Restricted Neighborhood Search Clustering Revisited: An Evolutionary Computation Perspective

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

[1] Institute for High Performance Computing and Networking,
National Research Council of Italy, CNR-ICAR,
Via P. Bucci 41C, 87036 Rende (CS), Italy
pizzuti@icar.cnr.it
[2] Department of Mathematics, Computer Science Section
Universitá degli Studi di Palermo, Palermo, Italy
90123 Palermo, via Archirafi 34, Italy
simona.rombo@math.unipa.it

Abstract. Protein-protein interaction networks have been broadly studied in the last few years, in order to understand the behavior of proteins inside the cell. Proteins interacting with each other often share common biological functions or they participate in the same biological process. Thus, discovering protein complexes made of groups of proteins strictly related, can be useful to predict protein functions. Clustering techniques have been widely employed to detect significative biological complexes. In this paper, we integrate one of the most popular network clustering techniques, namely the Restricted Neighborhood Search Clustering (RNSC), with evolutionary computation. The two cost functions introduced by $RNSC$, besides a new one that combines them, are used by a Genetic Algorithm as fitness functions to be optimized. Experimental evaluations performed on two different groups of interactions of the budding yeast *Saccaromices cerevisiae* show that the clusters obtained by the genetic approach are more accurate than those found by $RNSC$, though this method predicts more true complexes.

1 Introduction

Proteins are the basic constituents of living beings. It has been shown that studying how proteins interact inside the cell is necessary to understand the biological processes in which they are involved [37]. Thanks to the development of advanced high-throughput technologies, many protein-protein interactions have been discovered in the last few years (see, e.g., [15,21]). The set of all the protein-protein interactions of a given organism is its *interactome*, usually modeled by an indirect graph, called *protein-protein interaction network* (PPI network), where nodes represent involved proteins and edges encode their interactions. PPI networks received much attention in the last few years [2,10,33,36] since they can be usefully exploited to study protein functions and to infer information about conservations among species.

Proteins are organized into different putative protein complexes, each performing specific tasks in the cell [12,26]. Proteins interacting with each other often participate in the same biological processes, or can be associated with specific biological functions

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 59–68, 2013.
© Springer-Verlag Berlin Heidelberg 2013

being strongly related [35]. Indeed, cellular functions are likely to be accomplished in a modular way, meaning that a group of physically or functionally related proteins join together to accomplish a distinct function [4]. A protein complex can then be considered as a group (cluster) of proteins contributing to the same biological functions. Their detection allows the comprehension of biologically meaningful interactions and provides important knowledge about the organization of biological systems and cellular processes, giving a valuable help in understanding the behavior of organisms.

In the last few years there has been an increasing interest in studying clustering methods able to detect groups of proteins densely interconnected. Clustering approaches to PPI networks can be broadly categorized as distance-based and graph-based ones [17]. Distance-based clustering approaches apply traditional clustering techniques, such as hierarchical clustering, by employing the concept of distance between two proteins [5,25]. Graph-based clustering techniques consider the topology of the network. These techniques find the clusters by applying different strategies. One strategy searches for sub-graphs having maximum density (e.g., [23,28], by using different notions of subgraph density. Another approach partitions the graph by optimizing a cost function [14,34]. The concept of flow simulation, though applied in different ways, is exploited in [7,13]. A statistical approach to protein clustering is taken instead in [32,9]. Very few population-based stochastic search approaches have been used for developing algorithms for community detection in PPI networks (see, e.g., [18,30,31]). Surveys describing and comparing a number of methods presented in the literature can be found in [6,22,27,29,38].

In this paper we propose to embed the cost functions introduced by King et al. [14] in a genetic algorithm, in order to evaluate the capability of evolutionary computation in predicting complexes in PPI networks. Besides the *naive cost function* and *scaled cost function*, defined in [14], a new scaled function, that takes into account the connections of nodes constituting a cluster and the size of the clusters obtained, is introduced. Experimental results on two data sets of yeast protein interactions show that the genetic approaches, when compared with $RNSC$, though predict a lower number of complexes, the predicted clusters are composed of a high percentage of true positive proteins, thus a lower number of false positive occur inside them.

The paper is organized as follows. Section 2 briefly recalls the Restricted Neighborhood Search Clustering (RNSC) Algorithm. In Section 3 its evolutionary version is proposed and described in details. In Section 4 the evaluation measures exploited to validate the performances of the introduced methods are summarized. Section 5 describes experimental evaluations performed on the budding yeast *Saccaromices cerevisiae* PPI network and points out some peculiar characteristics of the evolutionary techniques proposed in this work. Finally, in Section 6 we draw our conclusive remarks.

2 Restricted Neighborhood Search Clustering Algorithm

Restricted Neighborhood Search Clustering (RNSC) is a popular method, proposed by King et al. [14], to detect complexes in protein-protein interaction networks. $RNSC$ explores the solution space of all the possible clusterings by minimizing cost functions that reflect the number of inter-cluster and intra-cluster edges. The method partitions a

network in clusters by using two cost functions. In order to formally define these two cost functions, some formalism must be introduced.

Let $G = (V, E)$ be a graph of n nodes and m edges modeling a PPI network, and $\mathcal{S} = \{S_1, \ldots, S_k\}$ a partitioning of G in k clusters. A cross-edge in a clustering is an edge whose vertices belong to different clusters. Given a node $v \in S$, let $c_s(v) = \{(v, u) \mid u \notin S\}$ denote the number of cross-edges incident with v, and $l_s(v) = \{u \in S \mid (v, u) \notin E\}$ be the number of nodes in S not connected with v.

The first function, called the *naive cost function*, is defined as:

$$C_n(G, \mathcal{S}) = \frac{1}{2} \sum_{v \in V} (c_s(v) + l_s(v)) \tag{1}$$

Thus, 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 ($c_s(v)$), or one that does not exist between v and another node in the same cluster as v ($l_s(v)$).

As the authors point out, $C_n(G, \mathcal{S})$ is considered naive since it does not take into account the importance of a vertex in a graph, i.e. if it belongs to either a very large cluster or a small cluster. To reflect this concept, a second function, called the *scaled cost function*, that measures the size of the area that v effects in the clustering is introduced:

$$C_s(G, \mathcal{S}) = \frac{n-1}{3} \sum_{v \in V} \frac{(c_s(v) + l_s(v))}{\mid N(v) \cup S_v \mid} \tag{2}$$

where S_v is the cluster v belongs to, and $N(v)$ is the set of neighbour nodes of v.

The algorithm begins with a random clustering, and attempts to find a best naive clustering by moving a vertex from a cluster to another one in order to minimize the naive cost function. The choice of using the naive cost function at first, is due to the necessity of having a fairly good clustering in a fast way. Then the algorithm tries to improve the obtained solution by searching for a clustering with low scaled cost function. Since the approach is greedy, the problem of getting stack at poor local minima is dealt by making diversification moves that mix up the clustering by scattering the clusters at random. Furthermore, RNSC maintains a list of tabu moves that forbid to cycle back to previously examined solutions.

3 Evolutionary RNSC

In this section we consider the cost functions described above, and reformulate them in terms of set of nodes constituting a cluster, instead of single nodes, to obtain fitness functions that will be optimized by the evolutionary approach. Furthermore, a simplification of the scaled cost function which scales the cost function with respect to the cluster size and the crossing edges of the cluster is introduced. These three objective functions will be adopted in the genetic approach and compared with $RNSC$.

Let $\mathcal{S} = \{S_1, \ldots, S_k\}$ be a partition of the graph $G = (V, E)$, modeling a PPI network, in k clusters. Let n_s and m_s denote the number of nodes and edges, respectively, of a cluster $S \in \mathcal{S}$. Then:

$$c_s = \sum_{v \in S} c_s(v)$$

is the total number of cross-edges of the nodes of S, and

$$\overline{l}_s = \sum_{v \in S} l_s(v)$$

is the number of pairs of nodes in S not connected. The naive cost function $C_n(G, \mathcal{S})$ can be rewritten as:

$$C_n(G, \mathcal{S}) = \frac{1}{2} \sum_{s \in \mathcal{S}} c_s + \overline{l}_s \tag{3}$$

As regards the scaled cost function, we must first compute the scaled cost function for each cluster $S \in \mathcal{S}$ as follows:

$$C_s(S) = \sum_{v \in S} \frac{c_s(v) + l_s(v)}{c_s(v) + n_s} \tag{4}$$

and then sum the contribution of each of them:

$$C_s(G, \mathcal{S}) = \frac{n-1}{3} \sum_{s \in \mathcal{S}} C_s(S) \tag{5}$$

A simplification of the function (5), which scales the naive cost function of each cluster in $\mathcal{S}$ with respect to its size and the crossing edges relative to it, can be obtained as follows:

$$C_{ss}(G, \mathcal{S}) = \frac{n-1}{3} \sum_{s \in \mathcal{S}} \frac{c_s + \overline{l}_s}{c_s + n_s} \tag{6}$$

Formula (6), instead of considering the influence of a single node, it normalizes the contribution of each cluster found with respect to its size and number of connections with nodes of other clusters.

The three cost functions described above can be used inside a genetic algorithm as fitness functions to minimize, in order to partition the graph G modeling a network in dense groups of proteins.

The pseudo-code of the genetic approach is reported in Figure 1. The genetic algorithm uses the locus-based adjacency representation proposed in [24], and adopted also in [30]. In this graph-based representation an individual of the population consists of n genes $g_1, \ldots, g_n$ and each gene can assume allele values j in the range $\{1, \ldots, n\}$. Genes and alleles represent nodes of the graph $G = (V, E)$ modeling a PPI network, and a value j, assigned to the ith gene, means that proteins i and j are connected and clustered together. The initialization process assigns to each each node i one of its neighbors j. The kind of crossover operator adopted is uniform crossover. Given two parents, a random binary vector is created. Uniform crossover then selects the genes where the vector is a 0 from the first parent, and the genes where the vector is a 1 from the second parent, and combines the genes to form the child. The mutation operator,

Given a network $\mathcal{N}$ and the graph $\mathcal{G} = (V, E)$ modeling it, perform the following steps:

1. **create** an initial population of random individuals whose length equals the number n of nodes of G
2. **while** termination condition is not satisfied **do**
3. **decode** each individual $I = \{g_1, \ldots, g_n\}$ of the population to obtain a partitiong $S = \{S_1, \ldots, S_k\}$ of the graph G in k connected components
4. **evaluate** the fitness of the translated individuals
5. **create** a new population of individuals by applying the variation operators
6. **end while**
7. **return** the individual having the best cost function

Fig. 1. The pseudo-code of the *GA-RNSC* approach

analogously to the initialization process, randomly assigns to each node i one of its neighbors.

The algorithm, for a fixed number of generations, evolves the population of individuals, decodes each chromosome to determine the division of the graph in k connected components, computes the fitness function of each member of the population, and applies the specialized variation operators described above to produce the new population. At the end of the evolution process, the individual having the best cost function is returned as solution. It is worth to note that decoding can be efficiently performed by using a disjoint set algorithm, as described in [8].

4 Evaluation Measures

In the following we describe some validation measures widely exploited in the literature [1,3,16] that will be used for the comparative analysis presented in this work. For the generic predicted cluster P_i and the generic known complex K_j, let $\mid P_i \mid$ and $\mid K_j \mid$ be their sizes, respectively. Furthermore, let $\mid P_i \cap K_j \mid$ be the size of the intersection set of the predicted cluster and the known complex. To evaluate how a predicted cluster P_i matches a known complex K_j, the *overlapping score* between P_i and K_j is defined as

$$OS(P_i, K_j) = \frac{\mid P_i \cap K_j \mid^2}{\mid P_i \mid \cdot \mid K_j \mid} \tag{7}$$

A known complex and a predicted cluster are considered a *match* [16] if $OS(P_i, K_j) \geq \sigma_{OS}$, i.e. their overlapping score is equal to or larger than a specific threshold σ_{OS}. To estimate the performance of algorithms for detecting protein complexes w.r.t. the overlapping score, the notions of *sensitivity* and *specificity*, commonly used in information retrieval and machine learning (also known as *recall* and *precision*), as well as a cumulative measure called *f-measure* are introduced.

Sensitivity: $S_n = \frac{TP}{TP+FN}$ is the fraction of the true-positive predictions out of all the true predictions, where TP (true positive) is the number of the predicted clusters matched by the known complexes with $OS(P_i, K_j) \geq \sigma_{OS}$, and FN (false negative) is the number of the known complexes that are not matched by the predicted clusters.

Specificity: $S_p = \frac{TP}{TP+FP}$ is the fraction of the true-positive predictions out of all the positive predictions, where FP (false positive) equals the total number of the predicted clusters minus TP.

F-measure: $F_m = \frac{2 \cdot S_n \cdot S_p}{S_n + S_p}$ is a measure that summarizes sensitivity and specificity. High values of f-measure means that both sensitivity and specificity are sufficiently high.

5 Experimental Results

In this section we present the results of the genetic approaches on two PPI networks and compare them with those obtained by running $RNSC$. In the following, depending on the fitness function used, i.e. formulas (3) for naive cost function, (5) for scaled cost function, and (6) for simplified scaled cost function, we refer to the genetic algorithm as GA_n-*RNSC*, GA_s-*RNSC*, and GA_{ss}-*RNSC*, respectively. The parameters of the genetic algorithm have been fixed as follows. Population size 100, number of generations 100, elite reproduction 10% of the population size, roulette selection function, crossover 0.8, mutation 0.2. This values have been chosen by taking into account the experimental evaluation reported in [30]. The implementation has been written in MATLAB 7.14 R2012a, using Genetic Algorithms and Direct Search Toolbox 2. As regards $RNSC$ we used the optimal parameter values reported in [6].

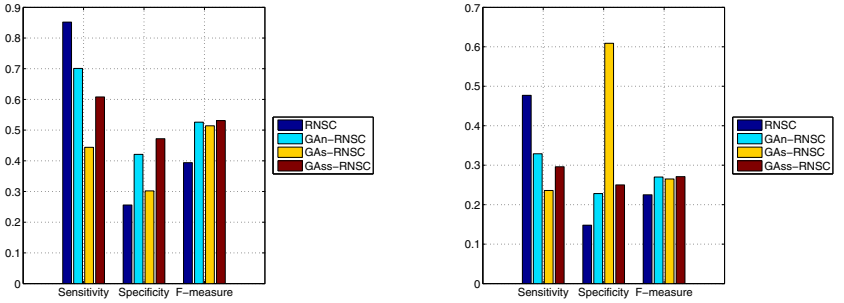
We ran the methods on two different data sets containing yeast protein interactions downloadable from http://faculty.uaeu.ac.ae/nzaki/ProRank.htm. The first dataset, denoted Yeast-D1, is that used by Gavin et al. in [11], and the second one, denoted Yeast-D2, contains yeast protein interactions generated by different experiments. Zaki et al. [39], however, filtered these two networks to delete unreliable interactions and obtained 990 proteins with $4,687$ interactions for Yeast-D1, and $1,443$ proteins with $6,993$ interactions for Yeast-D2. The reference sets of gold standard complexes include 81 (Cmplx-D1) and 162 (Cmplx-D2) hand-curated complexes from MIPS [19,20].

First of all in Table 1 the average number of complexes found by the genetic algorithms on the two yeast networks, along with the standard deviation std, are reported. The methods behave in a rather different way. $RNSC$ obtains the highest number of clusters. When the naive cost function (formula (3)) is adopted, a considerable number of clusters with smaller size with respect to the true complexes are obtained also by GA_n-*RNSC*. The opposite behavior can be observed with the scaled cost function

Table 1. Complexes found by the methods on Yeast-D1 and Yeast-D2 with 81 and 162 gold standard complexes, respectively

METHOD	YEAST-D1		YEAST-D2	
	NUMBER	STD	NUMBER	STD
RNSC	293	0	427	0
GA_n-*RNSC*	138.4	12.1	207.6	10.9
GA_s-*RNSC*	58.2	3.5	112.6	3.7
GA_{ss}-*RNSC*	107.8	3.2	171	3.6

(formula (5)) that induces a much lower number of clusters having larger size. With the simplified scaled cost function (formula (6)), GA_{ss}-*RNSC* produces a number of clusters higher than GA_s-*RNSC*, and lower than GA_n-*RNSC*. These numbers differ from the true number of complexes and suggest that $RNSC$ divides the complexes in small groups of proteins, GA_n-*RNSC* has a similar but less emphasized behavior, GA_s-*RNSC*, on the contrary, joins complexes, while GA_{ss}-*RNSC* also splits complexes, but for a lower percentage of groups than GA_n-*RNSC*. Thus the optimization of the cost functions of $RNSC$ through evolutionary computation produces predicted clusters that are sensibly dissimilar from those generated by $RNSC$.

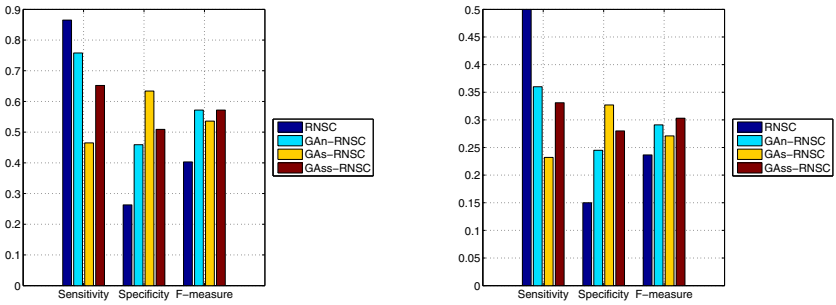


(a) Yeast-D1 (b) Yeast-D2

Fig. 2. Sensitivity, specificity, and f-measure values for (a) Yeast-D1 and (b) Yeast-D2 networks with overlapping score $OS(P_i, K_j) \geq 0.2$

Figure 2 shows sensitivity, specificity, and f-measure values obtained by the genetic approaches and $RNSC$ when the overlapping score $OS(P_i, K_j) \geq 0.2$. The first observation is that $RNSC$ has a higher sensitivity value compared with the genetic algorithms on both the two networks. This means that $RNSC$ is able to predict a higher number of complexes, out of all the true complexes. This result can be explained by the high number of clusters that $RNSC$ finds. It is worth to note that, the definition of overlapping score (formula (7)) penalizes those methods that obtain clusters with size $\mid P_i \mid$ much greater than the true complex size $\mid K_i \mid$. In fact the denumerator of (7) has a higher value if the cluster size $\mid P_i \mid$ is high, and, consequently, $OS(P_i, K_j)$ is lower. This bias can be observed also for the three evolutionary methods. GA_n-*RNSC*, GA_{ss}-*RNSC*, and GA_s-*RNSC* present a decreasing number of predicted clusters, and thus the predicted clusters are of increasing size. The figure shows that sensitivity values reflect the size of the predicted clusters. The lower the size, the higher the corresponding sensitivity values.

On the other hand, from the figure we can observe that specificity and f-measure are both higher for the genetic approaches. Higher specificity means that the predicted clusters have a high percentage of proteins effectively belonging to the true complex, thus the fraction of false positive is low. In particular, GA_s-*RNSC* is the best performing on Yeast-D2, while GA_{ss}-*RNSC* reaches better values of specificity on Yeast-D1.



(a) Yeast-D1 (b) Yeast-D2

Fig. 3. Sensitivity, specificity, and f-measure values for (a) Yeast-D1 and (b) Yeast D2 networks when overlapping score $OS_J(P_i, K_j) \geq 0.2$

In order to more deeply investigate the effects of the overlapping score $OS(P_i, K_j)$, we considered a different definition of overlapping score based on the Jaccard coefficient, that is:

$$OS_J(P_i, K_j) = \frac{\mid P_i \cap K_j \mid}{\mid P_i \cup K_j \mid} \tag{8}$$

Sensitivity, specificity and f-measure have been recomputed and the values obtained when the overlapping score $OS_J(P_i, K_j) \geq 0.2$ are reported in Figure 3. Also in this experiment it is possible to observe that sensitivity values obtained by $RNSC$ are higher. However, specificity and f-measure are better for all the three fitness functions used, confirming the above observations.

From the described experimental campaign, we can conclude that evolutionary computation allows to improve specificity w.r.t. the $RNSC$ method, still retaining good values of sensitivity. In particular, $RNSC$ returns in output many clusters, and each of them only partially overlaps with some true complexes. On the contrary, GA-$RNSC$ approaches predict a lower number of clusters, but their overlapping with true complexes is larger. As an example, GA_n-*RNSC* correctly found a complex of Yeast-D1 (20 of 22 proteins) recognized to be a RNA polymerase II holoenzyme/mediator subunit, while GA_s-*RNSC* was able to find a full complex in Yeast-D2 made of cAMP-dependent protein kinases.

6 Conclusions

In this work we showed the capability of evolutionary computation to predict complexes in PPI networks by embedding the cost functions introduced by King et al. [14] in a genetic algorithm. A new scaled function able to take into account, besides the connections of nodes constituting a cluster, also the size of the clusters obtained, is also introduced. Experimental results on two data sets of yeast protein interactions proved that the genetic approaches, when compared with $RNSC$, return complexes with a

higher percentage of true positive proteins. Future work aims to improve the evolutionary approach by considering different combinations of the fitness functions, possibly enriched with local search strategies.

Acknowledgements. This work has been partially supported by the project *MERIT : MEdical Research in Italy*, funded by MIUR.

References

1. Altaf-Ul-Amin, M., Shinbo, Y., Mihara, K., Kurokawa, K., Kanaya, S.: Development and implementation of an algorithm for detection of protein complexes in large interaction networks. BMC Bioinformatics 7(207) (2006)
2. Atias, N., Sharan, R.: Comparative analysis of protein networks: hard problems, practical solutions. Commun. ACM 55(5), 88–97 (2012)
3. Bader, G., Hogue, H.: An automated method for finding molecular complexes in large protein-protein interaction networks. BMC Bioinformatics 4(2) (2003)
4. Barabási, A., Oltvai, Z.N.: Network biology: Understanding the cell's functional organization. Nature Review Genetics 5, 101–113 (2004)
5. Blatt, M., Wiseman, S., Domany, E.: Superparamagnetic clustering of data. Phisical Review Letters 76(18), 3251–3254 (1996)
6. Brohèe, S., van Helden, J.: Evaluation of clustering algorithms for protein-protein interaction networks. BMC Bioinformatics 7, 488 (2006)
7. Cho, Y.-R., Hwang, W., Ramanathan, M., Zhang, A.: Semantic integration to identify overlapping functional modules in protein interaction networks. BMC Bioinformatics 8, 265 (2007)
8. Thomas, H., Cormen, C.E., Leiserson, R.L.: Rivest, and Clifford Stein. In: Introduction to Algorithms, 2nd edn. MIT Press (2007)
9. Farutin, V., Robinson, K., Lightcap, E., Dancik, V., Ruttenberg, A., Letovsky, S., Pradines, J.: Edge-count probabilities for the identification of local protein communities and their organization. Proteins: Structure, Function, and Bioinformatics 62, 800–818 (2006)
10. Ferraro, N., Palopoli, L., Panni, S., Rombo, S.E.: Asymmetric comparison and querying of biological networks. IEEE/ACM Transactions on Computational Biology and Bioinformatics 8, 876–889 (2011)
11. Gavin, A.C., et al.: Proteome survey reveals modularity of the yeast cell machinery. Nature 440, 631–636 (2006)
12. Hartwell, L.H., Hopfield, J.J., Leibler, S., Murray, A.W.: Clustering algorithm based graph connectivity. Nature 402, 47–52 (1999)
13. Hwang, W., Cho, Y.-R., Zhang, A., Ramanathan, M.: A novel functional module detection algorithm for protein-protein interaction networks. Algorithms for Molecular Biology 1(24) (2006)
14. King, A.D., Przulj, N., Jurisica, I.: Protein complex prediction via cost-based clustering. Bioinformatics 20(17), 3013–3020 (2004)
15. Krogan, N.J., et al.: Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. Nature 440(7084), 637–643 (2006)
16. Li, M., Chen, J., Wang, J., Hu, B., Chen, G.: Modifying the DPClus algorithm for identifying protein complexes based on new topological structures. BMC Bioinformatics 9 (2008)
17. Lin, C., Cho, Y., Hwang, W., Pei, P., Zhang, A.: Clustering methods in protein-protein interaction network. Knowledge Discovery in Bioinformatics: Techniques, Methods and Application. John Wiley & Sons, Inc. (2006)
18. Liu, H., Liu, J.: Clustering protein interaction data through chaotic genetic algorithm. In: Wang, T.-D., Li, X., Chen, S.-H., Wang, X., Abbass, H.A., Iba, H., Chen, G.-L., Yao, X. (eds.) SEAL 2006. LNCS, vol. 4247, pp. 858–864. Springer, Heidelberg (2006)

19. Mewes, H.W., et al.: MIPS: a database for genomes and protein sequences. Nucleic Acids Res. 30(1), 31–34 (2002)
20. Mewes, H.W., et al: MIPS: analysis and annotation of proteins from whole genomes in 2005. Nucleic Acids Res. 34(database issue 1), 169–172 (2006)
21. Miller, J.P., et al.: Large-scale identification of yeast integral membrane protein interactions. Proc. Natl. Acad. Sci. USA 102(34), 12123–12128 (2005)
22. Moschopoulos, C.N., Pavlopoulos, P.A., Iacucci, E., Aerts, J., Likothanassis, S., Schneider, R., Kossida, S.: Which clustering algorithm is better for predicting protein complexes? BMC Research Notes 4(549) (2011)
23. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435, 814–818 (2005)
24. Park, Y.J., Song, M.S.: A genetic algorithm for clustering problems. In: Proc. of 3rd Annual Conference on Genetic Algorithms, pp. 2–9 (1989)
25. Pei, P., Zhang, A.: A two-step approach for clustering proteins based on protein interaction profiles. In: IEEE Int. Symposium on Bioinformatics and Bioengeneering (BIBE 2005), pp. 201–209 (2005)
26. Pereira, J.B., Enright, A.J., Ouzounis, C.A.: Detection of functional modules from protein interaction networks. Proteins: Structure, Fuctions, and Bioinformatics (20), 49–57 (2004)
27. Pizzuti, C., Rombo, S.E.: Discovering Protein Complexes in Protein Interaction Networks in Biological Data Mining in Protein Interaction Networks. In: Li, X.-L., Ng, S.-K. (eds.). IGI Global- Medical Inf. Science Ref. (2009)
28. Pizzuti, C., Rombo, S.E.: A coclustering approach for mining large protein-protein interaction networks. IEEE/ACM Trans. Comput. Biology Bioinform. 9(3), 717–730 (2012)
29. Pizzuti, C., Rombo, S.E., Marchiori, E.: Complex detection in protein-protein interaction networks: A compact overview for researchers and practitioners. In: Giacobini, M., Vanneschi, L., Bush, W.S. (eds.) EvoBIO 2012. LNCS, vol. 7246, pp. 211–223. Springer, Heidelberg (2012)
30. Pizzuti, C., Rombo, S.E.: Experimental evaluation of topological-based fitness functions to detect complexes in PPI networks. In: Proc. of the Genetic and Evolutionary Computation Conference (Gecco 2012), pp. 193–200 (2012)
31. Ravaee, H., Masoudi-Nejad, A., Omidi, S., Moeini, A.: Improved immune genetic algorithm for clustering protein-protein interaction network. In: Proceedings of the 2010 IEEE International Conference on Bioinformatics and Bioengineering, BIBE 2010, pp. 174–179. IEEE Computer Society (2010)
32. Samantha, M.P., Liang, S.: Predicting protein functions from redundancies in large-scale protein interaction networks. Proc. of the National Academy of Science 100(22), 12579–12583 (2003)
33. Sharan, R., Ulitsky, I., Shamir, R.: Network-based prediction of protein function. Molecular Systems Biology 3(88) (2007)
34. Spirin, V., Mirny, L.A.: Protein complexes and functional modules in molecular networks. PNAS 100, 12123–12128 (2003)
35. Tornw, S., Mewes, H.W.: Functional modules by relating protein interaction networks and gene expression. Nucleic Acids Research 31(21), 6283–6289 (2003)
36. De Virgilio, R., Rombo, S.E.: Approximate matching over biological RDF graphs. In: Proceedings of the ACM Symposium on Applied Computing, SAC 2012, pp. 1413–1414 (2012)
37. von Mering, D., Krause, C., et al.: Comparative assessment of a large-scale data sets of protein-protein interactions. Nature 31, 399–403 (2002)
38. Wang, J., Li, M., Deng, Y., Pan, Y.: Recent advances in clustering methods for protein interaction networks. BMC Genomics 11(suppl. 3), S10 (2010)
39. Zaki, N., Berengueres, J., Efimov, D.: Prorank: a method for detecting protein complexes. In: Proc. of the Genetic and Evolutionary Computation Conference (Gecco 2012), pp. 209–216 (2012)