Multi-functional Protein Clustering in PPI Networks

Clara Pizzuti[1] and Simona E. Rombo[2]

[1] ICAR-CNR, Via P. Bucci 41C, 87036 Rende (CS), Italy,
pizzuti@icar.cnr.it
[2] DEIS - Università della Calabria, Via P. Bucci 41C, 87036 Rende (CS), Italy,
simona.rombo@deis.unical.it

Abstract. Protein-Protein Interaction (PPI) networks contain valuable information for the isolation of groups of proteins that participate in the same biological function. Many proteins play different roles in the cell by taking part in several processes, but isolating the different processes in which a protein is involved is often a difficult task. In this paper we present a method based on a greedy local search technique to detect functional modules in PPI graphs. The approach is conceived as a generalization of the algorithm PINCoC to generate overlapping clusters of the interaction graph in input. Due to this peculiarity, multi-facets proteins are allowed to belong to different groups corresponding to different biological processes. A comparison of the results obtained by our method with those of other well known clustering algorithms shows the capability of our approach to detect different and meaningful functional modules.

1 Introduction

Proteins are the building blocks of all organisms and play a fundamental role in executing and regulating most biological processes. Recently, it has been noted that, to fully understand cell activity, proteins cannot be analyzed independently from the other proteins because they seldom act in isolation to perform their tasks [25]. Advances in technology have allowed researches to derive, through experimental and in-silico methods, the collection of all interactions between proteins of an organism. The availability of protein-protein interaction (PPI) networks has thus stimulated the search for automated and accurate tools to analyze pair-wise protein interactions with the aim of extracting relevant functional modules. A functional module is a group of proteins participating to the same biological function. Their detection provides important knowledge to better understand the behavior of organisms.

PPI networks are naturally modelled as graphs where nodes represent proteins and edges represent pairwise interactions. Dense regions of a given PPI network correspond to highly interacting proteins that could be involved in common biological processes. One of the main difficulty in analyzing PPI graphs is their scale-free topology. A scale-free graph is characterized by the property that the degrees k of vertices are distributed according to a power law function, as $P(k) \propto k^{-\alpha}$, where $\alpha > 0$. This implies that most proteins interact with only a few other proteins, while a small number of proteins, known as $hubs$, have many interactions. Hubs proteins have been investigated [13] and recognized to have an important role for the life of organisms. Typically, because of

their characteristic of being connected to a high number of proteins, they participate in multiple biological processes. Traditional clustering methods, however, assign a protein to only one group, which is unlikely for biological systems. In such a way these methods hamper the possibility of proteins to be clustered in several groups, on the basis of the different functions they have in the cell. This represents a significant inability of these approaches to describe the complexity of biological systems. To overcome such a problem, recent proposals have suggested different strategies [19, 24, 3].

In this paper, we present a partitioning technique of protein-protein interaction networks to produce overlapping clustering of the interaction graph. The algorithm, named Multi-Functional *PINCoC* (MF-*PINCoC*), is an extension of the method *PINCoC*, a *PPI network Co-Clustering* based algorithm, presented in [20], suitably modified to allow the participation of proteins to multiple functional groups. Co-clustering methods [16], differently from clustering approaches, aim at simultaneously grouping both the dimensions of a data set.

The PPI network is represented through the binary adjacency matrix A of the associated 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. The algorithm searches for, eventually overlapping, dense sub-matrices containing the maximum number of ones by using a greedy local search technique. It starts with an initial random solution constituted by a single protein and finds a locally optimal solution by adding/removing connected proteins that best contribute to improve a *quality* function. In order to enable participation of a protein to more groups, its degree k, i.e. the number of other proteins with which it is connected, is computed. A protein can be added to the current cluster if the number of clusters to which it has already been assigned is less than its degree. The method is enriched with one step of backtracking, to limit the effects of the initial random choice of a protein to build a cluster, and a remove strategy of proteins, to escape poor local optima. When the algorithm cannot improve any more the solution found so far, the computed cluster is returned. At this point a new random protein is chosen, and the process is repeated until all the proteins are assigned to a group.

MF-*PINCoC* has two fundamental advantages with respect to other approaches presented in the literature. The first, inherited from *PINCoC*, is that the number of clusters is automatically determined by the algorithm. The second, which is its main characteristic, is that for each protein interacting with other proteins, MF-*PINCoC* is able to identify the different groups in which the protein is involved, each group being distinguished by a different biological property. Note that, differently from other techniques [24], MF-*PINCoC* allows the participation to different clusters not only to the highly connected proteins recognized as hubs [3], but also to all the other proteins. Such a peculiarity is automatically incorporated in the approach without any lack in efficiency, and it avoids leaving possible candidates to be multi-facets proteins out from the analysis.

In the experimental result section we show that MF-*PINCoC* is able *(i)* to efficiently isolate groups of proteins corresponding to the most compact sets of interactions, and *(ii)* to assign proteins to more than one cluster, each characterized by a different biolog-

[3] In [24] the authors recognized as *hubs* those proteins involved in a number of interactions between 40 and 283.

ical function. A comparison with other well known protein clustering methods points out the very good results of our approach with respect to them.

The paper is organized as follows. The next section describes the MF-*PINCoC* algorithm and the variations introduced w.r.t. *PINCoC* to allow overlapping clusterings. Section 3 reports the related work on protein clustering. Section 4 illustrates the experiments carried out on the Saccaromyces cerevisiae protein data set and compares the obtained results with those of [4, 14, 24]. Finally, in Section 5 we draw our conclusions.

2 Approach description

In this section we recall the notation adopted by both the MF-*PINCoC* and *PINCoC* algorithms, and describe the extensions realized to allow for multiple group participation of proteins.

A PPI network $\mathcal{P}$ is modelled 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 between nodes i and j, 0 otherwise. The problem of finding dense regions of a PPI network $\mathcal{P}$ can be transformed in that of finding dense subgraphs of the graph G associated with $\mathcal{P}$, and consequently, dense sub-matrices of the adjacency matrix A corresponding to G. Searching for dense sub-matrices of such a matrix A can be viewed as a special case of co-clustering a binary data matrix where the set of rows and the set of columns represent the same concept. In order to better explain the idea, first a definition of co-clustering is given, and then the formalization of the problem of clustering proteins as a co-clustering problem is provided. Co-clustering [16, 7], also known as bi-clustering, differently from clustering, tries to simultaneously group both the dimensions of a data set. A *co-cluster* of a matrix A is defined as a sub-matrix $B = (I, J)$ of A, where I is a subset of the rows $X = \{I_1, \ldots, I_N\}$ of A, and J is a subset of the columns $Y = \{J_1, \ldots, J_M\}$ of A.

Then, the problem of co-clustering may be formulated as follows: given a data matrix A, find row and column maximal groups which divide the matrix into regions that satisfy some homogeneity characteristics. The kind of homogeneity a co-cluster has to fulfil depends on the application domain. In our case we would like to find as many proteins as possible having the highest number of interactions. This corresponds to identify highly dense squared sub-matrices, i.e., containing as many values equal to 1 as possible. Higher the number of ones, more likely those proteins are to be functionally related.

Let a_{iJ} denote the *mean value* of the ith row of the co-cluster $B = (I, J)$, and a_{Ij} the mean of the jth column of B. More formally,

$$a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{ij}, \text{ and } a_{Ij} = \frac{1}{|I|} \sum_{i \in I} a_{ij}$$

The *volume* v_B of a co-cluster $B = (I, J)$ is the number of 1 entries a_{ij} such that $i \in I$ and $j \in J$, that is $v_B = \sum_{i \in I, j \in J} a_{ij}$.

Given a co-cluster $B = (I, J)$, the *power mean of B of order r*, denoted by $\mathbf{M}_r(B)$ is defined as

$$\mathbf{M}_r(B) = \frac{\sum_{i \in I}(a_{iJ})^r + \sum_{j \in J}(a_{Ij})^r}{|I| + |J|}$$

A measure based on volume and row/column mean, that allows the detection of maximal and dense sub-matrices, can be defined as follows.

Given a co-cluster $B = (I, J)$, let $\mathbf{M}_r(B)$ be the power mean of B of order r. The *quality* of B is defined as

$$Q(B) = \mathbf{M}_r(B) \times v_B$$

The problem of protein clustering can be formulated as follows: given a data matrix A, find row and column maximal groups that partition the matrix into sub-matrices $\{B_1, \ldots B_h\}$, each having maximal $Q(B_i)$ values.

It is worth to note that high values of the exponent r bias the *quality* function towards matrices containing a low number of zeroes. In fact, it amplifies the weight of the densely interconnected nodes, while reducing those of less connected in the computation of the *quality* function. In the following the terms co-cluster, cluster, and sub-matrix are used to express the same concept.

MF-$PINCoC$ starts with an initial random cluster $B = (I_i, J_i)$ constituted by a single row and a single column such that $I = \{l\}$ and $J = \{l\}$, where $1 \leq l \leq N$ is a random row/column index. Then it evolves the initial cluster by successive transformations of B_i, until the *quality* function is improved. The transformations consist in the change of membership (called $flip$ or $move$) of the row/column that leads to the largest increase of the *quality* function. If a bit is set from 0 to 1 it means that the corresponding protein, which was not included in the cluster B_i, is added to B_i. Vice versa, if a bit is set from 1 to 0 it means that the corresponding protein is removed from the cluster. During its execution, in order to avoid getting trapped into poor local maxima, instead of performing the flip maximizing the $quality$, with a user-provided probability p the algorithm selects the row/column of B_i scoring the minimum decrease of the *quality* function, and removes it from B_i. This kind of flip is called REMOVE-MIN. The flips are repeated until either a preset of maximum number of flips is reached, or the solution cannot ulteriorly be improved (get trapped into a local maximum). Until the number of flips is below a fixed maximum value and the quality function increases, MF-*PINCoC* executes a REMOVE-MIN move with probability p, and a greedy move with probability $(1-p)$; otherwise, the cluster $B_i = (I_i, J_i)$ is returned. At this point the algorithm performs one step of backtracking, i.e., for each $h \in I_i$ it temporary removes h from I_i and tries to find a node l such that $I_i - \{h\} \cup \{l\}$ improves the *quality* of B_i. In such a case h is removed and l is added. If more than one l node exists, the one generating the better improvement of $Q(B_i)$ is chosen. Finally, B_i is added to B, its rows/columns are removed from A, a new random cluster is generated, and the process is repeated until all the rows/columns have been assigned.

As previously pointed out, many proteins may be involved in several biological functions by interacting with different groups of proteins. In order to allow these multi-facets proteins to be assigned to more than one cluster, we relax the constraint adopted in *PINCoC* to exclude a protein to be considered for inclusion in another cluster, once it has already been put into a group. To this end, for each protein we compute its degree k, i.e. the number of other proteins with which it is connected. When building a new

cluster, a protein can be added to the current cluster if the number of clusters to which it has already been assigned is less than its degree. In such a way each protein, not only hubs, can belong to multiple clusters, provided that its contribution to the *quality* function is effective, i.e. it is the choice that produces the best improvement. In the next section we report the main proposals to protein clustering recently presented in the literature.

3 Related Work

Clustering approaches to PPI networks can be broadly categorized as distance-based and graph-based [15] ones. Distance-based clustering approaches apply traditional clustering techniques by employing the concept of distance between two proteins [2, 18]. Graph-based clustering approaches consider the network topology and partition the graph trying to optimize a cost function [12, 5, 10, 4, 23, 22, 14, 19, 24]. In the following some of the main proposals are described.

Molecular complex detection (MCODE) [4] detects dense and connected regions by weighting nodes on the basis of their local neighborhood density. To this end, the k-core concept is applied. A k-core is a graph 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, i.e. the density of the highest k-core of the vertices directly connected to it, is then used to give a weight to each vertex. MCODE performs three steps: vertex weighting, complex prediction, and optional postprocessing to add or remove proteins. In the first step nodes are weighted according to the density of the highest k-core. In the second step the vertex with the highest weight is selected as seed cluster, and new nodes are included in the cluster if their weight is above a fixed threshold. This process is repeated for the next-highest unexamined node. In such a way the densest regions of the graph are identified. Postprocessing is finally optionally executed to filter proteins according to certain connectivity criteria.

The Restricted Neighborhood Search Clustering (RNSC), proposed by King et al. [14], is a cost-based local search algorithm that explores the solution space of all the possible clusterings to minimize a cost function that reflects the number of inter-cluster and intra-cluster edges. The idea resembles our approach, however, RNSC uses two cost functions. The first, called the naive cost function, for each node v, computes the number of bad connections incident with v, i.e. one that exists between v and a node not belonging to the same cluster of v, or one that does not exist between v and another node in the same cluster as v. The second one, called the scaled cost function, measures the size of the area that v effects in the clustering. The algorithm begins with a random clustering, and attempts to find a clustering with low naive cost by moving a vertex from a cluster to another one. Then it tries to improve the solution by searching for a clustering with low scaled cost. Differently from the approach presented here, neither MCODE nor RNSC allow the participation of a protein to multiple clusters.

Pereira et al. [19] transform the interaction graph into the corresponding line graph, in which edges represent nodes and nodes represent edges. Then they apply the graph clustering algorithm TribeMCL of [10] to group the interaction network corresponding to the line graph, and transform back the obtained clusters. The approach of clustering

the line graph produces an overlapping graph partitioning of the original protein-protein interaction graph, thus allowing proteins to be present in multiple functional modules.

In [1] CFinder, a program for detecting and visualizing densely interconnected and overlapped groups of nodes, is presented. CFinder uses the Clique Percolation Method [9] to find k-clique percolation clusters, i.e. groups of nodes that can be reached via chains of k-cliques and the link in these cliques. The parameter k has to be provided in input. Approaches such as [1] may be viewed as general approaches to study the structure of networks, suitably represented as graphs (e.g., genetic or social networks and microarray data), rather than a specialized technique to cluster PPI networks.

In [8] Cho et al. propose a flow-based modularization approach to identify overlapping functional modules in a PPI network. The modularization process consists of three phases: informative protein selection, flow simulation to detect preliminary modules and a post-process to merge similar preliminary modules. Differently from such an approach, MF-*PINCoC* does not need any post-processing step to produce the final overlapping clusterings.

Ucar et al. [24] propose an approach to reduce the scale-free topology of PPI networks by duplicating the hub nodes. After this refinement, the resulting graph is clustered by using three known graph partitioning methods. Because of the duplication process, hub proteins can be placed in multiple groups. Of course this multiple participation, differently by our approach, is not possible for the other proteins.

A different method, based on an ensemble framework, is described in [3]. The authors use three traditional graph partitioning algorithms with two metrics to obtain six basis clusterings. Then apply different consensus methods to decide each protein to which cluster should belong. A soft consensus clustering variant has also been developed to allow proteins having high propensity towards multiple memberships, to be assigned to different clusters. Though amenability to multiple membership is computed for all the nodes, the authors note that hub proteins have the highest probability to participate in more than one cluster. Both these last two methods need as input parameter the number of clusters to find. Our approach, on the contrary, searches for all the possible clusters it can find in the network.

In the next section we report the results obtained by our approach and compare them with those obtained by MCODE and RNSC, two of the most known methods in the literature [6]. Such a comparison further confirms the importance of allowing for multiple-cluster participation; in fact, constraining each protein to belong to only one module causes clusterings that are often less significant. Moreover, a discussion regarding the participation of proteins to multiple clusters, with respect to the hub proteins identified in [24], will be reported in the last sub-section.

4 Experimental Validation

In this section we present the results obtained by running MF-*PINCoC* on the PPI network of budding yeast *Saccaromyces cerevisiae*. The data set has been extracted from the *DIP* database [21] (*http://dip.doe-mbi.ucla.edu/*). At the time of download (May 2007) it consisted of 5,027 proteins and 22,223 interactions.

4.1 Validation Metrics

Before presenting the experiments, we describe the validation metrics used to asses the quality of the results. We used two metrics, a topological measure (*clustering coefficient*) and a domain based measure (*p-value*).

Clustering Coefficient: the concept of clustering coefficient has been defined by Watt in [26] and takes into account only the nodes of a network and how they are linked together. Given a node i, let n_i be the number of links connecting the k_i neighbors of i to each other. The clustering coefficient of i is defined as $C_i = 2n_i/k_i(k_i-1)$. Note that n_i represents the number of triangles passing through i, and $k_i(k_i-1)/2$ the number of possible triangles that could pass through node i. The clustering coefficient C_{B_j} of a cluster B_j is the average of the clustering coefficients of the proteins belonging to B. Analogously, the clustering coefficient C_B of a clustering $B = \{B_1, \ldots, B_h\}$ is $C_B = \sum C_{B_j}/h$.

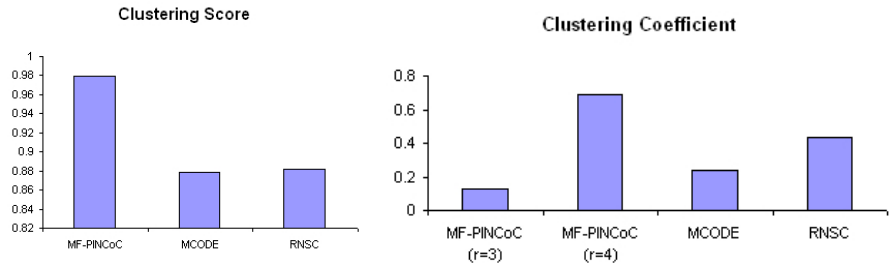
p-value: in the PPI networks it is important to verify if the clusters obtained correspond to a function meaningful from a biological point of view. This validation can be done by using the known biological associations from the *Gene Ontology Consortium Online DataBase* [11] . The Gene Ontology database provides three vocabularies of known associations: Molecular Function, Cellular Component, and Biological Process. We used the process vocabulary for validation by querying the GO Term-Finder tool (*http://db.yeastgenome.org/cgi-bin/GO/goTermFinder*) and the p-values returned to obtain a statistical and biological meaningfulness of a group of proteins. 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:

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

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 cut-off value to distinguish significant from insignificant groups. If a cluster has a p-value below the cut-off, it is considered insignificant. In our experiments we used a cut-off of 0.05. As observed in [24], it is interesting to have a global measure of an obtained clustering, instead of the p-value of a single group. The p-value score of a clustering is then defined as

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

where $min(p_i)$ is the smallest p-value of the partition i, n_S is the number of significant partitions, and n_I is the number of insignificant partitions.



(a) (b)

Fig. 1. Comparison among the three methods, showing:(a) Clustering Score; (b) Clustering Coefficient.

4.2 Comparison of MF-PINCoC, MCODE, and RNSC

In this section we compare the results obtained by running MF-PINCoC, MCODE, and RNSC on the S. Cerevisiae network. In particular, such a comparison has been carried out not only to investigate the ability of our method to discover significant functional modules w.r.t. other well consolidated techniques, but also to analyze how allowing proteins to participate in different clusterings may be useful to obtain more significant groups.

MF-*PINCoC* needs as input parameters the probability p of a REMOVE-MIN move, the number of maximum moves allowed, and the order r of the *quality* function. We set the former to 0.1, the second to 1,000, and the latter to 3. It is worth to note that *(i)* a low value of probability p is preferable to avoid the disruption of the greedy steps; *(ii)* the number of maximum flips has never been reached, in fact on average not more than 50 flips were executed before reaching a local optimum; *(iii)* the order value used is a compromise between the compactness of clusters and their size. As regards MCODE and RNSC, we run the two methods with the default parameters set by the authors. MF-*PINCoC* returned 6,108 clusters, 5,189 were couples of proteins, 145 cliques constituted by triples, 588 of size between 4 and 6, the remaining 186 with a number of proteins between 7 and 40. MCODE obtained only 57 clusters, 17 of which were triples. The cluster size is between 3 and 59. The clusters covered only 789 proteins out of the 5,027 present. RNSC obtained 2,524 clusters, 1,017 were singletons, 972 couples of proteins, 375 triples, 134 clusters of size between 4 and 7, and the remaining 26 of size between 8 and 21. Because of the different number of clusters obtained, we chose 50 random clusters returned by each method with maximum size and queried the GO Term-Finder tool. MF-*PINCoC* gave back one insignificant cluster, while MCODE and RNSC gave 6 and 5 insignificant groups, respectively.

Figure 1 graphically illustrates the behavior of MF-PINCoC, MCODE and RNSC in terms of both domain-based and topological measures. In particular, Figure 1 (a) shows the clustering scores, computed on the 50 chosen clusters, for the three methods. The figure points out that the clustering score of MF-PINCoC (0.980) is greater than those of the other two methods, which is 0.879 for MCODE and 0.882 for RNSC respectively.

This means that the biological meaning of the clusters obtained by MF-PINCoC is, on average, better than the clusters generated by the other two methods. Figure 1 (b) shows the clustering coefficients computed on all the obtained clusters. The clustering coefficient of MF-PINCoC has been computed for two different values of the parameter r ($r = 3, 4$). As already observed in section 2, higher values of r bias our method towards denser but smaller clusters. In fact, for $r = 4$ we obtained 6,322 clusters, 5,332 were couples, 138 cliques constituted by triples, 724 of size between 4 and 6, the remaining 128 of size between 7 and 33. Thus, with respect to the previous experiment, with $r = 3$, clusters have a lower number of proteins. However, the clustering coefficient is 0.69 with $r = 4$ and 0.13 with $r = 3$. On the other hand, MCODE scored a clustering coefficient 0.23, and RNSC 0.43. This points out that, in order to obtain a better value also in terms of topological connectivity, the input parameter r has to be properly tuned.

	Cluster	p-value	Associated process
MF-PINCoC	PFS2,PTI1,MPE1,REF2,YTH1,FIP1,CFT1, CFT2,PTA1,YSH1,HCA4,PAP1,RNA14,GLC7	2.17E-26	
MCODE	CFT1,CFT2,FIP1,GLC7,MPE1,PAP1,PFS2,PTA1,PTI1, MPE1,PAP1,PFS2,PTA1,PTI1,REF2,RNA14,YSH1,YTH1	4.67E-27	mRNA Polyadenylation
RNSC	REF2,PCF11,GLC7,RNA14,YTH1,FIP1,PAP1,CFT1, CFT2,PTA1,YSH1,PTI1,PFS2,MPE1,HCA4,SSU72	1.08E-28	
MF-PINCoC	NUP84,NUP60,CRM1,PAB1,MSN5,NUP57,NUP42,NUP49, GSP1,NUP145,SRP1,NUP2,NUP100,KAP123,KAP95,PSE1,NUP116	6.61E-26	Nuclear Transport
MCODE	MSN5, NTF2, NIC96, NUP145, NSP1, GSP1	1.66E-09	
MF-PINCoC	HAS1,MAK21,CIC1,SDA1,NOP6,NUG1,NOP7,CKA1,NOP2,SSF1, NOP4,BUD20,RPF2,YTM1,RLP7,NOP15,MAK5,NSA2,ERB1,TIF6,NOG1	1.68E-22	Ribosome Biogenesis and Assembly
RNSC	URB1, NOP4, MAK21, HAS1, NOC2, BRX1, CIC1, NOP12, PUF6, DBP10, NOP2, SSF1, RPF2, DRS1, MAK5	2.90E-15	
MF-PINCoC	NUP84,CRM1,NUP120,MSN5,NUP42,NUP145,NUP57, NUP49,SRP1,NUP2,NUP100,KAP95,PSE1,NUP116	2.03E-23	Nuclear mRNA splicing, via spliceosome
MCODE	SPP381, MSL1, LEA1, SMX3	1.19E-06	
RNSC	CDC6, ORC1, ORC2, ORC3, ORC4, ORC5, ORC6	6.16E-16	

Table 1. Some significant clusters obtained by the three methods MF-PINCoC, MCODE and RNSC.

Table 1 shows some of the clusters obtained by the three methods, for which the GO validation returned the same associated process. The table points out the good capability of MF-*PINCoC* to isolate functional modules.

4.3 Multi-functional proteins

We now show, with some examples, how MF-*PINCoC* is able to cluster multi-facets proteins into different functional modules, each characterized by a particular function. In table 2 we report the protein name, the number of proteins with which it is connected (denoted degree), the list of proteins participating to the same cluster, and the associated biological process. We consider three proteins KAP95, LSM8, and CKA1 that have

been discussed by Ucar et al. in [24], and compare their results with ours. As reported in [24], KAP95 is an essential protein known to take part in *nucleocytoplasmatic transport*. MF-*PINCoC* groups KAP95 with other 5 proteins (NTF2, GSP1, PSE1, SRP1, NUP1) participating to this same biological process. Ucar et al. point out that one the partitions they found (NTF2, SSA1, YRB1, RNA1, GSP1, SRM1, MTR10, KAP122, KAP142, KAP124, NUP1, NUP2, NUP42, NUP60, NUP82, NUP145, NUP157, NUP-170) contained 8 NUPs proteins and 3 KAPs proteins, known as nucleoporins and karyorephins respectively, with p-value 1.07E-27. We obtained an analogous result, in two different clusters. The former contains 9 NUPs proteins (NUP2, NUP84, NUP60, NUP57, NUP42, NUP49, NUP145, NUP100, NUP116) and two KAPs proteins (KAP-95 and KAP123), sharing the *Nuclear Transport* biological process with p-value 6.61 E-26, the second one contains 4 NUPs proteins (NUP116, NUP57, NUP60, NUP100, NUP145) and 3 KAPs proteins (KAP95, KAP104, KAP123), sharing the *cellular localization* process, with p-value 2.06E-09.

The hub protein LSM8 has been found by Ucar et al. with other 10 proteins (LSM2, LSM3, LSM5, PRP3, PRP4, PRP6, PRP21, PRP31, SMB1, SPP381) with biological process *mRNA splicing* and p-value 1.2E-12. We found the same protein in several groups, in particular, as reported in the table 2, LSM8, for this same process, has been grouped with 12 proteins (LSM3, PRP3, PRP4, PRP6, PRP8, PRP31, SMB1, SPP381, SMD3, SMX2, SNU114, SNU66) having p-value 1.46E-23. The two sets of proteins are almost the same, the difference is that the cluster found by MF-*PINCoC* does not contain LSM2, but has four new proteins, SMD3, SMX2, SNU114, SNU66, and a p-value much higher, thus a better biological meaning. However, LSM8 has been grouped with other proteins forming other functional modules, like reported in the table. For example, it is clustered with 7 proteins of the LSM family, which are known to interact each other in the *mRNA metabolic* process, with a very low p-value (3.02E-22).

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 2.3e-19, the second one with *protein amino acid phosphorylation* and p-value 1.2E-05, the third group is annotated with *organelle organization and biogenesis* and p-value 3.2E-12. MF-*PINCoC* found, among the others, a group with p-value 1.68E-22 and annotation *ribosome biogenesis and assembly*, another one with p-value 9.96E-07 and process *cellular component organization and biogenesis*, the third one with p-value 1.03E-07 and biological process *transcription, DNA-dependent*. Finally, figure 2 draws three clusters of proteins in which CKA1 and LSM8 are involved. In particular, figures 2(a) and 2(b) show the first and third clusters reported in table 2 relative to the CKA1 protein. Figure 2(c) displays the second cluster of table 2 relative to the LSM8 protein. The graphs have been drawn by using the PIVOT software [17]. These results point out that the strategy of allowing proteins to belong to different clusters seems to be effective in grouping multi-functional proteins into multiple functional groups, to individuate biologically significant modules, each corresponding to a different function in which these proteins are involved.

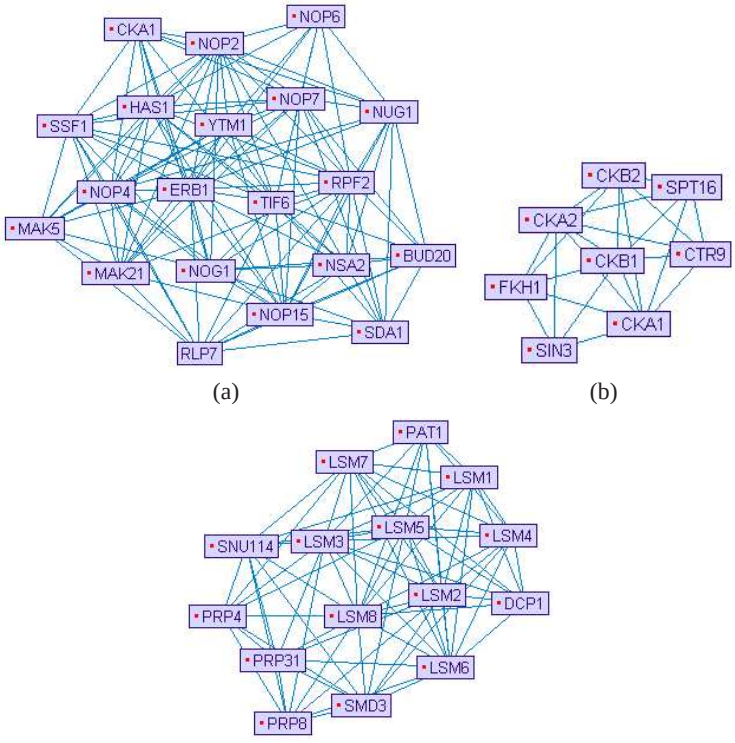


Fig. 2. PPI networks of clusters obtained showing:(a) first cluster reported in table 2 relative to the CKA1 protein; (b) third cluster reported in table 2 relative to the CKA1 protein; (c) second cluster of table 2 relative to the LSM8 protein.

Hub-Protein	degree	Clusters	p-value	Associated process
KAP95	58	KAP95,KAP123,NUP2,NUP84,NUP60, NUP42,NUP49,NUP145,NUP100,NUP116, SRP1,CRM1,PAB1,PSE1,GSP1	6.61E-26	Nuclear Transport
		MSN5,NUP57,KAP95,KAP104,KAP123, NUP116, NUP57,NUP60,NUP100,GSP1,SRP1,PSE1	2.06E-09	Cellular localization
		KAP95, NTF2,GSP1,PSE1,SRP1,NUP1	1.71E-09	Nucleocytoplasmatic transport
LSM8	71	LSM3, LSM8, PRP3, PRP31, PRP4, PRP6, PRP8, SMB1, SMD3, SMX2, SNU114, SNU66, SPP381	1.46E-23	Nuclear mRNA splicing, via spliceosome
		LSM8, LSM1, LSM2, LSM3, LSM4, LSM5, LSM6, LSM7, DCP1,PAT1,PRP31, PRP4, PRP8, SMD3, SNU114	3.02E-22	mRNA metabolic process
		LSM1, LSM8,LSM2,EDC3,KEM1,DCP2, LSM4	1.44E-06	Biopolymer catabolic process
CKA1	66	HAS1,MAK21,CIC1,SDA1,NOP6,NUG1,NOP7,CKA1, NOP2,SSF1,NOP4,BUD20,RPF2,YTM1, RLP7,NOP15,MAK5,NSA2,ERB1,TIF6,NOG1	1.68E-22	Ribosome biogenesis and assembly
		RPF2,YTM1,NOG1,ERB1,MAK5,HAS1,TIF6,CKA1, MAK21,NOP2,NOP4,NOP6,NOP7,NOP15,CIC1,SSF1	9.96E-07	Cellular component organization and biogenesis
		CKA1,CKB1,CKA2,CKB2,SPT16,CTR9,SIN3,FKH1	1.03E-07	Transcription, DNA-dependent

Table 2. Some examples of hub proteins and the clusters they participate.

5 Conclusions

We proposed the algorithm MF*-PINCoC*, an extension of the algorithm *PINCoC*, aiming at individuating clusters of multi-facets proteins in PPI networks. One of the main feature of the method consists in allowing proteins to be placed in multiple clusters. This is a distinguished advantage since it enables a more accurate representation of the complexity of biological systems and the detection of different functional modules in which proteins are involved. As proved by tests carried out on the *Saccaromyces cerevisiae* proteins data set, the presented method returns partitions that are biologically relevant, correctly clustering proteins which are known to participate in different biological processes. A comparison with other existing approaches shows that MF*-PINCoC* is competitive with respect to these methods according to validation techniques commonly adopted in the literature.

References

1. B. Adamcsek, G. Palla, I. J. Farkas, I. Dernyi, and T. Vicsek. Cfinder: locating cliques and overlapping modules in biological networks. *Bioinformatics*, 22(8):1021–1023, 2006.
2. V. Arnau, S. Mars, and I. Marìn. Iterative cluster analysis of protein interaction data. *Bioinformatics*, 21(3):364–378, 2004.
3. S. Asur, D. Ucar, and S. Parthasarathy. An ensemble framework for clustering protein-protein interaction networks. *Bioinformatics*, 23:i29–i40, 2007.
4. G. Bader and H. Hogue. An automated method for finding molecular complexes in large protein-protein interaction networks. *BMC Bioinformatics*, 4(2), 2003.
5. M. Blatt, S. Wiseman, and E. Domany. Superparamagnetic clustering of data. *Phisical Review Letters*, 76(18):3251–3254, 1996.

6. S. Brohèe and J. van Helden. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics*, 7:488, 2006.
7. Y. Cheng and G. M. Church. Biclustering of expression data. In *Proceedings of the 8th International Conference On Intelligent Systems for Molecular Biology (ISMB'00)*, pages 93–103, 2000.
8. Y.-R. Cho, W. Hwang, M. Ramanathan, and A. Zhang. Semantic integration to identify overlapping functional modules in protein interaction networks. *BMC Bioinformatics*, 8:265, 2007.
9. I. Derenyi el al. Clique percolation in random networks. *Physical Review Letters*, 94:160202, 2005.
10. A.J. Enright, S.V. Dongen, and C.A. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleid Acids Res*, 30(7):1575–84, 2002.
11. S. Asburner et al. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature Genetics*, 25:25–29, 2000.
12. E. Hartuv and R. Shamir. Clustering algorithm based graph connectivity. *Information Processing Letters*, 76:175–181, 2000.
13. H. Jeong, AL. Barabasi, and ZN Oltvai. Lethality and centrality in protein networks. *Nature*, 411:41–42, 2001.
14. A. D. King, Natasa Przulj, , and Igor Jurisica. Protein complex prediction via cost-based clustering. *Bioinformatics*, 20(17):3013–3020, 2004.
15. C. Lin, Y. Cho, W. Hwang, P. Pei, and A. Zhang. Clustering methods in protein-protein interaction network. *in Knowledge Discovery in Bioinformatics: Techniques, Methods and Application, John Wiley & Sons,Inc*, 2006.
16. S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE Transactions on Computational Biology and Bioinformatics*, 1(1):24–45, 2004.
17. N. Orlev, R. Shamir, and Y. Shiloh. Pivot: Protein interaction visualization tool. *Bioinformatics*, 20(3):424–425, 2004.
18. P. Pei and A. Zhang. A two-step approach for clustering proteins based on protein interaction profiles. In *IEEE Int. Symposium on Bioinformatics and Bioengeneering (BIBE'2005)*, pages 201–209, 2005.
19. J. B. Pereira, A.J. Enright, and C.A. Ouzounis. Detection of functional modules from protein interaction networks. *Proteins: Structure, Fuctions, and Bioinformatics*, (20):49–57, 2004.
20. C. Pizzuti and S. Rombo. Pincoc: a co-clustering based approach to analyze protein-protein interaction networks. In *Proceedings of the 8th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'07)*, 2007.
21. L. Salwinski, C.S. Miller, A.J. Smith, F.K. Pettit, J.U. Bowie, and D. Eisenberg. The database of interacting proteins: 2004 update. *Nucleic Acids Research*, 32(Database issue):D449–D451, 2004.
22. M.P. Samantha and S. Liang. Redundancies in large-scale protein interaction networks. In *Proceedings of the National Academy of Science, USA, 100*, pages 12579–12583, 2003.
23. V. Spirin and L.A. Mirny. Protein complexes and and functional modules in molecular networks. In *Proceedings of the National Academy of Science, USA, 100*, pages 12123–12128, 2003.
24. D. Ucar, S. Asur, Ü.V. Çatalyürek, and S. Parthasarathy. Improving functional modularity in protein-protein interactions graphs using hub-induced subgraphs. In *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 371–382, 2006.
25. D. von Mering, C. Krause, and *et al*. Comparative assessment of a large-scale data sets of protein-protein interactions. *Nature*, 31:399–403, 2002.
26. D. J. Watt. *Small worlds*. Princeton University Press, 1999.