PINCoC: a Co-Clustering based Approach to Analyze Protein-Protein Interaction 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. A novel technique to search for functional modules in a protein-protein interaction network is presented. The network is represented by the adjacency matrix associated with the undirected graph modelling it. The algorithm introduces the concept of *quality* of a sub-matrix of the adjacency matrix, and applies a greedy search technique for finding local optimal solutions made of dense sub-matrices containing the maximum number of ones. An initial random solution, constituted by a single protein, is evolved to search for a locally optimal solution by adding/removing connected proteins that best contribute to improve the *quality* function. Experimental evaluations carried out on *Saccaromyces Cerevisiae* proteins show that the algorithm is able to efficiently isolate groups of biologically meaningful proteins corresponding to the most compact sets of interactions.

1 Introduction

One of the most important challenges of the post-genomic era is the analysis of the complex biological processes in which proteins are involved. Recently, great attention has been addressed to the whole set of protein interactions of a given organism, known as *interactome* or *protein-protein interaction (PPI) network*. Many studies have been driven to predict and understand functional properties of proteins starting from interactomes (e.g., [6, 5, 12]). In the last few years, a vast amount of new protein interactions have been discovered and made available. This has spurred the search for automated and accurate tools to identify significant parts of this data.

PPI networks are often modelled as graphs where nodes represent proteins and edges represent pairwise interactions. Many current efforts aim at clustering dense regions of a given PPI network, since it has been observed by biologists that groups of highly interacting proteins could be involved in common biological processes. A number of approaches have been proposed to extract relevant modules from PPI networks [4, 5, 1, 13, 12]; some of them rely on traditional hierarchical clustering methods [7], other ones are based on graph partitioning algorithms [3, 11, 8]. The obtained results have been found to strongly depend on the adopted approach, and on the input parameters fixed by the user. Most methods, in fact, require the number of clusters to be known in advance. However, this information is not always available, thus some algorithms are executed with different cluster numbers and results satisfying a quality criteria are considered to be the most reliable. Obviously, the necessity of running an algorithm different times may cause losses in efficiency. Another problem that arises in PPI networks is the choice of the metric adopted to measure the distance between two proteins. In this kind of graphs, due to the structure of the interactions, it has been found that the distances among many nodes are often identical. In such a case the adopted clustering method fails in finding good solutions, due to the presence of ties that have to be solved arbitrarily.

In this paper, we present a novel technique, based on a co-clustering approach [9], to search for functional modules in protein-protein interaction networks. Co-clustering methods, differently from clustering approaches, aim at simultaneously grouping both the dimensions of a data set. 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. The *PPI network Co-Clustering* based algorithm, named *PINCoC*, applies a greedy search technique for finding local optimal solutions made of dense sub-matrices containing the maximum number of ones. The notion of *quality* of a sub-matrix is introduced. High quality sub-matrices should correspond to modules of the input interactome having a significant biological function. The algorithm starts with an initial random solution constituted by a single protein and searches for a locally optimal solution by adding/removing connected proteins that best contribute to improve the *quality* function. In order to escape poor local maxima, with a fixed probability, the protein causing the minimal decrease of the *quality* function is removed. When the algorithm cannot improve any more the solution found so far, the computed cluster is returned. To limit the effects of the initial random choice of a protein to build a cluster, one step of backtracking is executed. Each protein belonging to the solution is at turn temporary removed, and eventually substituted with a new one that best improves the *quality* function. At this point a new random protein is chosen, and the process is repeated until all the proteins are assigned to any group. In the hard scenario of strongly connected networks, where the detection of the most functionally related proteins is a difficult task due to the high number of connections, our algorithm is able to efficiently isolate those groups of proteins corresponding to the most compact sets of interactions. In the experimental result section we validate the clusters found by *PINCoC* through the *SGD Gene Ontology Term Finder* and compare our results with other studies made in the literature [1, 8]. We show that the obtained clusters are recognized to be biologically meaningful.

The paper is organized as follows. The next section defines the problem of clustering PPI networks and the adopted notation. Section 3 describes the proposed algorithm. Section 4 illustrates the experiments we carried out on a set of S. Cerevisiae proteins and compare the obtained results with those of [1, 8]. Finally, in Section 5 we draw our conclusions.

2 Notation and Problem definition

In this section the notation used in the paper is introduced and the formalization of the problem of clustering PPI networks as a co-clustering problem is provided.

A PPI network $\mathcal{P}$ can be 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 from node i to node j, 0 otherwise. Since the graph G is undirected, the adjacency matrix is symmetric. Note that the mathematical definition of adjacency matrix assumes that the main diagonal contains a 1 value at position (i, i) only if there is a loop at vertex i. In the biological context a protein connected with itself is not meaningful. However, by convention, we assume that the main diagonal of the adjacency matrix A of a PPI network contains all ones. This means that if a row of A is constituted by all zeroes except one position i with value 1, the protein corresponding to node i does not interact with any other protein. The problem of finding dense regions of a PPI network $\mathcal{P}$ can thus 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 a matrix A can be viewed as a special case of co-clustering a binary data matrix where the set of rows and 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 [9], also known as bi-clustering, differently from clustering, tries to simultaneously group both the dimensions of a data set. For example, when clustering genes with respect to a set of experimental conditions, not all the genes are relevant for all the experimental conditions, but groups of genes are often co-regulated and co-expressed only under specific conditions. In this application domains the idea of co-clustering both the dimensions turns to be more beneficial and interesting than clustering with only one dimension. Let A be an $N \times M$ data matrix of binary values. Let $X = \{I_1, \ldots, I_N\}$ denote the set of rows of A and $Y = \{J_1, \ldots, J_M\}$ the set of columns of A.

Definition 1. *A co-cluster is a sub-matrix $B = (I, J)$ of A, where I is a subset of the rows X of A, and J is a subset of the columns Y of A.*

The problem of co-clustering can then 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 must 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 1 values as possible. Higher the number of ones, more likely those proteins are to be functionally related. In the following we introduce a *quality* function that tries to obtain both these objectives. Note that the adjacency matrix A associated with a PPI network is a squared matrix of dimension $N \times N$, where N is the number of proteins. This means that any co-cluster $B = (I, J)$ of A has the property that the set I of rows and the set J of columns coincide. In particular, being A symmetric, any co-cluster found is symmetric too.

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 $B = (I, J)$. 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 $v_B = \sum_{i \in I, j \in J} a_{ij}$.

Definition 2. *Given a co-cluster $B = (I, J)$, let a_{iJ} be the mean of the ith row of B, and let a_{Ij} be the mean of the jth column of B. The power mean of B of order r, denoted as $\mathbf{M}_r(B)$ is defined as*

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

Since B is symmetric, $|I| = |J|$ and a_{iJ} = a_{Ij}, thus the power mean can be reduced to

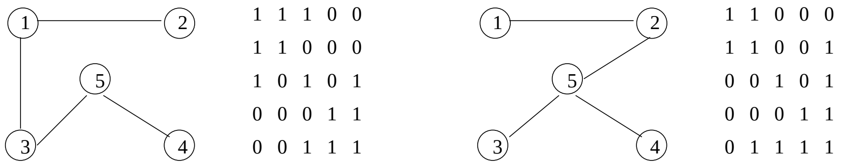
$$\mathbf{M}_r(B) \ = \frac{\sum_{i \in I}(a_{iJ})^r}{|I|}$$

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

Definition 3. *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 $Q(B) = \mathbf{M}_r(B) \times v_B$.*

i.e. the quality of a co-cluster B is the product between the power mean of B of order r, and the number of non-zero entries in B. The quality $Q(B)$ of the co-cluster $B = (I, J)$ is equal to $|I| \cdot |I|$ only when each entry of B is one, thus $Q(B)$ is upper bounded by its volume, i.e. $Q(B) \leq v_B \leq |I| \cdot |I|$. When B contains zero entries, the *quality* is a fraction of v_B. Notice that, adding a row/column composed only by ones or removing a row/column composed only by zeros, always improves the *quality* of the co-cluster.

When $r = 1$ the power mean coincides with the standard mean. However, the mean of a binary matrix of fixed volume (i.e., having the same number of ones), assumes always the same value independently where the 1/0 values are positioned. This means that it is not able to distinguish matrices corresponding to PPI networks having the same total number of interactions but different structure. Consider for example the two sub-matrices and the associated protein graphs showed in Figure 1. The total number of ones, i.e., of interactions, is equal to 13 in both cases, but the way the proteins interact is different. Intuitively, the graph in Figure 1(b) represents a more compact set of interactions than the one in Figure 1(a). If we compute the power mean of order 1 the value is 0.260 for both of the illustrated matrices, whereas the power mean of order 2 is 0.140 for the matrix on the left and 0.148 for the matrix on the right. Since the volume (the number of ones) is 13 for both matrices, the *quality* function in the former case is 3.38 for both the matrices, while for $r = 2$ it is 1.820 for the first matrix and 1.924 for the second one. Thus, $r = 2$ is more suited to characterize different ways in which proteins interact. However, it is worth to point out that increasing the value of r biases the *quality* function towards matrices containing a low number of zeroes but of lower volume. Thus the choice of r should be done by considering the density of the adjacency matrix. In the next section the *PPI network Co-Clustering* based algorithm *PINCoC*, is presented. The method uses the concept of *quality* to find maximally dense regions in the binary data adjacency matrix.



$$\begin{matrix}1&1&1&0&0\\1&1&0&0&0\\1&0&1&0&1\\0&0&0&1&1\\0&0&1&1&1\end{matrix}$$

(a)

$$\begin{matrix}1&1&0&0&0\\1&1&0&0&1\\0&0&1&0&1\\0&0&0&1&1\\0&1&1&1&1\end{matrix}$$

(b)

Fig. 1. Matrices with equal mean value but different network structure.

3 Algorithm Description

In this section we present *PINCoC*, an algorithm for clustering a PPI network $\mathcal{P}$ represented through the adjacency matrix A of the graph associated with $\mathcal{P}$. Let $A = (X, Y)$ be the $N \times N$ adjacency matrix where $X = \{I_1, \ldots, I_N\}$ denote the set of rows of A and $Y = \{J_1, \ldots, J_N\}$ the set of columns of A. Each row/column of A corresponds to a protein, thus in the following we use the two terms as synonyms.

A co-cluster $B = (I, J)$ can be encoded as a binary string b of length N, where N is the number of rows/columns of the adjacency matrix. If the value of the i-th bit is set to 1 it means that the corresponding i-th protein belongs to the co-cluster.

```
Algorithm PINCoC
Input:
- a 0-1 adjacency matrix $A = (X, Y)$ of dimension $N \times N$
- maximum number of iterations allowed $max\_flips$
- probability $p$ of a REMOVE-MIN move ($0 =$ no remove move)
Ouput: a clustering $B = \{B_1, \ldots, B_k\}$ of the PPI network corresponding to $A$
set $B = \emptyset$, $A_0(X_0, Y_0) = A(X, Y)$, i = 0
while $A_i$ is not the empty matrix
choose at random a row $k$ of $A_i$, set $I_i = \{k\}$, $J_i = \{k\}$, $B_i = (I_i, J_i)$
set $f = 0, quality = -\infty, local\_maximum = false$
    while $f < max\_flips$ and not $local\_maximum$
        let $0 \leq \overline{p} \leq 1$ a random generated number
        if $\overline{p} > p$ then
            let $\overline{B_i} = (\overline{I_i}, \overline{J_i})$ the co-cluster obtained from $B_i$
            after the move that maximize the quality
            if $Q(\overline{B_i}) > Q(B_i)$ then
                accept the move, $I_i = \overline{I_i}$, $J_i = \overline{J_i}$, and update $Q(B_i)$
            else
                set $local\_maximum = true$
        else
            remove the row/column of $B_i$ scoring the minimum decrease of the quality function
        $f = f + 1$
    end while
    execute one-pass backtracking
    $B = B \cup B_i$
    set $X_{i+1} = X_i - I_i, Y_{i+1} = Y_i - J_i, A_{i+1} = (X_{i+1}, Y_{i+1})$
end while
    assign the singletons
    return the clustering $B = \{B_1, \ldots, B_k\}$
```

Fig. 2. The *PINCoC* algorithm.

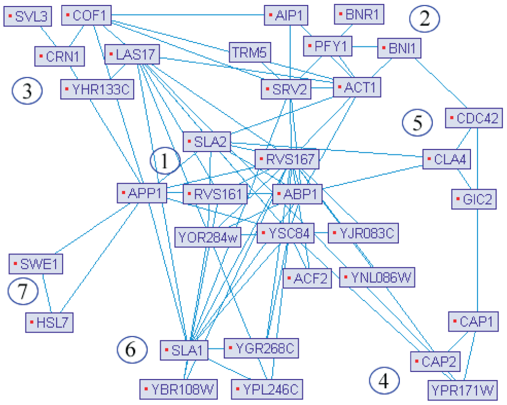
The algorithm, showed in figure 2, receives in input a 0-1 adjacency matrix, the maximum number of times (max_flips) that a flip can be done, and the probability (p) of executing a REMOVE-MIN move (these two latter input parameters are explained shortly). *PINCoC* starts with an initial random co-cluster $B = (I_i, J_i)$ constituted by a single row and a single column such that $I = \{k\}$ and $J = \{k\}$, where $1 \leq k \leq N$ is a random row/column index. Then it evolves the initial co-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 co-cluster B_i, is added to B_i. Viceversa, if a bit is set from 1 to 0 it means that the corresponding protein is removed from the co-cluster. During its execution, in order to avoid get 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 (max_flips) is reached, or the solution cannot ulteriorly be improved (get trapped into a local maximum). Until the stop condition is not reached, it executes a REMOVE-MIN move with probability p, and a greedy move with probability $(1 - p)$. When the inner loop stops, the co-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 co-cluster is generated, and the process is repeated until all the rows/columns have been assigned. Some of the clusters obtained at the end of the algorithm could be constituted by a single protein because all its neighboring nodes have already been assigned to a group. This situation happens for those proteins that have few interactions and thus they have not been assigned to any group because their contribution was considered marginal. However, we chose to handle such singletons by adopting the following strategy. Let h be a singleton protein, $n_1, \ldots, n_h$ its neighboring proteins, i.e. the proteins having a direct interaction with h, and $B_{n_1}, \ldots B_{n_h}$ their corresponding clusters (note that the B_{n_i} are not necessarily distinct). Then h is assigned to the cluster B_{n_i} s.t. $Q(B_{n_i} \cup \{h\})$ is maximum, i.e. whose *quality* function has the better improvement or the lowest decrease. In the experimental results section we show that *PINCoC* is able to generate clusters of proteins both dense and biologically meaningful. The temporal cost of the algorithm to compute a single cluster $B_i = (I_i, J_i)$ is upper bounded by

$$max\_flips \times C_q \times [(1-p) \times N + pN] + C_q \times \mid I_i \mid \times N = C_q \times N \times (max\_flips + \mid I_i \mid)$$

where C_q is the cost of computing the *quality* of the co-cluster after performing a move. In order to reduce the complexity of C_q, we maintain, together with the current co-cluster $B_i = (I_i, J_i)$, the mean values a_{iJ}, for each $i \in I$, and the volume v_{IJ}. Thus, computing the $|I_i|$ mean values a_{iJ} $(1 \leq i \leq |I|)$ after performing a move can be done efficiently in time $|I_i|$, i.e. in time linear in the co-cluster dimensions, by exploiting the values maintained together with the current co-cluster.

Clusters	Gene Ontology term	p-value
1	Actyn cytoskeleton organization and biogenesis	$2.25 \cdot 10^{-10}$
2	Rensponse to osmotic stress	$9.63 \cdot 10^{-07}$
3	Actin filament organization	$3.1 \cdot 10^{-04}$
4	Actin cytoskeleton organization and biogenesis	$4.63 \cdot 10^{-06}$
5	Rho protein signal transduction	$1.11 \cdot 10^{-06}$
6	Actin cortical patch assembly	$4.4 \cdot 10^{-04}$
7	G2/M transition of mytotic cell cycle	$5.3 \cdot 10^{-04}$

(a)



Clusters	Proteins
1	sla2 abp1 app1 yor284w rvs161 rvs167 ynl086w yjr083c ysc84 acf2
2	bnr1 pfy1 trm5 bni1 aip1 srv2 act1
3	svl3 crn1 cof1 las17 yhr133c
4	cap1 cap2 ypr171w
5	cdc42 cla4 gic2
6	sla1 ybr108w ypl246c ygr268c
7	swe1 hsl7

(b)

Fig. 3. (a) Clusters validation by Gene Ontology term finder; (b) graphical view of the obtained clusters drawn using PIVOT [10].

4 Experimental Validation

In this section we apply *PINCoC* on a set of 34 proteins coming from the well known *S. cerevisiae* network. This set, extracted from the *DIP* database (*http://dip.doe-mbi .ucla.edu/*) has already been well studied and characterized in the literature [6, 1]. In the following, we first present the clusters obtained by our method and we validate their biological meaningfulness by using the *SGD Gene Ontology Term Finder* (*http://db .yeastgenome.org/cgi-bin/GO/goTermFinder*). Then we compare our results with those obtained by Arnau and Marìn [1], and King et al. [8] showing that the clustering returned by our method is meaningful and comparable with the other two approaches.

The *PINCoC* algorithm has been implemented in C++, and all the experiments have been performed on a Pentium 4 machine, 1800MHz, 1GB RAM, by using $r = 2$, $max\_flips = 100$, $p = 0.1$. The results obtained are summarized in figure 3. In particular, figure 3(a) shows the table containing the seven clusters returned, the GO term obtained when querying the *SGD Gene Ontology Term Finder* with the proteins belonging to our clusters, and the corresponding p-value. The p-value is a commonly used measure of the statistical and biological significance of a cluster. It gives the probability that a given set of proteins occurs by chance. In particular, given a cluster of size n and 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 [2]. Thus, the closer the p-value to zero, the more significant the associated GO term. In the table we show the smallest p-value found over all the functional groups. We can observe that the p-value of our clusters varies between $2.25 \cdot 10^{-10}$ and $5.3 \cdot 10^{-04}$, values all sufficiently low to consider relevant the biological meaningfulness of the correspondent clusters. Figure 3(b) shows a graphical representation of the interactions among the considered proteins, indicated by the names according to the Gene Ontology notation, and a list of the proteins participating to each cluster. The graph has been drawn using PIVOT [10]. It is worth to point out that the biological significance of the seven clusters agrees with the functional classification reported in [6]. *PINCoC*, in fact, is able to correctly distinguish proteins involved in different processes such as, for example, actin patch assembly and patch mediated endocytosis (Cluster 1), actin-capping proteins (Cluster 4), CDC42 signaling pathway (Cluster 5), control of the morphogenesis checkpoint (Cluster 7).

In order to better assess the quality of the results obtained by *PINCoC*, we now compare them with those obtained in [1] and [8]. Arnau and Marìn [1] proposed the hierarchical clustering method UVCLUSTER, that iteratively explores the distance data sets to analyze protein-protein interaction networks. UVCLUSTER uses an agglomerative hierarchical clustering twice. The first time it considers the *primary distances*, that is, the minimum number of interactions required to connect two proteins, and generates K alternative clustering solutions. The value of K must be given by the user. The second time it clusters again the set of proteins but using the *secondary distances*, defined as the percentage of clusters in which two proteins do not appear together.

The second algorithm we consider for comparison is the Restricted Neighborhood Search Clustering (RNSC), proposed by of King et al. [8]. RNSC is a cost-based local search algorithm that explores the solution space of all the possible clusterings to minimize a cost function that refelcts the number of inter-cluster and intra-cluster edges.

Table 1 reports the clusterings obtained by *PINCoC*, UVCLUSTER and RNSC with the list of proteins for each cluster, the fraction of proteins in each cluster that have been recognized to participate to a specific biological process with the p-value reported in the last column. RNSC needs some input parameters. In our experimentation we used the values reported by Brohèe and van Helden [4], who have extensively analyzed RNSC to determine the best parameter values with respect to (i) the best matching complex found in a cluster, denoted by RNSC$_a$, and (ii) how well a given cluster isolates complexes from other clusters, denoted by RNSC$_s$. Note that the p-values of the clusters reported for UVCLUSTER differ from those appearing in [1] because the authors computed the values on the January 2004 release of the DIP database, containing 4721 proteins. At present DIP contains 5027 proteins. For each cluster found by *PINCoC*, we report the cluster (or the clusters) obtained by UVCLUSTER and RNSC that has the maximum number of common proteins with *PINCoC*. The names of the common proteins with UVCLUSTER are highlighted in bold, those of the common proteins with RNSC are highlighted in italic. The symbol '–' means that no significant ontology term has been found for that cluster. The table points out that our first cluster is bigger than those generated by both UVCLUSTER and RNSC, and has a lower p-value. The second cluster found by *PINCoC* partially includes two different clusters found by UVCLUSTER, and other two different clusters found by RNSC (note that we use RNSCa, s for short when both the two RNSC runs returned the same cluster). Both the two groups generated by UVCLUSTER and those generated by RNSC have higher p-value than the *PINCoC* cluster. In correspondence of the third cluster generated by *PINCoC*, both UVCLUSTER and RNSC found two groups without any biological meaning. The fourth and fifth clusters are identical for all the methods, except than RNSC$_s$, which was able to score the best p-value for the cluster $\{$*cdc42, cla4, gic2, bni1*$\}$ thanks to the protein *bni1*, which does not appear in the correspondent cluster of the other methods. This is the only case in which *PINCoC* does not reach the best p-value score. The seventh cluster generated by *PINCoC* does not contain the protein $app1$, in fact this protein is not involved in the biological process of the other two. Finally, it worth to note that UVCLUSTER and RNSC returned the singleton clusters acf2, yjr083c, ynl086w, ypi236c, ygr268c, ybr108w, and trm5. In our approach this is not possible because of our policy of assigning singleton elements to the most suited clusters. Interestingly, *PINCoC* assigns acf2, yjr083c, and ynl086w to the first cluster, ypi236c, ygr268c, and ybr108w to the sixth cluster, and trm5 to the second one, by obtaining a better p-value. The table points out the very good results of *PINCoC*, comparable with those obtained from the other two methods.

5 Concluding Remarks

We proposed a novel technique to detect significant functional modules in a protein-protein interaction network. The main novelty of the approach is the formalization of the problem of finding dense regions of a PPI network as a co-clustering problem. The method has two fundamental advantages with respect to other approaches in the literature. The first is that the number of clusters is automatically determined by the algorithm. Furthermore, the problem of ties occurring in protein-protein distances plaguing algorithms based on hierarchical clustering is implicitly solved. As proved by tests carried out on *S. cerevisiae* proteins, the presented method returns partitions that are biologically relevant, correctly clustering proteins which are known to be involved in different biological processes. Future research aims at using *PINCoC* on sets of proteins of other organisms, to characterize proteins whose biological functions are not yet completely known.

References

1. V. Arnau, S. Mars, and I. Marìn. Iterative cluster analysis of protein interaction data. *Bioinformatics*, 21(3):364–378, 2004.
2. S. Asur, D. Ucar, and S. Parthasarathy. An ensemble framework for clustering protein-protein interaction networks. *Bioinformatics*, 23:i29–i40, 2007.
3. G. Bader and H. Hogue. An automated method for finding molecular complexes in large protein-protein interaction networks. *BMC Bioinformatics*, 5(2), 2003.
4. S. Brohèe and J. van Helden. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics*, 7, 2006.

Methods	Clusters	Proteins Fraction	p-value
PINCoC	***sla2*** ***abp1*** ***yor284w*** ***rvs167*** ***ysc84*** app1 *rvs161* ynl086w *yjr083c* acf2	7/10	$2.25 \cdot 10^{-10}$
UVCluster	**sla2** **abp1** **yor284w** **rvs167** **ysc84** sla1 ygr268c	6/7	$3.29 \cdot 10^{-09}$
RNSC$_{a,s}$	*sla2* *abp1* *yor284w* *rvs167* *ysc84* *rvs161* *yjr083c*	7/7	$7.63 \cdot 10^{-07}$
PINCoC	***bnr1*** ***bni1*** ***pfy1*** ***act1*** ***srv2*** ***aip1*** **trm5**	6/7	$9.63 \cdot 10^{-07}$
UVCluster	**bnr1** **bni1** **pfy1**	2/3	$3.99 \cdot 10^{-05}$
UVCluster	**act1** **srv2** **aip1** **trm5** cof1	2/5	$3.50 \cdot 10^{-04}$
RNSC$_a$	*bnr1* *bni1* *pfy1*	2/3	$3.99 \cdot 10^{-05}$
RNSC$_s$	*bnr1* *pfy1*	2/2	$3.70 \cdot 10^{-04}$
RNSC$_{a,s}$	*act1* *srv2* *aip1* cof1	4/4	$1.30 \cdot 10^{-04}$
PINCoC	***svl3*** ***crn1*** ***las17*** ***yhr133c*** cof1	3/5	$3.1 \cdot 10^{-04}$
UVCluster	ypl246c **las17** **yhr133c**	–	–
UVCluster	**crn1** **svl3**	–	–
RNSC$_{a,s}$	*las17* *yhr133c*	–	–
RNSC$_{a,s}$	*crn1* *svl3*	–	–
PINCoC	***cap1*** ***cap2*** ***ypr171w***	3/3	$4.63 \cdot 10^{-06}$
UVCluster	**cap1** **cap2** **ypr171w**	3/3	$4.63 \cdot 10^{-06}$
RNSC$_{a,s}$	*cap1* *cap2* *ypr171w*	3/3	$4.63 \cdot 10^{-06}$
PINCoC	***cdc42*** ***cla4*** ***gic2***	3/3	$1.11 \cdot 10^{-06}$
UVCluster	**cdc42** **cla4** **gic2**	3/3	$1.11 \cdot 10^{-06}$
RNSC$_a$	*cdc42* *cla4* *gic2*	3/3	$1.11 \cdot 10^{-06}$
RNSC$_s$	*cdc42* *cla4* *gic2* bni1	4/4	$2.83 \cdot 10^{-09}$
PINCoC	*sla1* ***ybr108w*** ypl246c *ygr268c*	3/3	$1.11 \cdot 10^{-06}$
UVCluster	rvs161 **ybr108w**	–	–
RNSC$_a$	*sla1* *ybr108w*	–	–
RNSC$_s$	*sla1* ypl246c *ygr268c*	2/3	$3.90 \cdot 10^{-04}$
PINCoC	***swe1*** ***hsl7***	2/2	$5.3 \cdot 10^{-04}$
UVCluster	**swe1** **hsl7** app1	2/3	$1.91 \cdot 10^{-03}$
RNSC$_{a,s}$	*swe1* *hsl7* app1	2/3	$1.91 \cdot 10^{-03}$

Table 1. Clusters validation by Gene Ontology term finder, updated at September 2007, for *PINCoC*, UVCLUSTER [1] and RNSC [8].

5. C. Brun, C. Herrmann, and A. Guenoche. Clustering proteins from interaction networks for the prediction of cellular functions. *BMC Bioinformatics*, 5, 2004.
6. B.L. Drees, B. Sundin, and *et al*. A protein interaction map for cell polarity development. *Journal of Cellular Biology*, 154:549–571, 2001.
7. R. D. A. Jain. *Algorithms for Clustering Data*. Prentice Hall, 1988.
8. A. D. King, Natasa Przulj, , and Igor Jurisica. Protein complex prediction via cost-based clustering. *Bioinformatics*, 20(17):3013–3020, 2004.
9. 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.
10. N. Orlev, R. Shamir, and Y. Shiloh. Pivot: Protein interaction visualization tool. *Bioinformatics*, 20:424–425, 2004.
11. N. Przulj, D. A. Wigle, and I. Jurisica. Functional topology in a network of protein interactions. *Bioinformatics*, 20(3):340–348, 2004.
12. 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.
13. D. Ucar, S. Parthasarathy, S. Asur, and C. Wang. Effective pre-processing strategies for functional clustering of a protein-protein interaction network. In *IEEE Int. Symposium on Bioinformatics and Bioengeneering (BIBE'2005)*, pages 129–136, 2005.