Optimizing filter processes on protein interaction clustering results using genetic algorithms

Charalampos Moschopoulos[1], Grigorios Beligiannis[2], Sophia Kossida[3] and Spiridon Likothanassis[1]

[1] Computer Engineering & Informatics,University of Patras, GR-26500 Rio, Patras, Greece
(mosxopul, likothan)@ceid.upatras.gr
[2] Department of Business Administration of Food and Agricultural Enterprises, University of Western Greece, G. Seferi 2, GR-30100, Agrinio, Greece
gbeligia@cc.uoi.gr
[3] Bioinformatics & Medical Informatics Team, Biomedical Research Foundation of the Academy of Athens, Soranou Efesiou 4, GR-11527, Athens, Greece
skossida@bioacademy.gr

Abstract. In this manuscript, a Genetic Algorithm is applied on a filter in order to optimize the selection of clusters having a high probability to represent protein complexes. The filter was applied on the results (obtained by experiments made on five different yeast datasets) of three different algorithms known for their efficiency on protein complex detection through protein interaction graphs. The derived results were compared with three popular clustering algorithms, proving the efficiency of the proposed method according to metrics such as successful prediction rate and geometrical accuracy.

Keywords: protein-protein interactions, protein interaction graph, genetic algorithm, protein complexes prediction.

1 Introduction

The importance of protein interactions is given as they play important role on fundamental cell functions. They are crucial for forming structural complexes, for extra-cellular signaling, for intra-cellular signaling [1]. Recently, new high throughput experimental methods [2-5] have been developed which detect thousands protein-protein interactions (PPIs) with a single experiment. As a result, enormous datasets have been generated which could possibly describe the functional organization of the proteome. However, these data are extremely noisy [6], making it difficult for researchers to analyze them and extract valuable conclusion such as protein complex detection or characterizing the functionality of unknown proteins.

Due to the vast volume of PPI data, they are usually modeled as graphs G=(V,E) where V is the set of vertices (proteins) and E the set of adjacent edges between two nodes (protein interactions). The model of graph makes it easy for bioinformatics researchers to apply various algorithms derived from graph theory in order to perform

clustering and detect protein complexes which are represented as dense subgraphs [7-9]. According to [10, 11], the most prevailed algorithms are MCL (Markov clustering) [12] and RNSC (Restricted Neighbourhood Search Clustering) [13]. Besides them, spectral clustering can achieve similar results [14]. While these methods use the PPI graph structure to detect protein complexes, additional information could also be used such as gene expression data [15], functional information [16] as well as other biological information [17]. However, the use of additional information has the disadvantage that can not cover the aggregation of proteins that constitute the PPI graph.

The aforementioned algorithms assign each protein of the initial PPI graph to a cluster, constructing clusters that could hardly be characterized as dense ones. As a result, their prediction rate of protein complexes is pretty low. One way to deal with this problem is to filter the results of such an algorithm using additional information such as Gene Ontology [13]. However, the sources of the additional information usually do not cover all the recorded interactions that form the PPI graphs. Moreover, the parameters of these filters are almost always empirically defined, leading to biased solutions.

In this contribution, a filter is constructed by four methods which are based on graph properties such as density, haircut operation, best neighbor and cutting edge and it is applied on the results of MCL, RNSC and spectral algorithm. Furthermore, the parameters of the filter methods are optimized by a Genetic Algorithm (GA) which takes into account the rate of successful prediction, the absolute number of valid predicted protein complexes and the geometrical accuracy of the final clusters. Extended experiments were performed using five different PPI datasets. The derived results were compared with the recorded protein complexes of the MIPS database [18], while statistical metrics were calculated such as sensitivity (Sn), positive predictive value (PPV) and geometrical accuracy (Acc_g). To demonstrate the efficiency of the proposed filter, we compare the derived results with 3 other algorithms (SideS [8], Mcode [7], HCS [9]).

2 Our method

We chose to perform our filtering method on the results of 3 clustering algorithms that assign each protein of the initial PPI graph to a cluster and they are considered as the best of their category: MCL, RNSC and spectral. The MCL algorithm [12] is a fast and scalable unsupervised clustering algorithm based on simulation of stochastic flow in graphs. The MCL algorithm deterministically computes the probabilities of random walks through a graph and uses two operators transforming one set of probabilities into another. It does so by using the language of stochastic matrices (also called Markov matrices) which capture the mathematical concept of random walks on a graph. The RNSC algorithm [13] performs an initial random clustering and then iteratively by moving one node from one cluster to another is trying to improve the clustering cost. In order to avoid local minima, it maintains a tabu list that prevents cycling back to a previously explored partitioning. Due to the randomness of the algorithm, different runs on the same input data produce different outputs. For the

spectral clustering algorithm, we used [19] spectral graph decomposition and mapped the set of nodes of PPI graph into the k-dimensional space. Following the spectral decomposition, the EM algorithm [20] was applied to produce the final clusters.

The developed filter was based on four graph metrics that would help to detect the denser clusters out of the above algorithms results and it was first introduced in [21]. These graph metrics are:

- **Density** of a subgraph is calculated as $2|E|/|V|(|V|-1)$ where $|E|$ is the number of edges and $|V|$ the number of vertices of the subgraph.
- **Haircut operation** is a method that detects and excludes vertices with low degree of connectivity from the potential cluster that these nodes belong to.
- **Best neighbor** tends to detect and enrich the clusters with candidate vertices that the proportion of their edges adjacent to the cluster divided by the total degree of the vertex is above a threshold.
- **Cutting edge** metric is used to detect those clusters that are more isolated than the remaining of the graph by dividing the number of edges that are adjacent to two cluster vertices with the total number of edges of the cluster.

The difference of the filter presented in this manuscript with the filter presented in [21] is that its parameters are optimized by a genetic algorithm in order to achieve better quality results concerning the rate of successful prediction, the detection of more real protein complexes and the highest accuracy.

2.1 Optimizing clustering results filter

Genetic Algorithms (GAs) are one of the most popular techniques to solve optimization problems with very satisfactory results [22, 23]. In this manuscript, a GA is used to optimize the fitness function containing metrics such as successful prediction, absolute number of valid protein complexes and accuracy by choosing the appropriate values for the filter parameters. In order to implement the GA, we used GALIB library [24] which is a set of C++ GA objects and well known about its computing efficiency.

The chromosome used in our case is represented as a one dimensional array, where each cell represents a filter parameter: the first parameter is used in density method, the second one in best neighbor method, the third in cutting edge method and the fourth in haircut method.

The proposed GA uses simple mutation and crossover operations while as a selection scheme we decided to use Roulette Wheel selection [25]. The mathematical representation of the evaluation function is as follows:

$$Max(10*percentage+0,05*\#valid\_clusters+5*Acc\_g)$$

where percentage is the percentage of successful predictions, #valid_clusters is the absolute number of valid clusters, Acc_g is the geometrical accuracy. Depending on how important each metric of the fitness function is, we multiplied it with a constant number. In our case, we considered as best solutions those that succeed high prediction rate without having small absolute number of valid predicted protein complexes. These values were selected after performing exhaustive experiments.

Finally, in order to retain the best chromosome in every generation, as elitism schema [25, 26] is used. The best chromosome of each generation is copied to the next generation (and not to the intermediate one, the individuals of which may crossover or mutate) assuring that it is preserved in the current population for as long as it is the best compared to all other chromosomes of the population. This choice assures that the best chromosome of each generation will be, if not better, at least equal to the best chromosome of the previous generation. Furthermore, it helps GA to converge faster to a near optimal solution.

3 Experimental Results

In order to prove the efficiency of our method, we performed experiments with 5 different yeast datasets derived by online databases (MIPS [18], DIP [27]) or individual experiments (Krogan [28], Gavin_2002 [29], Gavin_2006 [30]) and compared the results with the algorithms: SideS, HCS and Mcode. As an evaluation set, we used the recorded yeast complexes stored in MIPS database.

For all the results presented in this section the same set of GA's operators and parameters were used in order to provide a fair comparison of the algorithm's efficiency and performance. The crossover probability was equal to 0.7, while the mutation probability was equal to 0.1. The size of the population was set to 20 and the algorithm was left to run for 200 generations (the number of generations was used as termination criterion).

In Figure 1, the percentage of successful predictions is presented. As it can easily be seen, the optimized filtered algorithms achieve better percentages than those of the other algorithms in all cases. The combination of the MCL algorithm with the optimized filter gives the best results except in the case where the MIPS dataset is tested.

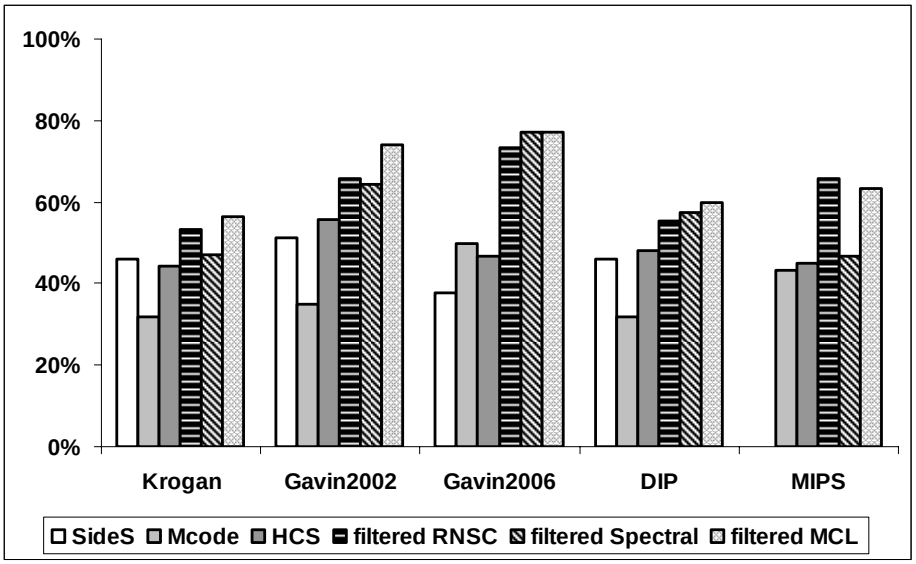

Fig. 1. Percentage of successful prediction.

Concerning the geometrical accuracy metric, the results are satisfactory comparing to all other algorithms (Figure 2). In most of the cases, our methodology surpasses the other algorithms. In contrast with Figure 1, it seems that the combination of RNSC algorithm and the optimized filter is equally good with the one which uses MCL.

On the other hand, as it is shown in Figure 3, the absolute number of valid predicted clusters is lower than the other algorithms. The variables used in the fitness function of the GA caused those results. There is a trade off between the absolute number of valid predicted clusters and the other metrics used in the fitness function. We decided to give a priority to the quality of the produced solutions even if this leads to fewer final clusters.

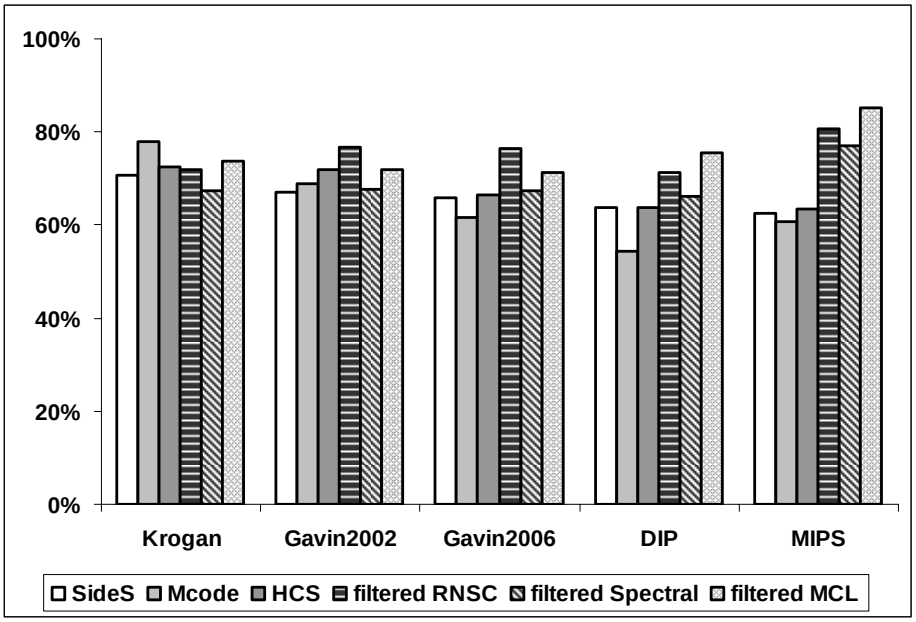

Fig. 2. Geometrical accuracy of results.

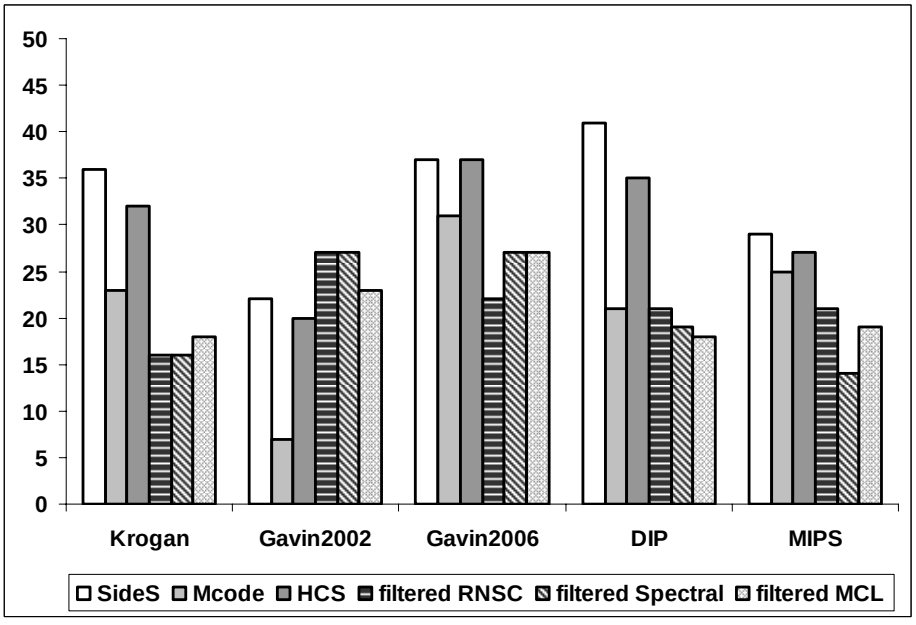

Fig. 3. Absolute number of the resulting valid predicted clusters.

Finally, it has to be noted that after the GA training, the parameters of the filter methods have different values depending on the input dataset, as it is shown in Table

1. This proves that there is not an optimal solution that suits in all datasets. The filter composition should be adjusted in each time specific PPI dataset properties in order to obtain better results.

Table 1. Parameters of the filter methods in each time dataset.

	MCL	RNSC	Spectral
Krogan	Density: 0.370 Cutting Edge: 0.455 Best Neighbor: 0.216 Haircut: 4.528	Density: 0.306 Cutting Edge: 0.753 Best Neighbor: 0.825 Haircut: 0.059	Density: 0.027 Cutting Edge: 0.584 Best Neighbor: 0.670 Haircut: 4.539
Gavin2002	Density: 0.250 Cutting Edge: 0.609 Best Neighbor: 0.887 Haircut: 4.983	Density: 0.342 Cutting Edge: 0.613 Best Neighbor: 0.587 Haircut: 3.615	Density: 0.123 Cutting Edge: 0.602 Best Neighbor: 0.836 Haircut: 3.672
Gavin2006	Density: 0.173 Cutting Edge: 0.685 Best Neighbor: 0.824 Haircut: 4.807	Density: 0.010 Cutting Edge: 0.674 Best Neighbor: 0.905 Haircut: 4.610	Density: 0.196 Cutting Edge: 0.647 Best Neighbor: 0.695 Haircut: 3.357
DIP	Density: 0.503 Cutting Edge: 0.399 Best Neighbor: 0.616 Haircut: 1.174	Density: 0.344 Cutting Edge: 0.398 Best Neighbor: 0.606 Haircut: 4.785	Density: 0.153 Cutting Edge: 0.398 Best Neighbor: 0.630 Haircut: 3.152
MIPS	Density: 0.711 Cutting Edge: 0.365 Best Neighbor: 0.798 Haircut: 2.729	Density: 0.700 Cutting Edge: 0.237 Best Neighbor: 0.958 Haircut: 3.196	Density: 0.437 Cutting Edge: 0.050 Best Neighbor: 0.209 Haircut: 3.196

4 Conclusions

In this contribution, we presented a filter, optimized by a GA, which was applied on the results of three well known for their efficiency clustering algorithms namely MCL, RNSC and spectral. Furthermore, we compared the derived results with three other algorithms: SideS, HCS and Mcode to demonstrate the superiority of the proposed method. For the implementation of the GA we used GALIB library, while the filter is composed by four different methods derived from graph theory. As a future prospective, we plan to apply the proposed methodology on specific experimental methods datasets to extract rules about the properties that a cluster in these PPI graphs should have in order to be considered as protein complex.

References

1. Ryan, D.P. and J.M. Matthews: Protein-protein interactions in human disease. Curr Opin Struct Biol, 15(4): 441-6 (2005)
2. Ito, T., et al.: A comprehensive two-hybrid analysis to explore the yeast protein interactome Proceedings of the National Academy of Science, 98(8): 4569-4574 (2001)
3. Puig, O., et al.: The tandem affinity purification (TAP) method: a general procedure of protein complex purification. Methods, 24(3): 218-29 (2001)
4. Stoll, D., et al.: Protein microarrays: applications and future challenges. Curr Opin Drug Discov Devel, 8(2): 239-52 (2005)
5. Willats, W.G.: Phage display: practicalities and prospects. Plant Mol Biol, 50(6): 837-54 (2002)
6. Sprinzak, E., S. Sattath, and H. Margalit: How Reliable are Experimental Protein-Protein Interaction Data? Journal of Molecular Biology, 327: 919-923 (2003)
7. Bader, G.D. and C.W. Hogue: An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinformatics, 4: 2 (2003)
8. Koyuturk, M., W. Szpankowski, and A. Grama: Assessing significance of connectivity and conservation in protein interaction networks. J Comput Biol, 14(6): 747-64 (2007)
9. Hartuv, E. and R. Shamir: A clustering algorithm based on graph connectivity Information Processing Letters, 76(4-6): 175-181 (2000)
10. Brohee, S. and J. van Helden: Evaluation of clustering algorithms for protein-protein interaction networks. BMC Bioinformatics, 7: 488 (2006)
11. Li, X., et al.: Computational approaches for detecting protein complexes from protein interaction networks: a survey. BMC Genomics, 11 Suppl 1: S3 (2009)
12. Enright, A.J., S. Van Dongen, and C.A. Ouzounis: An efficient algorithm for large-scale detection of protein families. Nucleic Acids Res, 30(7): 1575-84 (2002)
13. King, A.D., N. Przulj, and I. Jurisica: Protein complex prediction via cost-based clustering. Bioinformatics, 20(17): 3013-20 (2004)
14. Kritikos, G., et al.: Spectral Clustering of Weighted Protein Interaction Networks. submitted, (2010)
15. Eisen, M.B., et al.: Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci U S A, 95(25): 14863-8 (1998)
16. The Gene Ontology (GO) project in 2006. Nucleic Acids Res, 34(Database issue): D322-6 (2006)
17. Huh, W.K., et al.: Global analysis of protein localization in budding yeast. Nature, 425(6959): 686-91 (2003)
18. Mewes, H.W., et al.: MIPS: analysis and annotation of proteins from whole genomes in 2005. Nucleic Acids Res, 34(Database issue): D169-72 (2006)
19. Ng, A.Y., M.I. Jordan, and Y. Weiss: On Spectral Clustering: Analysis and an algorithm Advances in Neural Information Processing Systems, 14: 849 - 856 (2001)

20. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc. (2006)
21. Moschopoulos, C.N., et al.: An enchanced Markov clustering method for detecting protein complexes. In: 8th IEEE International Conference on BioInformatics and BioEngineering (BIBE 2008), pp., Athens (2008)
22. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley (1989)
23. Bandyopadhyay, S. and S.K. Pal: Classification and Learning Using Genetic Algorithms: Applications in Bioinformatics and Web Intelligence. Springer (2007)
24. GALIB, http://lancet.mit.edu/galib-2.4/.
25. Mitchell, M.: An Introduction to Genetic Algorithms. MIT press, London (1995)
26. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, New York (1999)
27. Xenarios, I., et al.: DIP: the database of interacting proteins. Nucleic Acids Res, 28(1): 289-91 (2000)
28. Krogan, N.J., et al.: Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. Nature, 440(7084): 637-43 (2006)
29. Gavin, A.C., et al.: Functional organization of the yeast proteome by systematic analysis of protein complexes. Nature, 415(6868): 141-7 (2002)
30. Gavin, A.C., et al.: Proteome survey reveals modularity of the yeast cell machinery. Nature, 440(7084): 631-6 (2006)