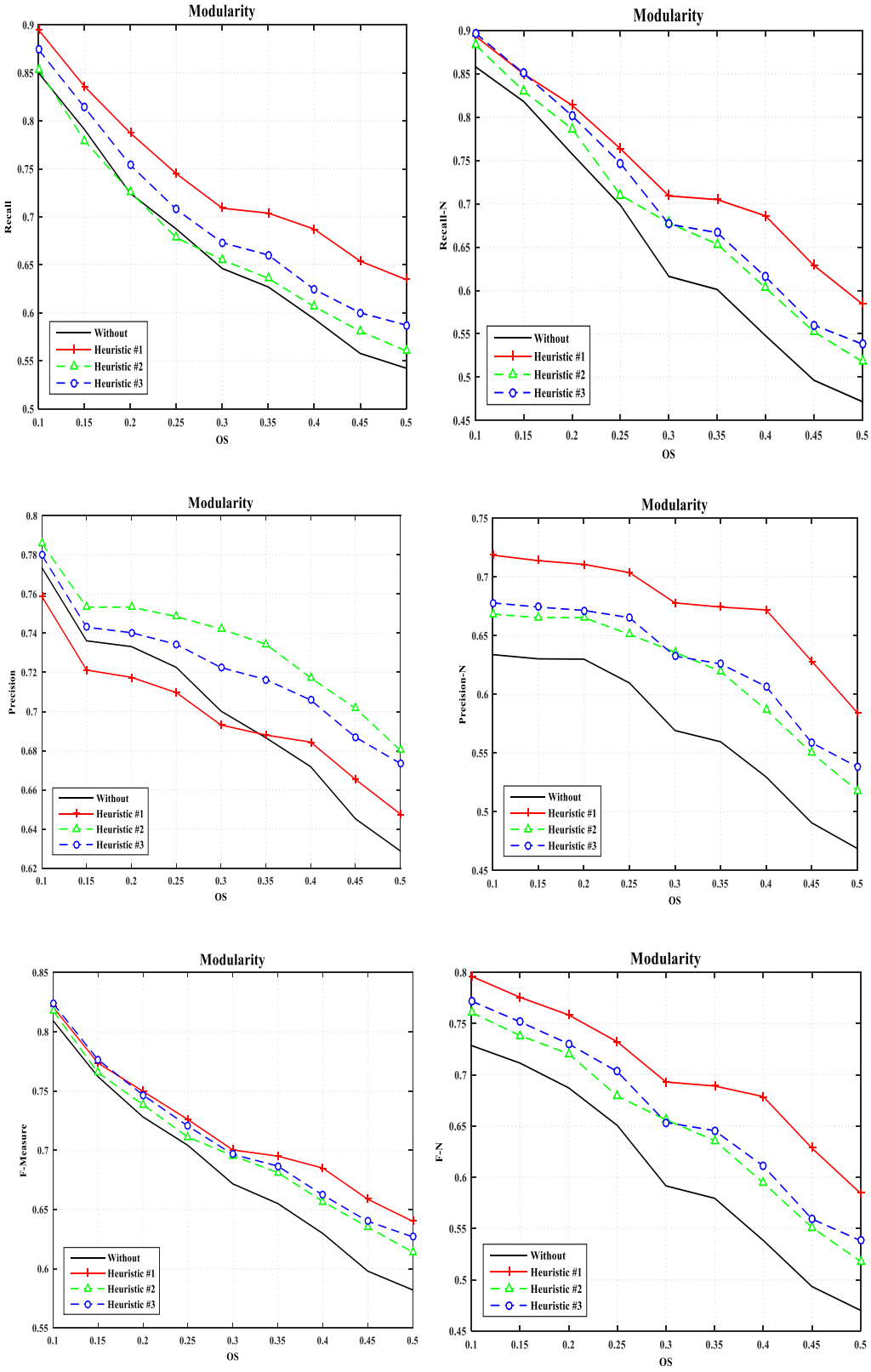


**Figure 2**-Performance at the complex and protein levels for $\mathcal{N}1$ when $Q$ is adopted without heuristic, in one hand, and with the proposed heuristics, on the other hand. Threshold of overlapping score $\sigma_{OS}$ is varied from 0.1 to 0.5 in step of 0.05.

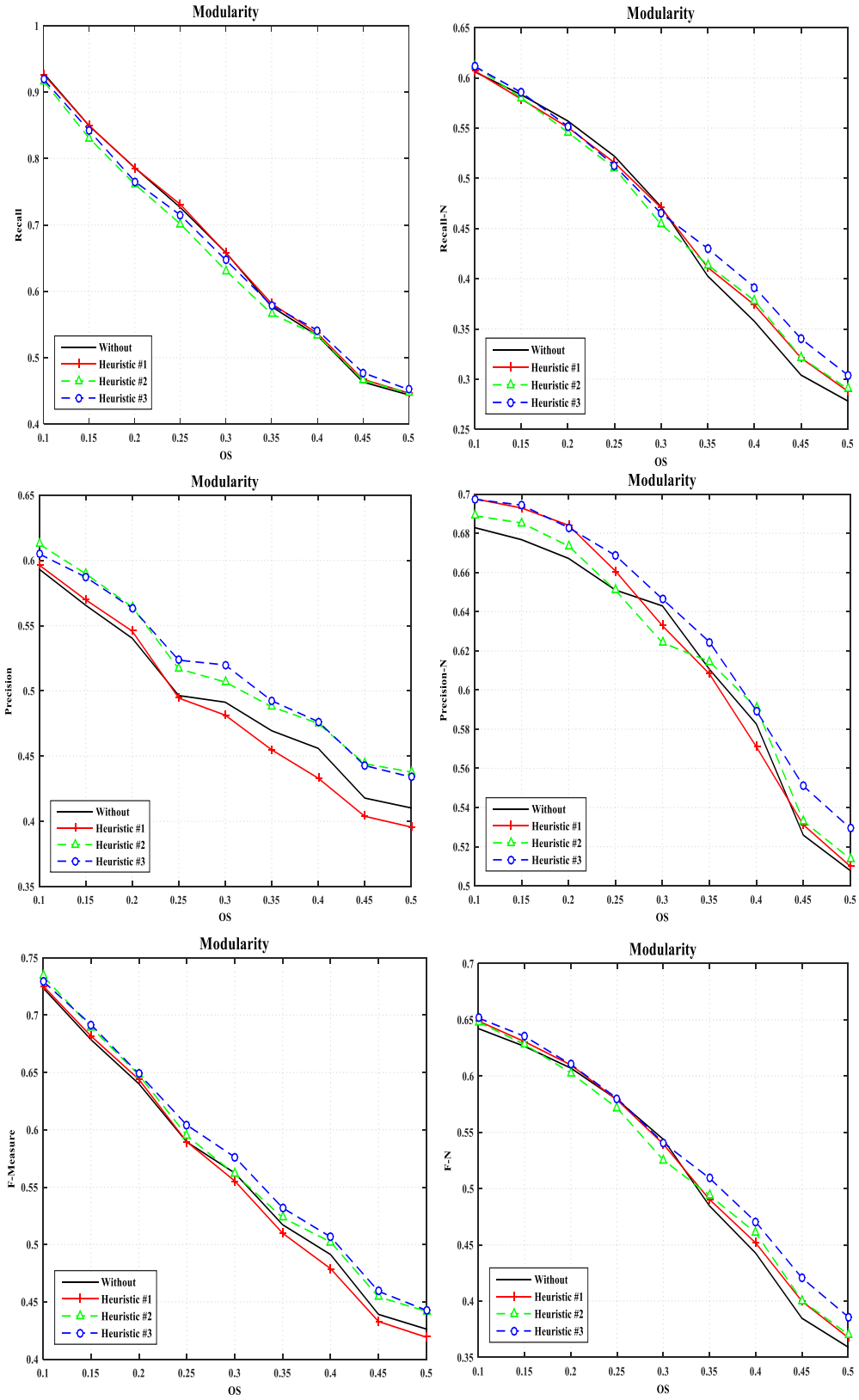


**Figure 3-**Performance at the complex and protein levels for $\mathcal{N}2$ when $Q$ is adopted without heuristic, in one hand, and with the proposed heuristics, on the other hand. Threshold of overlapping score $\sigma_{OS}$ is varied from 0.1 to 0.5 in step of 0.05.

## 6 Conclusions

The results reported in this paper show the importance on explicitly considering neighboring relations and heuristic movement models to improve the detection reliability of the modularity based EA. The proposed heuristic models emphasize the complex oriented structures where dense intra connections and sparse inter connections are declared. Generally, the results provided in this paper are encouraging, especially for the future of the proposed heuristic operators and models, and their extensions in the application to other complex detection models (e.g. community score, normalized cut, and ratio cut) and difficult mining problems in complex networks, for example for community detection problem in weighted networks and overlapping complex detection.

## References


1. Lancichinetti, A. and Fortunato, S. **2009.** Community detection algorithms: a comparative analysis. Physical review E, **80**(5): 056117.
2. Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z. and Wagner, D. **2008.** On modularity clustering. *IEEE transactions on knowledge and data engineering*, **20**(2): 172-188.
3. Clauset, A., Newman, M.E. and Moore, C. **2004.** Finding community structure in very large networks. *Physical review E*, **70**(6): 066111.
4. Coello, C.A.C., Van Veldhuizen, D.A. and Lamont, G.B**. 2002.** *Evolutionary algorithms for solving multi – objective problems* (Vol. 242). New York: Kluwer Academic.
5. Newman, M.E.J and Girvan, M. **2004.** "Finding and evaluating community structure in networks," *Physical Review E*, **69**(026113).
6. Girvan, M. and Newman, M.E. **2002.** Community structure in social and biological networks. Proceedings of the national academy of sciences, **99**(12): 7821-7826.
7. Attea, B.A., Hariz, W.A. and Abdulhalim, M.F. **2016.** Improving the performance of evolutionary multi-objective co-clustering models for community detection in complex social networks. *Swarm and Evolutionary Computation*, **26**: 137-156.
8. Gavin, A. C., Aloy, P., Grandi, P., Krause, R., Boesche, M., Marzioch, M. and Edelmann, A. 2006. Proteome survey reveals modularity of the yeast cell machinery. *Nature*, **440**(7084): 631-636.
9. Mewes, H. W., Frishman, D., Güldener, U., Mannhaupt, G., Mayer, K., Mokrejs, M. and Weil, B. **2002.** MIPS: a database for genomes and protein sequences. *Nucleic acids research*, **30**(1): 31-34.
10. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V. and Parisi, D. **2004.** Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, **101**(9): 2658-2663.