A Coclustering Approach for Mining Large Protein-Protein Interaction Networks

Clara Pizzuti and Simona E. Rombo

Abstract—Several approaches have been presented in the literature to cluster Protein-Protein Interaction (PPI) networks. They can be grouped in two main categories: those allowing a protein to participate in different clusters and those generating only nonoverlapping clusters. In both cases, a challenging task is to find a suitable compromise between the biological relevance of the results and a comprehensive coverage of the analyzed networks. Indeed, methods returning high accurate results are often able to cover only small parts of the input PPI network, especially when low-characterized networks are considered. We present a coclustering-based technique able to generate both overlapping and nonoverlapping clusters. The density of the clusters to search for can also be set by the user. We tested our method on the two networks of yeast and human, and compared it to other five well-known techniques on the same interaction data sets. The results showed that, for all the examples considered, our approach always reaches a good compromise between accuracy and network coverage. Furthermore, the behavior of our algorithm is not influenced by the structure of the input network, different from all the techniques considered in the comparison, which returned very good results on the yeast network, while on the human network their outcomes are rather poor.

Index Terms—Coclustering, biological networks, protein-protein interaction networks, protein complexes, hub proteins.

1 Introduction

Proteins are the building blocks of all the organisms and play a fundamental role in executing and regulating many biological processes. Recently, great attention has been addressed to the whole set of protein-protein interactions (PPI) of a given organism, known as *interactome* or *protein-protein interaction network*. Indeed, there is evidence that, to understand cell activity, proteins cannot be analyzed independently from the other proteins they interact with [44]. Advances in technology have allowed researchers to derive, through experimental and in-silico methods, the collection of all the interactions among the proteins of an organism. The availability of protein-protein interaction networks has thus stimulated the search for automated and accurate tools to analyze pairwise protein interactions, with the aim of understanding how proteins work together to perform their tasks, and also for predicting the function of unknown proteins [10]. Several studies have recognized that biological systems are structured as interacting and separable modules [21], [23], [38], [41], [45]. Modularity means that a group of physically or functionally related proteins join together to accomplish distinct functions [10]. Thus, proteins can be grouped in clusters such that the proteins in the same cluster share common biological features, such as participating in the same processes, having similar functions, belonging to the same cellular compart. The detection of such clusters provides important knowledge about the organization of biological systems and cellular processes, giving a valuable help in understanding how organisms behave.

Some proteins present the characteristic of being connected to a high number of other proteins, often participating in multiple biological processes and performing different functions. To detect such multifacets proteins, recent techniques search for overlapping clusters, where a protein is allowed to belong to several clusters (e.g., [6], [22], [26], [31]).

More in general, clustering techniques should be able to single out biologically relevant clusters without neglecting to explore any significative part of the input network. Thus, an important problem is that of finding a solution constituting a suitable compromise between high accuracy and comprehensive coverage of the analyzed networks.

1.1 A Brief Overview

In the last few years, there has been an increasing interest in studying clustering methods able to detect groups of proteins densely interconnected. PPI networks clustering approaches can be broadly categorized as distance-based and graph-based ones [27]. Distance-based clustering approaches apply traditional clustering techniques, such as hierarchical clustering, by employing the concept of distance between two proteins [7], [11], [32]. Graph-based clustering techniques consider the topology of the network. These techniques find the clusters by applying different strategies.

A first strategy searches for subgraphs having maximum density [4], [6], [9], [17], [22], [28], [31], [34], [35]. In such a case, a subgraph can be considered *dense* according to different notions of density. For example, Bader and Hogue [9] apply the concepts of k-core and core clustering coefficient to define the weight of a node. A k-core is a

- *C. Pizzuti is with the Institute for High Performance Computing and Networking (ICAR), National Research Council of Italy (CNR), Via P. Bucci 41C, 87036 Rende (CS), Italy. E-mail: pizzuti@icar.cnr.it.*
- *S.E. Rombo is with the Institute for High Performance Computing and Networking (ICAR), National Research Council of Italy (CNR), and with the Department of Electronics, Computer Science and Systems (DEIS), University of Calabria, Via P. Bucci 42C, 87036 Rende (CS), Italy. E-mail: simona.rombo@deis.unical.it.*

Manuscript received 20 Apr. 2011; revised 2 Dec. 2011; accepted 8 Dec. 2011; published online 22 Dec. 2011.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-2011-04-0100.
Digital Object Identifier no. 10.1109/TCBB.2011.158.

1545-5963/12/$31.00 © 2012 IEEE Published by the IEEE CS, CI, and EMB Societies & the ACM

subgraph in which each vertex has degree at least k. The highest k-core of a graph is the most densely connected subgraph. The core-clustering coefficient of a node is the density of the highest k-core of the vertices directly connected to it, including itself. The weight of a node is then defined as the product of the node core-clustering coefficient and the highest k-core of its neighborhood. Palla et al. [31] and Adamcsek et al. [4] use the concept of k-clique, i.e., a complete subgraph constituted by k nodes such that there is an edge between each pair of nodes. They consider two k-cliques adjacent if they have $k - 1$ common nodes. A k-clique-community is then defined as the union of all the k-cliques that can be reached through adjacent k-cliques. Altaf et al. [6] discover protein complexes in large interaction graphs by using the concepts of *density* and *neighborhood*. The authors introduce the definitions of *cluster property* and *node weight* that take into account the common neighbors of nodes belonging to the same cluster. Lubovac et al. [28] identify dense subgraphs by introducing two network measures that combine functional information with topological properties of the networks. These measures, weighted cluster coefficient and weighted nearest neighbors degree, compute the strengths of interactions between the proteins by using their semantic similarity based on the Gene Ontology (GO) terms of the proteins. Georgii et al. [22] define the density of a module as the average pairwise weight of the nodes belonging to the module, where the weight is a value below or equal to 1. Fixed a density threshold, the authors find all the modules whose density is above the threshold. Another approach partitions the graph by optimizing a cost function [25], [41]. The concept of flow simulation, though applied in different ways, is exploited in [15], [16], [19], [24], [33]. A statistical approach to protein clustering is taken instead in [20], [39]. A method that models protein relationships as a signal transduction model is described in [24]. Many other clustering algorithms have been proposed [13], [38], [42], [46]. A complete list of all the proposals is beyond the aim of this paper. Surveys describing and comparing a number of methods presented in the literature can be found in [5], [12], [27], [36], [37]. All the above methods are able to separate relevant dense clusters. However, different methods return diverse results. Barabasi et al. [10] observed that obtaining multiple results is not only a limitation of present clustering methods, but it is also due to the network's hierarchical modularity. Indeed, modules have not a precise size, thus a network can be divided in many small modules, or in larger, fewer clusters. At present, however, there are no objective mathematical criteria to decide that one outcome is better than another. As they pointed out, the identification of groups of proteins of various sizes that together accomplish specific cellular functions is a key issue in network biology.

1.2 Contributions

We propose a technique based on a coclustering approach [29] to search for, possibly overlapping, dense clusters in protein-protein interaction networks. We model a protein-protein interaction network by an undirected graph and represent it as the binary adjacency matrix A of this graph, where rows and columns correspond to proteins and a 1 entry at the position (i, j) means that the proteins i and j interact. By drawing inspiration by previous successfully coclustering approaches [34], [35], we present RANCoC, a coclustering algorithm based on the search of dense submatrices in A, that suitably shifts its rows and columns in order to optimize a special notion of *quality* of a submatrix. Indeed, high-quality submatrices should correspond to modules of the input interactome whose proteins share important biological features (e.g., they participate in the same processes, they have similar functions, they belong to the same cellular compart). The algorithm starts with an initial random solution constituted by a single protein and expands it by adding/removing connected proteins that best contribute to improve the *quality* function. Differently from the previous techniques [34], [35], a new heuristics is introduced to avoid entrapment in local optima. The basic process is repeated until all the proteins are assigned to any group.

The main contributions of the algorithm can be summarized as follows:

- RANCoC automatically derives the number of modules present in the interaction network. This number is determined by the local optimal value of the *quality* function.
- A peculiarity of the *quality* function is that it has a positive real-valued resolution parameter that controls the size of the groups obtained in output. The higher the value of the parameter, the smaller the size of the clusters found. This gives the user the opportunity to analyze the network at different hierarchical levels.
- RANCoC can work in two different modes: the nonoverlapping mode, where proteins are allowed to belong to only one cluster, and the overlapping mode, where clusters can overlap. Thus, besides partitioning and isolating groups of proteins corresponding to the most compact sets of interactions, our approach is also able to identify overlapping modules in which a protein is involved, each group being distinguished by different biological properties. Such characteristic allows multifacets proteins to be recognized and clustered with a number of distinct groups.

RANCoC has been evaluated on two well-known PPI networks: the *Saccharomyces cerevisiae* network and the *Homo sapiens* network. Though the first network has been deeply studied in many approaches, many interactions of the second have not yet been discovered and/or studied. A comparison of RANCoC with six well-known protein clustering methods, Molecular COmplex DEtection (MCODE) [9], Restricted Neighborhood Search Clustering (RNSC) [25], Markov CLuster (MCL) [19], CFINDER [31], Dense Module Enumeration (DME) [22], and IPCA [26], shows comparable results on the *Saccharomyces cerevisiae* network, still finding a good compromise between the quality of the discovered clusters and the percentage of network that has been covered by the clustering process. Regarding the *Homo sapiens* network, the other approaches performed rather poorly, mainly when the overlapping modules are requested. RANCoC, instead, behaves very well

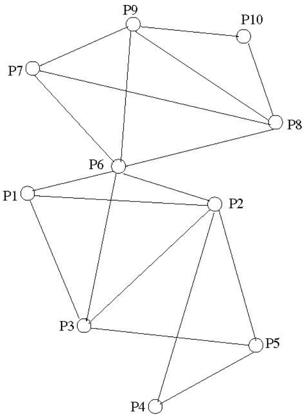


Fig. 1. An example protein-protein interaction graph.

for both the overlapping and nonovelapping case, thus outperforming all the other approaches. This points out that our method is robust in analyzing different PPI networks, also when they are not completely characterized and thus more sparse.

As a further validation campaign, RANCoC has been tested on the manually curated MIPS [30] known complexes for both yeast and human. A comparison with the other state-of-the art approaches shows the ability of the method in correctly classifying most of the considered benchmark complexes, with results better than those obtained by the comparison methods.

The software we developed is available at http://wwwinfo.deis.unical.it/~rombo/co-clustering/.

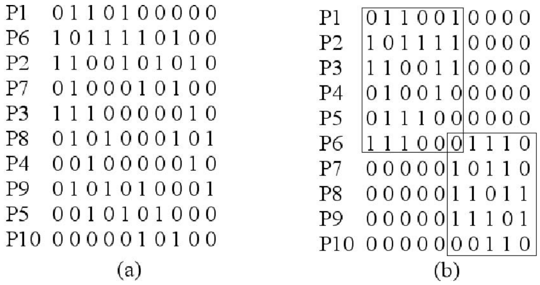
1.3 Organization of the Paper

The paper is organized as follows. In the next section, a description of our method is given. In Section 3, an extensive experimental study on the two mentioned networks is reported along with a comparison between our approach and the others. Section 4 finally draws some conclusions.

2 Methods

A protein-protein interaction network P can be modeled as an undirected graph $G = (V, E)$, where the nodes V correspond to the proteins and the edges E correspond to the pairwise interactions. If the network is constituted by N proteins, the associated graph can be represented with its $N \times N$ adjacency matrix A, where the entry at position (i, j) is 1 if there is an edge from node i to node j, 0 otherwise. Since the graph G is undirected, the adjacency matrix is a square symmetric matrix. The problem of finding dense regions of a network P can thus be transformed in that of rearranging the rows/columns of A to find dense subgraphs of the graph G associated with P and, consequently, dense square symmetric submatrices of the adjacency matrix A corresponding to G. We would like to find as many proteins as possible having the highest number of interactions. This corresponds to identify highly dense square submatrices, i.e., containing as many 1 values as possible. The higher the number of ones, the more likely those proteins are to be functionally related.

Searching for dense submatrices of a matrix A can be viewed as a special case of coclustering a binary data matrix, where the set of rows and columns represents the same concept. Coclustering [29], also known as biclustering, differently from clustering, tries to simultaneously group both the dimensions of a data set. To better understand the concept of coclustering, consider the protein interaction graph shown in Fig. 1 and the corresponding adjacency matrix (Fig. 2a), where we considered the rows in the order $P1$, $P6$, $P2$, $P7$, $P3$, $P8$, $P4$, $P9$, $P5$, $P10$. Coclustering this matrix means rearranging its rows/columns to obtain dense maximal submatrices, possibly sharing elements, i.e., overlapping. For the example, the reordering of its rows that at best accounts for the intuitive idea of dense maximal submatrices is that shown in Fig. 2b, where now the two dense submatrices constituted by the rows/columns 1-6 and 6-10, corresponding to the two subgraphs composed by the proteins $P1$-$P6$ and $P6$-$P10$, are clearly discernible. Note that the protein $P6$ belongs to both the clusters because it has a number of significant interactions with proteins of the two groups. Hence, a module S in a PPI network is a cocluster S, i.e., a submatrix, constituted by a subset I of the rows of A satisfying a quality measure. The more natural choice of quality function is to consider submatrices of maximum size having the maximum number of ones. In the next section, we introduce a quality function that fulfills both these requirements.



P1	0	1	1	0	1	0	0	0	0	0
P6	1	0	1	1	1	1	0	1	0	0
P2	1	1	0	0	1	0	1	0	1	0
P7	0	1	0	0	0	1	0	1	0	0
P3	1	1	1	0	0	0	0	0	1	0
P8	0	1	0	1	0	0	0	1	0	1
P4	0	0	1	0	0	0	0	0	1	0
P9	0	1	0	1	0	1	0	0	0	1
P5	0	0	1	0	1	0	1	0	0	0
P10	0	0	0	0	0	1	0	1	0	0

(a)

P1	0	1	1	0	0	1	0	0	0	0
P2	1	0	1	1	1	1	0	0	0	0
P3	1	1	0	0	1	1	0	0	0	0
P4	0	1	0	0	1	0	0	0	0	0
P5	0	1	1	1	0	0	0	0	0	0
P6	1	1	1	0	0	0	1	1	1	0
P7	0	0	0	0	0	1	0	1	1	0
P8	0	0	0	0	0	1	1	0	1	1
P9	0	0	0	0	0	1	1	1	0	1
P10	0	0	0	0	0	0	0	1	1	0

(b)

Fig. 2. Two different adjacency matrices corresponding to the graph in Fig. 1.

2.1 Optimization Function

Let A be the adjacency matrix modeling a network. A quality function that find dense and maximal submatrices, introduced in [34], can be defined as

$$Q(S) = M_r(S) \times v_S,$$

where S is a submatrix constituted by a subset $I = \{I_1, \dots, I_h\}$ of rows of A, $v_S = \sum_{i \in I, j \in I} a_{ij}$ is the number of 1 entries a_{ij} such that $i, j \in I$ and

$$M_r(S) = \frac{\sum_{i \in I} (a_{iI})^r}{|I|},$$

where $a_{iI} = \frac{1}{|I|} \sum_{j \in I} a_{ij}$ denotes the *mean value* of the ith row of the submatrix S.

The parameter r controls the size of the groups found. When $r = 1$, $M_r(S)$ coincides with the standard mean. The higher its value, the lower the size of the clusters found. In fact, since $a_{iI} < 1$, $M_{r+\epsilon}(S) \leq M_r(S) \leq M_{r-\epsilon}(S)$, for $\epsilon > 0$, thus the higher the value of r, the lower the value of $M_r(S)$, and, consequently, the lower the value of $Q(S)$. This implies that, given a submatrix S, if rows containing a low number

```
Algorithm: RANCoC
Input:  An adjacency matrix $A$; the maximum number of iterations $maxIter$;
        a probability value $p$; overlapping allowed (true/false);
Output: a clustering $S = \{S_1, \dots S_k\}$ of the PPI network corresponding to $A$
1.  repeat
2.    set $iter = 0$, $localMaximum = false$
3.    choose a row $j$ of $A$ at random and let $S_i = \{j\}$
4.    while $iter \leq maxIter$ and not $localMaximum$ do
5.      let $0 \leq \overline{p} \leq 1$ a random generated number
6.      if $\overline{p} < p$ then
7.        let $\overline{S_i}$ the co-cluster obtained from $S_i$ after adding/removing
          the row from $S_i$ that gives the best $Q(S_i)$ value
8.        if $Q(\overline{S_i}) > Q(S_i)$ then
9.          accept the move and update $Q(S_i)$
10.       else
11.         set $localMaximum = true$
12.     else
13.       remove a row from $S_i$ at random
14.     $iter = iter + 1$
15.   end while
16.   $S = S \cup S_i$
17.   if overlapping not allowed
18.     delete from $A$ the rows of $S_i$
19.   end if
20. until there are available rows
```

Fig. 3. The algorithm RANCoC.

of ones are added to S, then the quality function get trapped earlier in local maxima for increasing values of r. Thus, increasing r biases the quality function toward matrices containing a low number of zeroes but of lower volume, because the number of proteins that can be assigned to a cluster diminishes. The choice of r allows to take into account both the coverage of the network, and the goodness of the solution found in terms of sufficiently high number of interactions among the proteins belonging to the module.

In the next section, the *PPI network Coclustering*-based algorithm RANCoC is presented. The method uses the concept of *quality* to find maximally dense regions in the binary data adjacency matrix. Then, in Section 3, we will show how different values of r allow an analysis of the network at different hierarchical levels.

2.2 The Algorithm RANCoC

The algorithm RANCoC is an extension of the methods proposed in [34], [35] that allows a more efficient search of the solution space by changing the strategy that avoids to get trapped in local optima. The pseudocode of the algorithm is shown in Fig. 3. The method receives in input an adjacency matrix A, the maximum number of iterations $maxIter$, a probability value p, and the option to find overlapping clusters.

RANCoC is constituted by two main loops. The external loop is executed until all the proteins have been assigned to at least one cluster (steps 1-20), the internal loop (steps 4-15) starts with an initial random cocluster S constituted by a single protein (a row in step 3), and expands the cocluster with new proteins until either a preset of maximum number of iterations $maxIter$ is reached, or the solution cannot further be improved, i.e., the quality function $Q(S)$ does not increase any more because trapped into a local maximum (steps 8-11).

RANCoC is based on the concept of local search; thus, it evolves S by adding or removing rows from A (step 7) in order to maximize the quality function $Q(S)$. It is known that the main problem in applying local search methods is that the search space presents many local optima and, consequently, the algorithm could get trapped at local minima. The heuristics employed by RANCoC to overcome this problem consists in removing from S, with a fixed probability p, a row at random, even if the value of the quality function diminishes (step 13). This strategy is more efficient in terms of computation than that applied in the methods [34], [35], that eliminated the row scoring the minimum decrease of $Q(S)$, and it is more efficacious in avoiding entrapments in local optimal solutions since it allows the method to move from a solution to another possibly far one, and thus to better explore the space of candidate solutions. At the end of the ith internal loop, the obtained cocluster S_i is added to S (step 16) and its rows/columns are removed from A (steps 17-19), unless the user requires overlapping clusters. In such a case, the number of clusters a protein can belong to cannot exceed its degree k, that is, the number of other proteins it is connected with. In such a way, a protein can be reconsidered in the computation and assigned to multiple clusters, provided that its contribution to the *quality* function is effective, i.e., it is the choice that produces the best improvement. At this point, a new random cocluster is generated, and the process is repeated until all the rows/columns have been assigned.

3 Results

We validated our approach by testing it on two different PPI networks, the budding yeast *Saccharomyces cerevisiae* network and the *Homo sapiens* (also referred in the following as human) network. The two networks have been downloaded from the MINT database [14], that is one of the resources of the International Molecular interaction Exchange (IMEx) consortium of molecular interaction databases [1]. Since such databases provide reliability values associated with the protein-protein interactions, depending on the nature of the techniques exploited to obtain such interactions, low-reliable interactions have not been included in the input PPI networks we considered (we chose a cutoff value equal to 0.1 for the MINT confidence score). In particular, the yeast protein-protein interactions data include 5,443 proteins and 36,251 interactions, while the human network has 6,716 nodes and 16,322 interactions.

All the experimental evaluations have been performed by running RANCoC 50 times on each network, and then considering the mean values of the validation measures described below over the 50 executions. RANCoC needs two input parameters, p and $maxIter$. In particular, p represents the probability to remove a row, and $maxIter$ determines the maximum number of iterations allowed. We set the former to 0.1, and the latter to 1,000. It is worth to note that 1) a low value of probability p is preferable to avoid the disruption of the greedy steps; 2) the number of maximum iterations has never been reached, in fact on average not more than 50 iterations were executed before reaching a local optimum.

3.1 Validation Measures

To assess the quality of the results, we considered both the biological relevance of the returned clusters and the ability of the method to cover a significant portion of the analyzed networks. To measure cluster biological significance, we referred at first to the *Gene Ontology Consortium Online*

DataBase [8]. For each cluster, the GO annotations and the corresponding *p-values*, that evaluates the probability that a given cluster occurs by chance, have been computed by exploiting the software modules available at http://search.cpan.org/dist/GO-TermFinder/, according to [7]. Such a tool attempts to determine whether an observed level of annotation for a group of genes/proteins is significant within the context of annotation for all genes/proteins of the genome, also providing suitable correction factors for the obtained p-values. In our validation, we used all the three vocabularies provided by the Gene Ontology database, that are, *molecular function*, *cellular component*, and *biological process*.

In the following, we provide a short description of all the measures we considered.

p-value. The p-value is a commonly used measure of the functional homogeneity of a cluster. It gives the probability that a given set of proteins occurs by chance. In particular, given a cluster of size n with m proteins sharing a particular biological annotation, then the probability of observing m or more proteins that are annotated with the same GO term out of those n proteins, according to the Hypergeometric Distribution, is

$$p\text{-}value = \sum_{i=m}^{n} \frac{\binom{M}{i}\binom{N-M}{n-i}}{\binom{N}{n}},$$

where N is the number of proteins in the database with M of them known to have that same annotation. Thus, the closer the p-value to zero, the more significant the associated GO term. The biological significance of a group is settled by using a cutoff value to distinguish significant from insignificant groups. If the p-value of a cluster is above the cutoff that cluster is considered insignificant.

As observed by [43], it is interesting to have a global measure of an obtained clustering, instead of the p-value of a single group. The following measure is useful to this aim.

Clustering score. The clustering score of a clustering is defined as

$$c\text{-}score = 1 - \frac{\sum_i^{n_S} \min(p_i) + (n_I \times cutoff)}{n_I + n_S},$$

where $min(p_i)$ is the smallest p-value of the partition i, *cutoff* is the threshold imposed on the p-value to distinguish significant from insignificant groups, n_S is the number of significant partitions, and n_I is the number of insignificant partitions. In our evaluations, we adopted a *cutoff* equal to 0.05, which is that commonly employed in the literature.

The meaning of clustering score is that of evaluating the clustering obtained by an algorithm, by computing the probability that the output clusters of proteins could occur by chance. The clustering score alone, however, could be misleading since it does not take into account the percentage of proteins involved in a clustering. Thus, it could happen that a method has a high clustering score but only a small portion of all the proteins contained in the PPI network have been grouped. To measure how much a method is able to cover a considerable portion of the network under analysis, during the clustering process, we introduce the *coverage percentage*. Given a network of n nodes, let n' be the number of proteins that a clustering method did not assign to any of the returned clusters. Then, the coverage percentage is given by[1]

$$cp = \frac{n - n'}{n}.$$

High coverage is important since, as pointed out by Sharan et al. [40], an approach to functional annotation of proteins is based on assigning the function that is prevalent in a group of proteins, obtained by dividing the PPI network in dense, possibly overlapping, clusters. A measure that takes into account both the biological meaning of the clusters obtained and the coverage percentage can be defined as follows.

Normalized Clustering Score. The normalized clustering score for a given clustering returned by a method applied on a PPI network is defined as

$$nc\text{-}score = c\text{-}score \times cp.$$

Since for both yeast and human the MIPS databases [30] provide known protein complexes, it is possible to evaluate the effectiveness of a method in detecting such known complexes by comparing the predicted clusters with the true known complexes. To this end, we employ the same validation measures used in [6], [9], [26]. Such measures are described below.

Overlapping Score. The overlapping score between a predicted cluster P_c and a known complex K_c is defined as

$$OS(P_c, K_c) = \frac{|V_{P_c} \cap V_{K_c}|^2}{|V_{P_c}| \cdot |V_{K_c}|},$$

where $|V_{P_c} \cap V_{K_c}|$ is the size of the intersection set of the predicted cluster and the known complex, $|V_{P_c}|$ is the size of the predicted cluster and $|V_{K_c}|$ is the size of the known complex.

A known complex and a predicted cluster are considered a match if their overlapping score is equal to or larger than a specific threshold σ_{OS}.

Other two important measures to estimate the performance of algorithms for detecting protein complexes are sensitivity and specificity.

Sensitivity. Sensitivity is the fraction of the true-positive predictions out of all the true predictions, defined as

$$S_n = \frac{TP}{TP + FN},$$

where TP (true positive) is the number of the predicted clusters matched by the known complexes with $OS(P_c, K_c) \geq \sigma_{OS}$, and FN (false negative) is the number of the known complexes that are not matched by the predicted clusters.

Specificity. Specificity is the fraction of the true-positive predictions out of all the positive predictions, defined by the following formula:

$$S_p = \frac{TP}{TP + FP},$$

where false positive (FP) equals the total number of the predicted clusters minus TP.

[1] As will be further explained below, we do not consider the returned singletons as clusters in our analysis.

TABLE 1
Structural Properties for Different Values of r

r	Species	Number	Maximum Size	Coverage Percentage
0.5	*yeast*	358 (13.157)	781 (5.912)	0.957 (0.2001)
	human	531 (6.985)	780 (4.148)	0.780 (0.004)
1.0	*yeast*	673 (8.293)	68 (11.580)	0.935 (0.001)
	human	945 (8.959)	94 (15.447)	0.886 (0.006)
2.0	*yeast*	974 (8.318)	53 (0.700)	0.915 (0.002)
	human	1,171 (11.843)	45 (2.705)	0.709 (0.268)
3.0	*yeast*	1,235 (8.753)	51 (1.033)	0.908 (0.001)
	human	1,480 (11.232)	39 (3.561)	0.541 (0.261)
4.0	*yeast*	1,237 (9.350)	50 (0.707)	0.899 (0.214)
	human	1,493 (6.387)	33 (0.937)	0.512 (0.248)

TABLE 2
Normalized Clustering Score for Different Values of r

r	Species	Process	Component	Function
0.5	*yeast*	0.446	0.502	0.670
	human	0.273	0.391	0.509
1.0	*yeast*	**0.532**	**0.586**	**0.733**
	human	**0.545**	**0.557**	**0.759**
2.0	*yeast*	0.458	0.473	0.627
	human	0.259	0.257	0.408
3.0	*yeast*	0.377	0.440	0.598
	human	0.144	0.188	0.279
4.0	*yeast*	0.368	0.425	0.589
	human	0.139	0.185	0.294

According to [26], a predicted cluster and a known complex are considered to be a *match* if $OS(P_c, K_c) \geq \sigma_{OS}$.

3.2 Analysis of the Parameter r

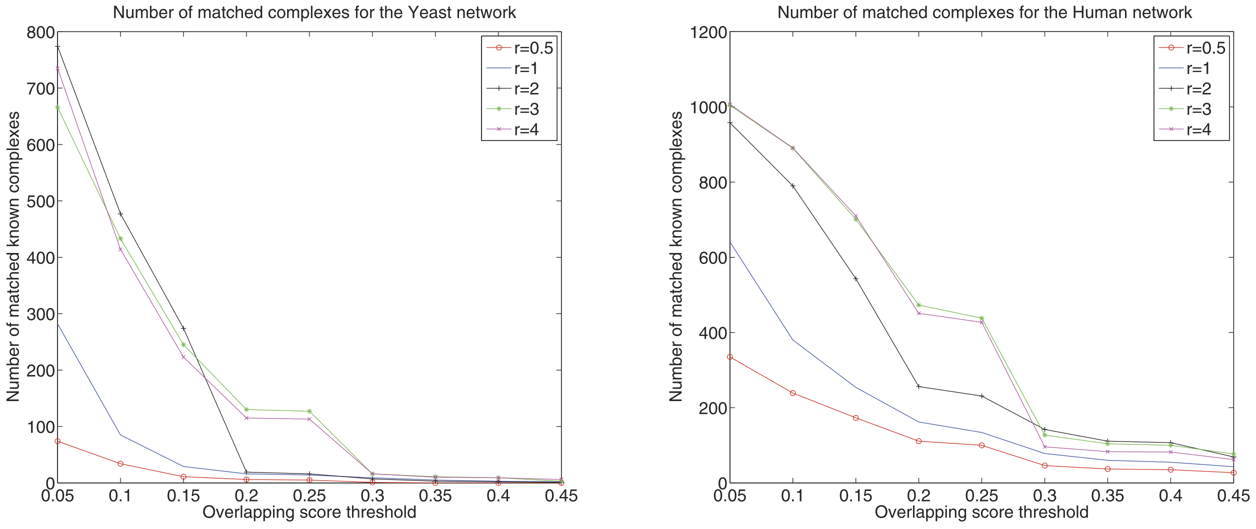
As a first series of experiments, we studied how the algorithm behaves in terms of both biological meaning and data coverage for different values of the parameter r (see Section 2). In particular, we run RANCoC in the nonoverlapping mode on the yeast network at first with five different values of r: 0.5, 1.0, 2.0, 3.0, and 4.0, respectively. Note that, for values of r lower than 0.5, we obtained a clustering made of a very large cluster (above 1,000 elements) and almost singletons. Such results are not much meaningful for our analysis, and we thus suggest to exploit values of r that are greater than 0.5. Table 1 illustrates the number and the maximum size of the returned clusters, and the coverage percentage of RANCoC, averaged over the 50 runs, for varying values of r. Also, the standard deviation over the 50 runs is shown within brackets.

As expected, the algorithm returns a greater number of clusters of smaller size as r increases, while the coverage percentage decreases for greater values of r. This behavior is explained by the fact that, when r has a low value, RANCoC is biased toward less dense groups of proteins; thus, a higher number of nodes can participate in a cluster. In particular, according to our experimental campaign, clusters obtained for greater values of r are contained (except for one or two proteins) in clusters obtained for lower values of r.

The low values for standard deviation scored over the 50 runs (from 0.001 to 15.447) confirm the stability of RANCoC.

In Table 2, the *nc-score* values obtained w.r.t. the three different GO vocabularies (called *process*, *component*, and *function* for short) are shown for both yeast and human. For all the three vocabularies, the clusters returned for $r = 1$ are the most biologically relevant; thus, we set the value of r equal to 1 in the experimental validations concerning GO annotations.

As a further set of evaluations, we computed also the overlapping score w.r.t. known protein complexes downloaded from the MIPS database [2], [3] for different values of the parameter r. These tests aimed at understanding which is the optimal value of r in recognizing protein complexes. We recall that, as already specified before, a known complex K_c and a predicted cluster P_c are considered to be a match if their overlapping score $OS(P_c, K_c)$ is equal to or greater than a specific threshold σ_{OS}. Figs. 4a and 4b show the number of matched complexes when the overlapping score is greater than σ_{OS} for both yeast and human, respectively. The number of matched complexes is illustrated for overlapping score threshold σ_{OS} varying from 0.05 to 0.45, and for values of r



(a) (b)

Fig. 4. Number of matched complexes for different values of r, with σ_{OS} between 0.05 and 0.45, for: (a) yeast and (b) human.

equal to 0.5, 1, 2, 3, and 4. Differently from the previous tests, where $r = 1$ resulted to guarantee the best results according to our purposes, looking around a reasonable threshold value $\sigma_{OS} = 0.2$ [26], $r = 3$ seems to be the optimal value to be set for protein complexes detection; thus, we set $r = 3$ for the analysis we performed on the MIPS complexes.

We finally note that we used the same values of r also when RANCoC is executed in the overlapping case since, as already pointed out, increasing values of r bias the method toward denser clusters, independently of the multiple appearing of a protein in different clusters.

3.3 Comparisons with the Other Methods

We compared the results returned by our system with those of other methods, in both the nonoverlapping and the overlapping operating mode. For the nonoverlapping case, we considered MCODE [9], RNSC [25], and MCL [19]. For the overlapping case, we compared our method with MCODE, CFINDER [31], DME [22], and IPCA [26]. For all the considered techniques, we took into account only clusters with size greater than or equal to two, by neglecting singletons in our analysis. Furthermore, for each system we compared with, we set the corresponding parameters by choosing, among those suggested by the authors, that configuration corresponding to the best results for the considered method. We briefly recall the main features of these methods in the following.

MCODE: *Molecular COmplex DEtection* [9] is a method that detects dense and connected regions by weighting nodes on the basis of their local neighborhood density. MCODE performs three main steps. In the first step, nodes are weighted. In the second step, the vertex with the highest weight is selected as seed cluster, and neighborhoods nodes are included in the cluster if their weight is above a fixed threshold. The neighbors of this node are recursively checked to verify if they can be part of the complex. When no more nodes can be added to the cluster, the process stops and it is repeated for the next-highest unexamined node. Postprocessing is finally optionally executed to filter proteins according to certain connectivity criteria. The method can be exploited to extract both overlapping and nonoverlapping clusters. We run it, in both cases, with the best parameter configuration reported by Brohèe and van Helden in [12], that is 0 for the *node score percentage* and 0.2 for *complex fluffing*.

RNSC: The *Restricted Neighborhood Search Clustering* Algorithm [25] searches for a *low-cost* clustering by first composing an initial random clustering, and then iteratively moving one node from one cluster to another in a randomized fashion to improve a specific cost function. The RNSC approach resembles our approach, however, RNSC uses two different cost functions. The first one computes the number of bad connections incident with a node. The second one measures the size of the area that a node effects in the clustering. RNSC is able to detect only nonoverlapping clusters.

MCL: The *Markov CLuster* algorithm [19] exploits algebraic processes defined on stochastic matrices to manage alternate expansions and contractions of flow simulations of the input graphs. The heuristics underlying such an approach is the expectation that flow between dense regions, which are sparsely connected, will evaporate. The input graph can be both weighted and directed. The input parameters requested by both MCL and RNSC have been set by using the best values obtained by Brohèe and van Helden in [12].

CFINDER: Palla et al. [31] presented a method based on locating all cliques (maximal complete subgraphs) of an input network and then identifying the clusters (called *communities*) by carrying out a standard component analysis of the clique-clique overlap matrix. In particular, the algorithm first determines from the degree-sequence the largest possible clique size in the input network. Then, starting with such a clique size, CFINDER repeatedly chooses a node, extracts every clique of such a largest size containing that node, and deletes the node and its edges. When no nodes are left, the clique size is decreased by 1 and the clique finding procedure is restarted on the original graph. CFINDER allows for overlapping clustering.

DME: *Dense Module Enumeration* [22] is the most recent method of the considered approaches for extracting dense modules from a weighted interaction network. It allows to incorporate constraints with respect to additional data sources. DME detects all the node subsets that satisfy a user-defined minimum density threshold. The method returns only locally maximal solutions, i.e., modules where all the direct supermodules (containing one additional node) do not satisfy the minimum density threshold. The obtained modules are ranked according to the probability that a random selection of the same number of nodes produces a module with at least the same density.

IPCA: This method [26] is a variation of the method DPCLUS, previously proposed in [6], that searches for subgraph structures having small diameters, i.e., small average vertex distance. Analogously to DPCLUS, IPCA starts by assigning a weight to each vertex on the base of the number of shared neighbors. Nodes are then ordered with respect to their weight, and considered as seeds for cluster detection, by picking at first the highest weighted vertices. Then, a cluster is extended by recursively adding neighboring nodes that satisfy a property of being strongly connected with the current cluster. The concept of strong connection between a node v and a cluster S is defined as the ratio between the number of edges between the vertexes v and S, and the number of nodes in S. If this ratio is above a threshold T_{in}, then the node is added to the cluster.

3.3.1 Nonoverlapping Case

We first analyze the nonoverlapping case for both the yeast and the human networks. Table 3 shows the coverage percentage, the number of returned clusters and the size of the greatest cluster for each of the compared methods. From such table, it is possible to see that MCL and RANCoC obtained the highest values for the coverage percentage, and they returned comparable number of clusters with comparable maximum size for both the yeast and the human networks. In Fig. 5, both the clustering score and the normalized clustering score for the three GO vocabularies (process, component, and function, for short) are graphically illustrated for the four considered methods on the yeast and human data sets. Figs. 5a and 5b show that RANCoC is below MCODE for both process and component annotations w.r.t. the clustering score on yeast, while

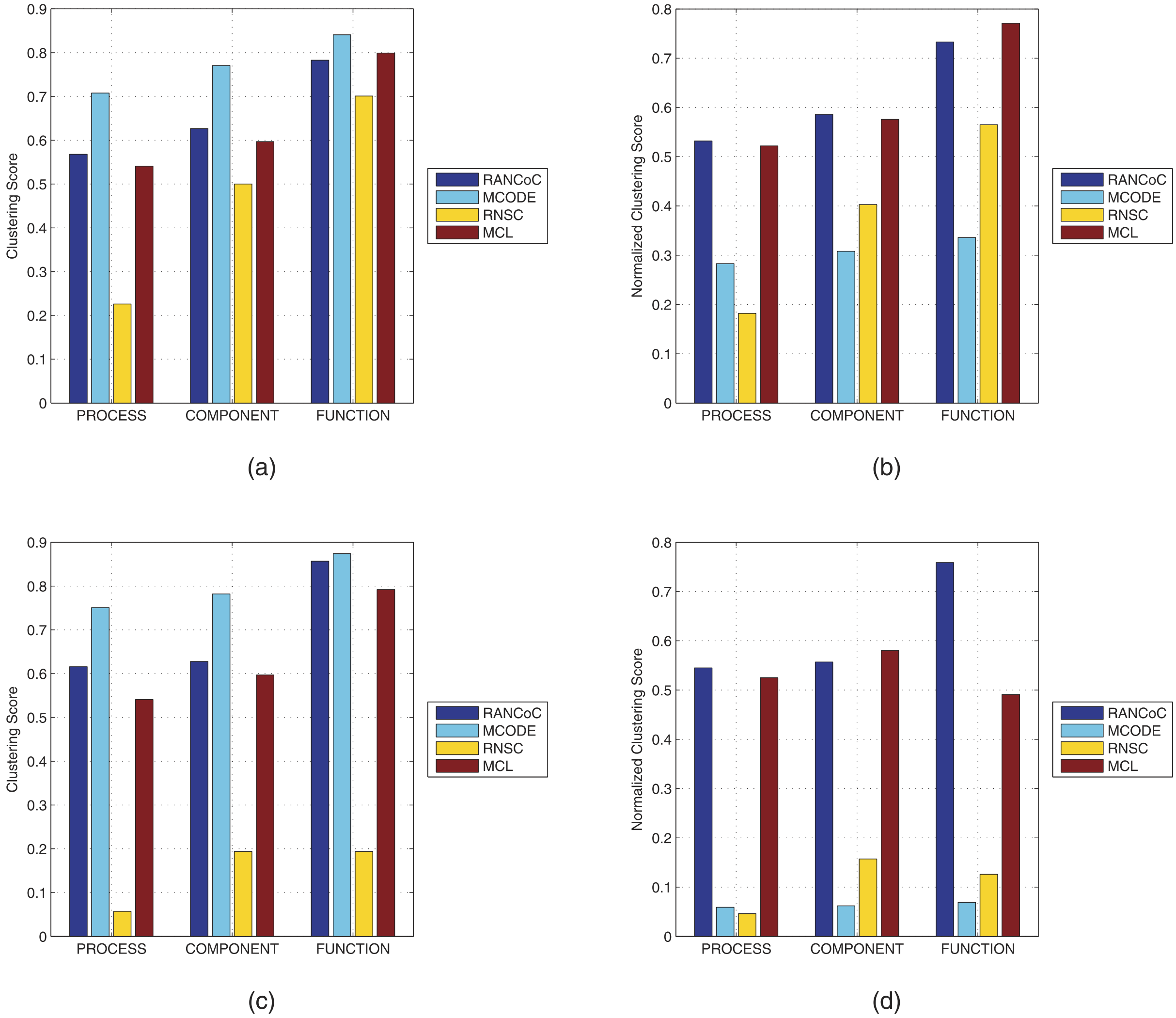
TABLE 3
The Nonoverlapping Case for Yeast and Human

METHOD	YEAST			HUMAN		
	Coverage Percentage	*Number of Clusters*	*Max Size*	*Coverage Percentage*	*Number of Clusters*	*Max Size*
RANCoC	0.935	673	95	0.886	945	94
MCODE	0.399	102	167	0.079	61	128
RNSC	0.806	1, 324	52	0.810	2, 820	34
MCL	0.965	866	96	0.972	1, 251	132

MCL is slightly better for the function annotations. The worst method in this evaluation is RNSC, which scored very low values of clustering score almost on the process annotations. For the normalized clustering score, instead, on the yeast network RANCoC performed the best values for both the process and component annotations, while it is outperformed by MCL for the function annotations. Anyway, we can say that both RANCoC and MCL returned significantly better results than the other two methods. As regards the human network, as shown in Figs. 5c and 5d, RANCoC results are again below MCODE for the clustering score, but RANCoC and MCL obtain a normalized clustering score much higher than that obtained by MCODE and RNSC. In particular, the former is the best for both process and function annotations, while the latter is the best for the component vocabulary.

Looking again at Table 3, we can argue why MCODE obtained very high values for the clustering score but not for the normalized clustering score. Indeed, this tool is more accurate but it is able to cover only a small portion of the input network. Finally, as regards RNSC, we observe that it returned many clusters of small sizes, and this possibly caused the lower clustering score and normalized clustering score w.r.t. the other three techniques.

We also point out that RANCoC was able to correctly separate groups of proteins whose functions are known



(a) (b)

(c) (d)

Fig. 5. Clustering Score and Normalized Clustering Score, in the Nonoverlapping Case for: (a-b) yeast; (c-d) human.

TABLE 4
The Overlapping Case for Yeast and Human

METHOD	YEAST			HUMAN		
	Coverage Percentage	*Number of Clusters*	*Max Size*	*Coverage Percentage*	*Number of Clusters*	*Max Size*
RANCoC	0.947	1,945	157	0.890	1,452	225
MCODE	0.804	134	903	0.160	95	175
CFINDER	0.367	238	957	0.106	53	144
DME	0.420	2,859	17	0.147	781	5
IPCA	0.983	3,413	119	0.985	5,292	75

from the literature. For example, in the nonoverlapping mode, RANCoC found the well-characterized group of proteins participating to *actin cytoskeleton organization and biogenesis*, as discussed by [18]. MCODE, CFINDER, and RNSC failed in grouping together such proteins, or clustered them in groups scoring worse p-value than those obtained by RANCoC.

3.3.2 Overlapping Case

Table 4 shows the coverage percentage, the number of returned clusters and the size of the greatest cluster for the five compared methods in the overlapping case. For both the yeast and the human network, IPCA scores the highest value of coverage percentage, although RANCoC is able to reach almost the same values of this measure while returning a much smaller number of clusters than IPCA. We also point out that the coverage percentages of the other methods for the human network are very poor.

Results of clustering score and normalized clustering score are shown in Figs. 6a and 6b for yeast and in Figs. 6c and 6d for human, respectively. All the compared methods scored high values of clustering score on the yeast network, while DME seems to be the less accurate on the human network for this measure. For the normalized clustering score, RANCoC scored the best values on both the yeast and human networks and for all the three GO vocabularies. In particular, on the yeast network also MCODE performed well in terms of normalized clustering score, and the behavior of all the methods is in general not bad for this network. On the contrary, for the human network, the resulting normalized clustering scores of all the other methods are worse than those of RANCoC, in most cases also significantly. This confirms the robustness of our method in analyzing PPI networks independently of how much characterized they are.

3.3.3 Discussion

We now consider a comparative analysis of all the considered methods. Figs. 7a and 7b illustrate diagrams of the normalized clustering score for both the yeast and the human PPI networks, for the three GO vocabularies. In such figures, RANCoC-OV and MCODE-OV denote the two methods in the overlapping mode. For both yeast and human, RANCoC in the overlapping mode is the best one, followed by MCODE in the overlapping mode on the yeast network and by MCL on the human network. For yeast, RANCoC in both the working modes, MCL and MCODE overlapping obtain values that are significatively better than the other tools. For human, only the former three methods perform the best values of clustering score, while MCODE has the worst performances on that network.

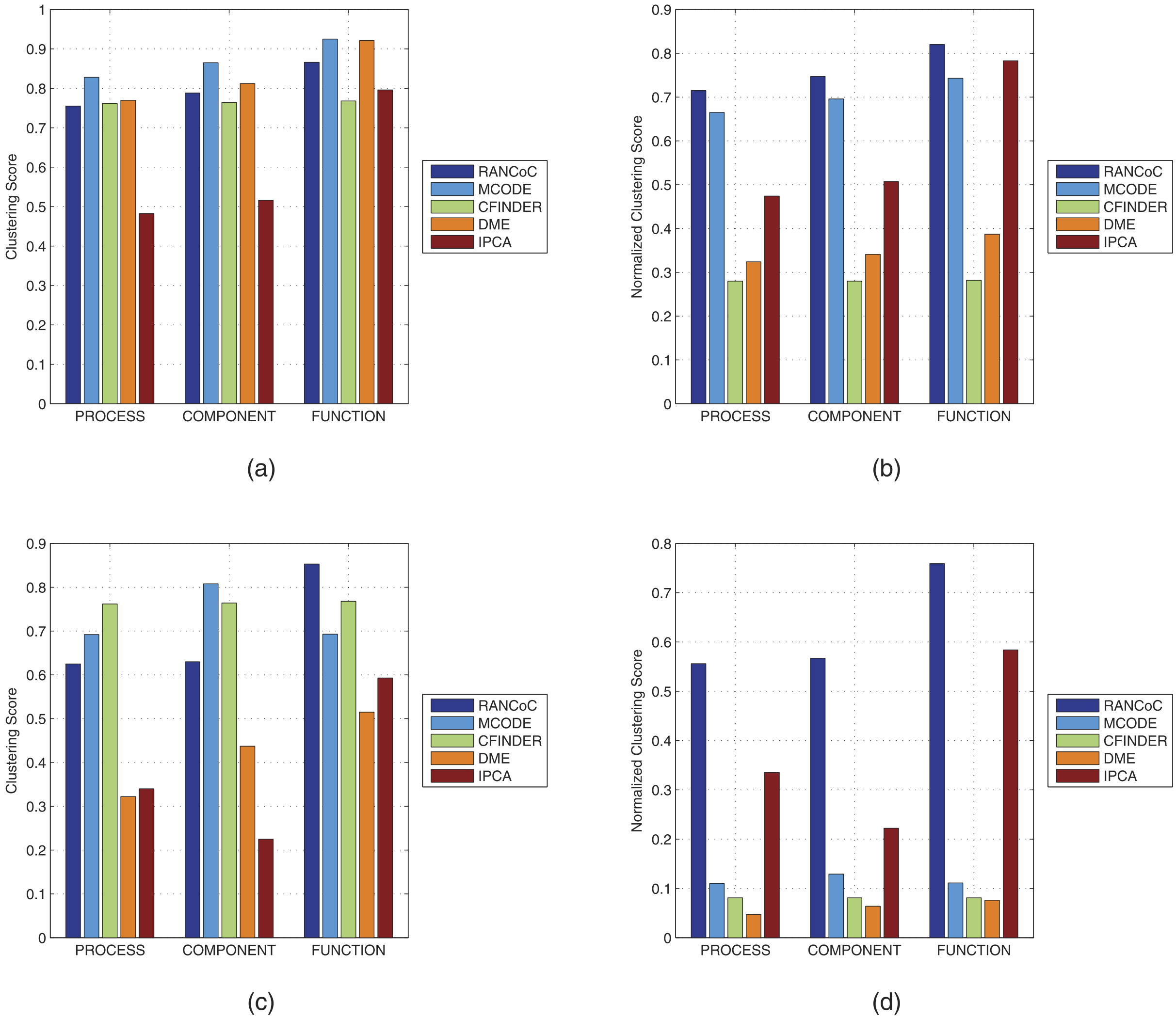
We can conclude that our technique seems to be the best one guaranteeing both high biological significance and network coverage, in both the nonoverlapping and the overlapping case, and it is robust since its performances are comparable on both the two analyzed networks.

3.4 MIPS Complexes Validation

We downloaded 975 known and curated complexes for yeast from [2] and 1,083 known and curated complexes for human from [3]. The size of each complex can vary from 2 to about 200 proteins, although most of the considered complexes are quite small, and the same protein can belong to different complexes. According to the analysis illustrated in Section 3.2, in the following evaluation, in order to compare the ability of RANCoC and the other approaches in predicting known protein complexes, we set equal to 3 the parameter r of RANCoC for both the nonoverlapping and overlapping mode. Figs. 8a, 8b, 8c, and 8d show a comparison among the different methods with respect to the number of matched complexes for the values 0.05, 0.1, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45 of the overlapping score threshold σ_{OS}. In particular, Fig. 8a shows the number of known complexes matched by RANCoC, RNSC, MCL, and MCODE on the yeast network. The figure points out that RANCoC predicts a higher number of known complexes than the other techniques. Fig. 8b shows the same results for the human network. In such a case, RANCoC and RNSC alternatively finds the higher number of matched complexes. As regards the methods that obtain overlapped modules, Fig. 8c shows that DME and RANCoC predict almost the same number of complexes, much higher than the other three methods on the yeast network, while on the human network, Fig. 8d, RANCoC performs the best. Note that such an analysis has meaning only in comparative terms, since the reference MIPS complexes do not cover all the considered PPI networks and some of the many predicted clusters that may be true complexes, could be regarded as false positives if they do not match with the known complexes [26].

Figs. 9a, 9b, 9c, and 9d show the sensitivity and specificity of all the methods for values of the overlapping score threshold σ_{OS} equal to 0.05, 0.1, 0.15, 0.20, 0.25, and 0.30. The corresponding values of sensitivity and specificity are those reported in the figures starting from right to left, and top to bottom. Fig. 9a points out that RANCoC obtains the higher values of both these two measures on the yeast network. As regards the human network (Fig. 9b), MCL obtains higher values of S_n and S_p for $\sigma_{OS} < 0.2$, while the sensitivity of RANCoC is better for $\sigma_{OS} \geq 0.2$. Figs. 9c and 9d show that DME overcome the other overlapping methods as regards the specificity, but the sensitivity of RANCoC-OV is higher with respect to all the other methods.

However, we note that specificity is less meaningful than sensitivity in this kind of analysis, since there could be lots



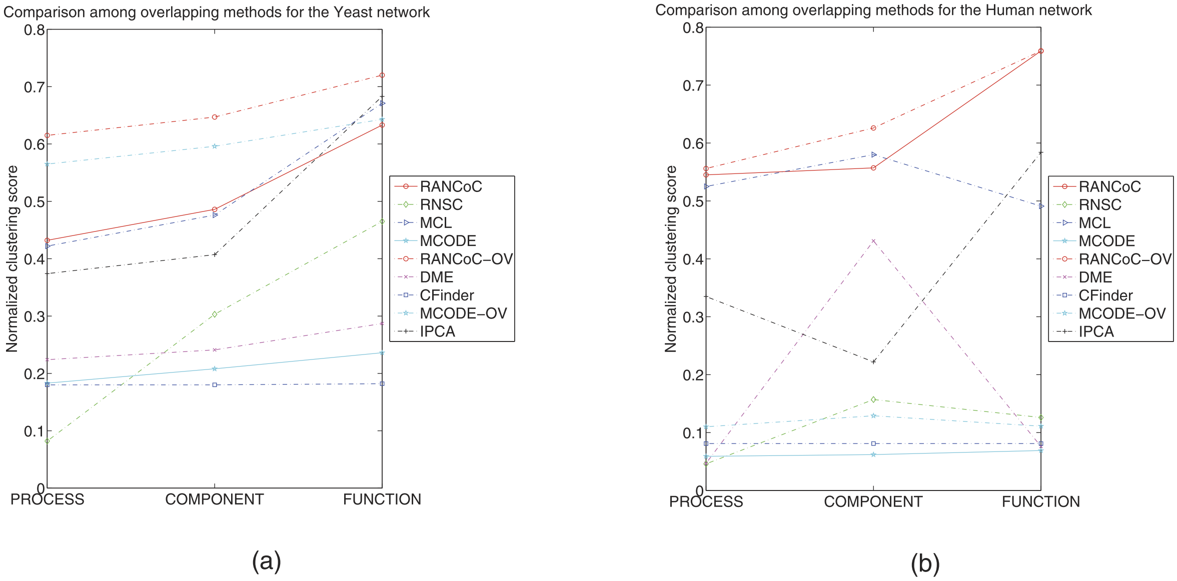
(a)

(b)

(c)

(d)

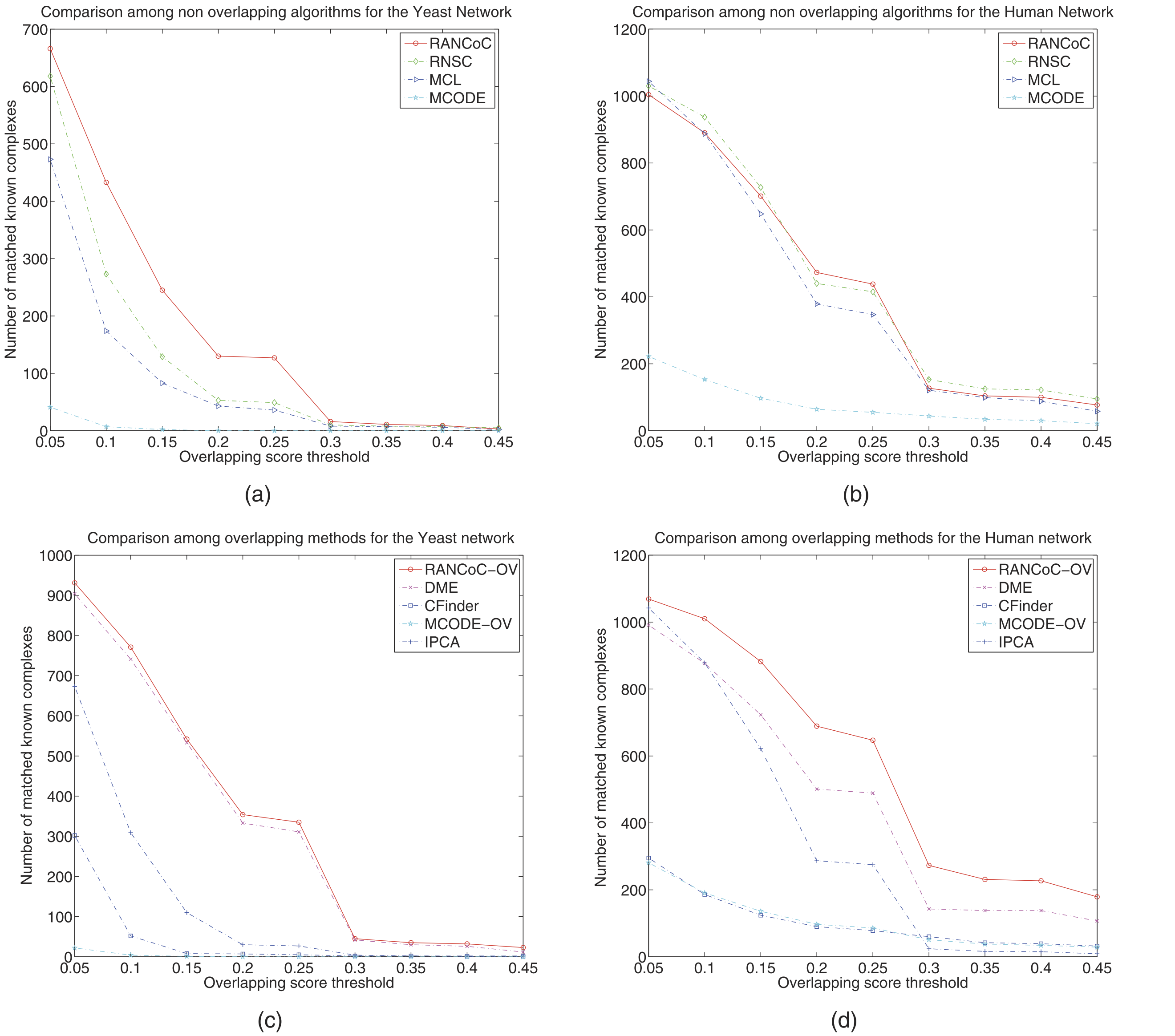
Fig. 6. Clustering Score and Normalized Clustering Score, in the overlapping case for: (a-b) yeast; (c-d) human.



(a)

(b)

Fig. 7. Normalized clustering score of all the methods in both the nonoverlapping and the overlapping case for (a) the yeast PPI network and (b) the human PPI network.



(a)

(b)

(c)

(d)

Fig. 8. Number of matched complexes by the different methods in the nonoverlapping mode for (a) yeast, (b) human, in the overlapping mode (c) yeast, and (d) human.

of complexes that are not yet known and annotated in the MIPS database.

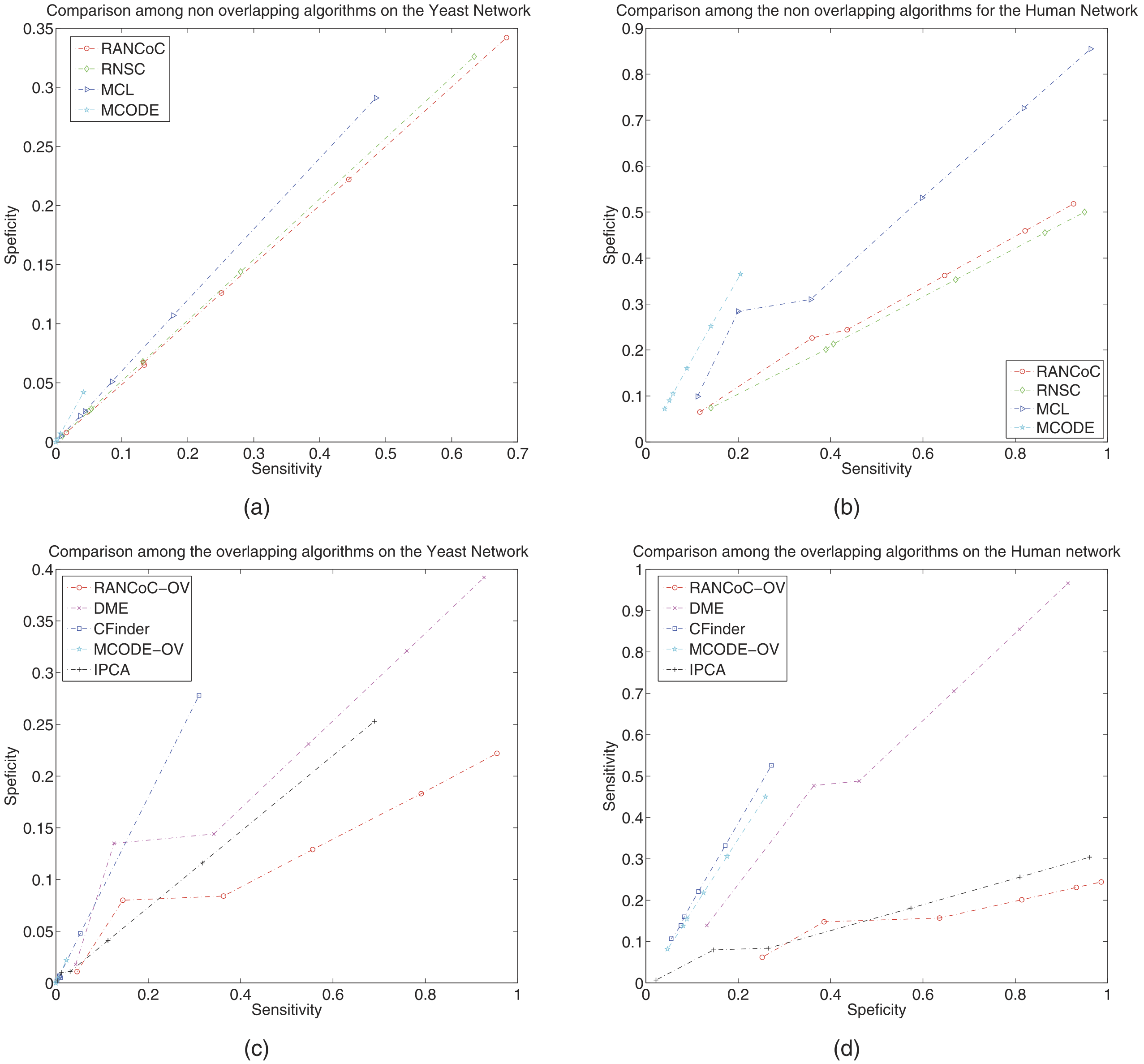
In conclusion, this validation campaign showed that RANCoC can be also usefully exploited in order to recognize protein complexes in PPI networks.

3.5 Multifunctional Proteins

We now mention some of the multifacets proteins that RANCoC grouped with other proteins in different clusters, each characterized by biological relevant meaning. We refer to proteins discussed by Ucar et al. [43] and to biological process GO annotations.

As reported in [43], KAP95 is an essential protein with many functionalities that is known to take part in *nucleocytoplasmatic transport*. RANCoC grouped KAP95 with other nine proteins (PBS2, ZDS1, YKL214C, NAP1, NUP1, NUP60, PCT1, ULP1, NUP2) participating to this same biological process with p-value $1.60 \cdot 10^{-10}$. Furthermore, RANCoC found this protein in other clusters. Among the most relevant clusters involving KAP95, we mention one containing 140 proteins participating in *macromolecule metabolic process* with p-value $5.16 \cdot 10^{-26}$, and another one containing proteins ATG16, VMA6, VPS5, PSE1, VPS17, SEC26, TPO1, PEP12, RET3, PMC1, VPH1, CHS5, VMA2, VAC8, YMR010W, SEC7, HXT1, FTH1, VPS35, DNF1, RET2, YBT1, ATP14, VMA1, GTR1, ATG27, DOP1, SEC2, DRS2, HXT2, VPS29, SEC21, SEC27, SFT2, BZZ1, VMA13, COG3, VMA8, MUP1, TVP15, GLO3, KAP95, VPS26, GTR2, HSP30, ATG19, AKR1, CTR2, ARF1, MEH1, PDR12, HNM1, NUP1, VMA5, VMA7, MIA40, COT1, FLC2, VPS68, YIP3 involved in *transport* and in *establishment of localization* with p-value $2.54 \cdot 10^{-19}$.

The hub protein LSM8 has been found by Ucar et al. [43] with other 10 proteins (LSM2, LSM3, LSM5, PRP3, PRP4, PRP6, PRP21, PRP31, SMB1, SPP381) with biological process *mRNA splicing* and p-value $1.2 \cdot 10^{-12}$. We found the same protein in several groups, in particular, it participates with PUB1, LSM5, LSM3, DHH1, DCP2, NOT4, KEM1, LSM4, LSM2, EDC3, POP2, PUF4, LSM7, LSM6, NOT1, LSM1, NOT3, CCR4, NOT2 and NOT5 to the



(a)

(b)

(c)

(d)

Fig. 9. Comparison of sensitivity and specificity for the nonoverlapping methods for (a) yeast, (b) human, and the overlapping methods for (c) yeast, (d) human. The values of the overlapping score threshold σ_{OS} have been fixed to 0.05, 0.1, 0.15, 0.20, 0.25, and 0.30.

mRNA catabolic process with p-value $1.32 \cdot 10^{-39}$, and with PUB1, LSM5, LSM3, DHH1, NOT4, DCP2, CAF40, KEM1, LSM4, LSM2, DCP1, EDC3, POP2, LSM7, PUF4, LSM6, NOT1, LSM1, NOT3, NOT2, CCR4, NOT5 to the *RNA catabolic process* with p-value $1.05 \cdot 10^{-41}$.

CKA1 is a protein involved in several cellular events. Ucar et al. located CKA1 in three different partitions. One is annotated with the biological process *transcription, DNA-dependent* and p-value $2.3 \cdot 10^{-19}$, the second one with *protein amino acid phosphorylation* and p-value $1.2 \cdot 10^{-05}$, the third group is annotated with *organelle organization and biogenesis* and p-value $3.2 \cdot 10^{-12}$. RANCoC found, among the others, a group with p-value $4.40 \cdot 10^{-25}$ and annotation *cellular component organization and biogenesis*, the group CEG1, CKA2, LEO1, SPT16, FKH1, CTR9, HTA1, RTF1, CKA1, CDC73, CKB2, PAF1, CKB1, HTB1, POB3, CHD1 with p-value $2.98 \cdot 10^{-12}$ and annotation *regulation of transcription, DNA-dependent*; the group CKA1, CKB2, CKA2, CKB1 involved in *regulation of transcription from RNA polymerase III promoter* with p-value $1.08 \cdot 10^{-08}$ and other two groups involved in *response to DNA damage stimulus* and *regulation of nucleobase, nucleoside, nucleotide, and nucleic acid metabolic process* with p-value $2.83 \cdot 10^{-10}$ and $2.95 \cdot 10^{-07}$, respectively.

4 Conclusions

Overlapping and nonoverlapping clustering of PPI networks are important analysis methods that allows to uncover and understand the complex structure of interconnections among proteins. Nonoverlapping clustering is usually exploited when separating groups of proteins with different biological meaning is the main aim. On the other hand, overlapping clustering allows to identify proteins involved in several biological processes. The algorithm RANCoC supports both the possibilities and an exploration of the network at different resolution levels. Indeed, the choice of the parameter r allows for a suitable tradeoff between the coverage of the network and the biological relevance of the output solution. An extensive experimental evaluation showed that our method outperforms other state-of-the-art approaches in finding a good compromise between accuracy and network coverage. Furthermore, the behavior of RANCoC is not influenced by how much characterized (and/or dense) is the input network. Finally, RANCoC showed to outperform the other considered approaches in correctly predicting known and manually curated MIPS complexes.

As future work, we plan to apply our approach to cluster also other types of biological networks, e.g., metabolic networks. Furthermore, we think to test also other strategies to avoid get trapped in local optima, such as, for example, tabu search and simulated annealing.

Acknowledgments

The authors are grateful to Luigi Palopoli for his suggestions useful to improve the quality of the paper. Many thanks also to the anonymous Reviewers for their precious comments.

References

[1] Int'l Molecular Interaction Exchange (imex) Consortium of Molecular Interaction Databases: http://imex.sf.net, 2012.

[2] Website title: ftp://ftpmips.gsf.de/yeast/catalogues/complexcat, 2012.

[3] Website title: http://mips.helmholtz-muenchen.de/genre/proj/corum, 2012.

[4] B. Adamcsek, G. Palla, I.J. Farkas, I. Dernyi, and T. Vicsek, "Cfinder: Locating Cliques and Overlapping Modules in Biological Networks," *Bioinformatics,* vol. 22, no. 8, pp. 1021-1023, 2006.

[5] B. Aittokallio and B. Schwikowski, "Graph-Based Methods for Analyzing Networks in Cell Biology," *Briefing in Bioinformatics,* vol. 7, no. 3, pp. 243-255, 2006.

[6] M. Altaf-Ul-Amin, Y. Shinbo, K. Mihara, K. Kurokawa, and S. Kanaya, "Development and Implementation of an Algorithm for Detection of Protein Complexes in Large Interaction Networks," *BMC Bioinformatics,* vol. 7, article 207, 2006.

[7] V. Arnau, S. Mars, and I. Marìn, "Iterative Cluster Analysis of Protein Interaction Data," *Bioinformatics,* vol. 21, no. 3, pp. 364-378, 2005.

[8] S. Asburner et al., "Gene Ontology: Tool for the Unification of Biology. The Gene Ontology Consortium," *Nature Genetics,* vol. 25, pp. 25-29, 2000.

[9] G. Bader and H. Hogue, "An Automated Method for Finding Molecular Complexes in Large Protein-Protein Interaction Networks," *BMC Bioinformatics,* vol. 4, article 2, 2003.

[10] A. Barabási and Z.N. Oltvai, "Network Biology: Understanding the Cell's Functional Organization," *Nature Reviews Genetics,* vol. 5, pp. 101-113, 2004.

[11] M. Blatt, S. Wiseman, and E. Domany, "Superparamagnetic Clustering of Data," *Physical Review Letters,* vol. 76, no. 18, pp. 3251-3254, 1996.

[12] S. Brohèe and J. van Helden, "Evaluation of Clustering Algorithms for Protein-Protein Interaction Networks," *BMC Bioinformatics,* vol. 7, article 488, 2006.

[13] C. Brun, C. Herrmann, and A. Guenoche, "Clustering Proteins from Interaction Networks for the Prediction of Cellular Functions," *BMC Bioinformatics,* vol. 5, article 95, 2004.

[14] A. Ceol et al., "Mint, the Molecular Interaction Database: 2009 Update," *Nucleic Acids Research,* vol. 38, pp. D532-D539, 2010.

[15] Y.-R. Cho, W. Hwang, M. Ramanathan, and A. Zhang, "Semantic Integration to Identify Overlapping Functional Modules in Protein Interaction Networks," *BMC Bioinformatics,* vol. 8, article 265, 2007.

[16] Y.-R. Cho, W. Hwang, and A. Zhang, "Identification of Overlapping Functional Modules in Protein Interaction Networks: Information Flow-Based Approach," *Proc. Sixth Int'l Conf. Data Mining-Workshops (ICDMW '06),* 2006.

[17] I. Derenyi, G. Palla, and T. Vicsek, "Clique Percolation in Random Networks," *Physical Review Letters,* vol. 94, no. 16, pp. 160-202, 2005.

[18] B.L. Drees et al., "A Protein Interaction Map for Cell Polarity Development," *J. Cellular Biology,* vol. 154, pp. 549-571, 2001.

[19] A.J. Enright, S.V. Dongen, and C.A. Ouzounis, "An Efficient Algorithm for Large-Scale Detection of Protein Families," *Nucleic Acids Research,* vol. 30, no. 7, pp. 1575-1584, 2002.

[20] V. Farutin, K. Robinson, E. Lightcap, V. Dancik, A. Ruttenberg, S. Letovsky, and J. Pradines, "Edge-Count Probabilities for the Identification of Local Protein Communities and Their Organization," *Proteins: Structure, Function, and Bioinformatics,* vol. 62, pp. 800-818, 2006.

[21] A.C. Gavin et al., "Proteome Survey Reveals Modularity of the Yeast Cell Machinery," *Nature,* vol. 440, pp. 631-636, 2006.

[22] E. Georgii, S. Dietmann, T. Uno, P. Pagel, and K. Tsuda, "Enumeration of Condition-Dependent Dense Modules in Protein Interaction Networks," *Bioinformatics,* vol. 25, no. 7, pp. 933-940, 2009.

[23] L.H. Hartwell, J.J. Hopfield, S. Leibler, and A.W. Murray, "Clustering Algorithm Based Graph Connectivity," *Nature,* vol. 402, pp. C47-C52, 1999.

[24] W. Hwang, Y.-R. Cho, A. Zhang, and M. Ramanathan, "A Novel Functional Module Detection Algorithm for Protein-Protein Interaction Networks," *Algorithms for Molecular Biology,* vol. 1, no. 24, 2006.

[25] A.D. King, N. Przulj, and I. Jurisica, "Protein Complex Prediction via Cost-Based Clustering," *Bioinformatics,* vol. 20, no. 17, pp. 3013-3020, 2004.

[26] M. Li, J. Chen, J. Wang, B. Hu, and G. Chen, "Modifying the DPClus Algorithm for Identifying Protein Complexes Based on New Topological Structures," *BMC Bioinformatics,* vol. 9, 2008.

[27] C. Lin, Y. Cho, W. Hwang, P. Pei, and A. Zhang, "Clustering Methods in Protein-Protein Interaction Network," *Knowledge Discovery in Bioinformatics: Techniques, Methods and Application,* John Wiley and Sons, Inc., 2006.

[28] Z. Lubovac, J. Gamalielsson, and B. Olsson, "Combining Functional and Topological Properties to Identify Core Modules in Protein Interaction Networks," *Proteins: Structure, Function, and Bioinformatics,* vol. 64, pp. 948-959, 2006.

[29] S.C. Madeira and A.L. Oliveira, "Biclustering Algorithms for Biological Data Analysis: A Survey," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 1, no. 1, pp. 24-45, Jan.-Mar. 2004.

[30] H.W. Mewes, D. Frishman, U. Güldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Münsterkötter, S. Rudd, and B. Weil, "MIPS: A Database for Genomes and Protein Sequences," *Nucleic Acids Research,* vol. 30, no. 1, pp. 31-34, 2002.

[31] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, "Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society," *Nature,* vol. 435, pp. 814-818, 2005.

[32] P. Pei and A. Zhang, "A Two-Step Approach for Clustering Proteins Based on Protein Interaction Profiles," *Proc. IEEE Int'l Symp. Bioinformatics and Bioengeneering (BIBE '05),* pp. 201-209, 2005.

[33] J.B. Pereira, A.J. Enright, and C.A. Ouzounis, "Detection of Functional Modules from Protein Interaction Networks," *Proteins: Structure, Functions, and Bioinformatics,* vol. 20, pp. 49-57, 2004.

[34] C. Pizzuti and S.E. Rombo, "PINCoC: A Co-Clustering Based Approach to Analyze Protein-Protein Interaction Networks," *Proc. Eighth Int'l Conf. Intelligent Data Eng. and Automated Learning (IDEAL '07),* pp. 821-830, 2007.

[35] C. Pizzuti and S.E. Rombo, "Multi-Functional Protein Clustering in PPI Networks," *Proc. Second Int'l Conf. Bioinformatics Research and Development (BIRD '08),* pp. 318-330, 2008.

[36] C. Pizzuti and S.E. Rombo, "Discovering Protein Complexes in Protein Interaction Networks," *Biological Data Mining in Protein Interaction Networks,* X.-L. Li and S.-K. Ng, eds., IGI Global-Medical Inf. Science Ref., 2009.

[37] N. Przulj, "Functional Topology in a Network of Protein Interactions," *Knowledge Discovery in Proteomics,* I. Jurisica and D. Wigle, eds. CRC Press, 2005.

[38] A.W. Rives and T. Galitski, "Modular Organization of Cellular Networks," *Proc. Nat'l Academy of Science USA,* vol. 100, no. 3, pp. 1128-1133, 2003.

[39] M.P. Samantha and S. Liang, "Predicting Protein Functions from Redundancies in Large-Scale Protein Interaction Networks," *Proc. Nat'l Academy of Science USA,* vol. 100, no. 22, pp. 12579-12583, 2003.

[40] R. Sharan, I. Ulitsky, and R. Shamir, "Network-Based Prediction of Protein Function," *Molecular Systems Biology,* vol. 3, no. 88, 2007.

[41] V. Spirin and L.A. Mirny, "Protein Complexes and Functional Modules in Molecular Networks," *Proc. Nat'l Academy of Science USA,* vol. 100, pp. 12123-12128, 2003.

[42] S. Tornw and H.W. Mewes, "Functional Modules by Relating Protein Interaction Networks and Gene Expression," *Nucleic Acids Research,* vol. 31, no. 21, pp. 6283-6289, 2003.

[43] D. Ucar, S. Asur, Ü.V. Çatalyürek, and S. Parthasarathy, "Improving Functional Modularity in Protein-Protein Interactions Graphs Using Hub-Induced Subgraphs," *Proc. 10th European Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD),* pp. 371-382, 2006.

[44] D. von Mering et al., "Comparative Assessment of a Large-Scale Data Sets of Protein-Protein Interactions," *Nature,* vol. 31, pp. 399-403, 2002.

[45] S. Zhang, H.-W. Liu, X.-M. Ning, and X.-S. Zhang, "A Graph Theoretic Method for Mining Functional Modules in Large Sparse Protein Interaction Networks," *Proc. IEEE ICDM Workshop Data Mining in Bioinformatics (ICDMW '06),* pp. 130-135, 2006.

[46] E. Zotenko, K.S. Guimaraes, R. Jothi, and T.M. Przytycka, "Decomposition of Overlapping Protein Complexes: A Graph Theoretical Method for Analyzing Static and Dynamic Protein Associations," *Algorithms for Molecular Biology,* vol. 1, no. 7, 2006.

Clara Pizzuti received the Laurea degree in mathematics from the University of Calabria, Italy. She is a senior researcher in the Institute for High Performance Computing and Networking (ICAR) at the National Research Council of Italy (CNR). Since 1995, she has been a contract professor in the Department of Electronics, Computer Science and Systems (DEIS) at the University of Calabria. In the past, she worked in the research division of a software company on deductive databases, advanced logic-based systems, and abduction. Her research interests include knowledge discovery in databases, data mining, data streams, bioinformatics, e-health, social network analysis, evolutionary computation, genetic algorithms, and genetic programming.

Simona E. Rombo received the PhD degree in computer science, biomedicine, and telecommunications from the University "Mediterranea" of Reggio Calabria. She is a research fellow in the Institute for High Performance Computing and Networking (ICAR) at the National Research Council of Italy (CNR). For four years, she was research fellow in the Department of Electronics, Computer Science and Systems (DEIS) at the University of Calabria. She was a visiting scientist at Purdue University, Georgia Institute of Technology, and Freie Universitat of Berlin. Her research interests include algorithms and data structures, bioinformatics, combinatorics, and pattern discovery.