MTopGO: a tool for module identification in PPI Networks

Danila Vella[1,2], Simone Marini[3,4], Francesca Vitali[5,6,7], Riccardo Bellazzi[1,4]

[1]Clinical Scientific Institute Maugeri, Pavia, Italy, [2] Department of Informatics, Systems and Communication, University of Milano-Bicocca, Milan, Italy, [3]Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI, USA, [4] University of Pavia, Pavia, Italy, [5] Center for Biomedical Informatics and Biostatistics, The University of Arizona, Tucson, AZ, USA, [6] BIO5 Institute, The University of Arizona, Tucson, AZ, USA, [7] Department of Medicine, The University of Arizona, Tucson, AZ, USA

Contact: simone.marini@unipv.it, d.vella1@campus.unimib.it

Introduction

With the advance in high-throughput technologies, we are experiencing an increase in amount and quality of –omics data. These technologies provide a more and more accurate snapshot of the investigated biological systems. However, a challenge remains in how to decipher the molecular mechanisms underlying the emergent phenotypes from the large volume of biomedical data. A common approach to interpret these data is the analysis of protein-protein interaction (PPI) networks, in particular network clustering [1,2]. Network clustering aims at identifying network regions showing specific topological and/or functional characteristics, commonly called modules. The identification of such modules can therefore be crucial to better understand the mechanisms underlying a disease and to suggest novel drug treatments. Due to their large size, typical ranging from thousand to tens of thousands of nodes and edges, the analysis of PPI networks is not a trivial task and it requires efficient computation methods to automatically process the contents [3].
To this aim we introduce MTopGO, an algorithm for module identification that exploits both topological network properties and biological knowledge. The output of the developed approach consists in the network partition and it provides for each identified module the biological function (GO term) that better describes it. In this way, in a single step, the network can be analysed both under the topological aspect, thanks the identification of a meaningful partition, and under the biological aspect, through the identification of the main cellular mechanisms involving the network proteins.

Methods

A PPI network can be represented as a graph $G=(V,E,\mathbf{\Delta})$, where V is the set of nodes, E is the set of edges, and $\mathbf{\Delta}$ is the set of GO terms associated with the network nodes. Each element of the set $\mathbf{\Delta}$ is a GO term and it points to the subset of proteins (nodes) annotated with that GO term. This network model can be used to investigate the main molecular processes/pathways involved in the biological system represented by PPI network. MTopGO is an algorithm designed to support the analysis of PPI networks, pursuing two main objectives: first, it finds the groups of nodes sharing interactions, called clusters or communities; second, it finds the Gene Ontology terms describing these groups. To perform this meaningful clustering, MTopGO employs repeated partitions of the network until a steady state is reached, improving at each iteration the topological and biological quality of the clusters. In detail, starting from a random partition, a new partition is created at each step by mixing up the nodes among the clusters. During this process, a *Moving List* is created to store the nodes that are hard to assign to clusters and used as a temporary depository. The *Moving List* is filled and emptied during each iteration. Next, a new partition is computed repeating, for each cluster $\mathbf{C_i}$, four main steps:

1. Selection of a GO term $\mathbf{\Delta_h}$ that better describes the cluster $\mathbf{C_i}$. This selection is performed, through the *Selection* function (1), by identifying the GO term $\mathbf{\Delta_h}$ that minimizes the number of nodes $\mathbf{N_a}$ in $\mathbf{C_i}$ not annotated with $\mathbf{\Delta_h}$ (i.e. $\mathbf{N_a} \in \mathbf{C_i}$ and $\mathbf{N_a} \notin \mathbf{\Delta_h}$)

and the number of nodes $\mathbf{N_c}$ not belonging to $\mathbf{C_i}$ but annotated with $\mathbf{\Delta_h}$ (i.e. $\mathbf{N_c} \notin \mathbf{C_i}$ and $\mathbf{N_c} \in \mathbf{\Delta_h}$). These two set of nodes are shown in Figure 1 with blue and yellow color, respectively.

2. The *Moving List* is filled with the nodes $\mathbf{N_a}$
3. The set of nodes $\mathbf{N_b}$ corresponding to the nodes belonging to $\mathbf{C_i}$ and $\mathbf{\Delta_h}$ (i.e.$\mathbf{N_b} \in \mathbf{C_i}$ and $\mathbf{N_b} \in \mathbf{\Delta_h}$) is used as core to create a new cluster
4. The remaining node set $\mathbf{N_c}$ is assigned to the new cluster according the value of *Contribute Modularity (CM)* function (2).

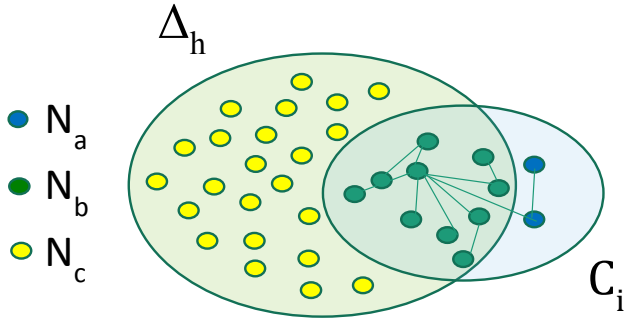

Figure 1 The cluster C_i and the three set of nodes Na, Nb, Nc used to compute the new clusters.

These steps are based on two functions: (i) *Selection* that accounts for the functional information of the nodes (GO); and (ii) *CM*, a function that takes into account the topological properties of the PPI network.
The *Selection* function is so defined:

$$Selection(\mathrm{Ci}, \Delta\mathbf{h}) = \frac{|\mathrm{Nc}|}{|\Delta\mathbf{h}| - 1} + \frac{|\mathrm{Na}|}{|\mathbf{Ci}| - 1} \quad (1)$$

This function is used to assign to a cluster a best fitting GO as model to drive the building process of a cluster.
The *CM* function is so defined:

$$CM\ (C_i,N_c) = \boldsymbol{q}(C_i+N_c) - \boldsymbol{q}(C_i-N_c)\ ;\ \boldsymbol{q}(\mathrm{Ci}) = \frac{\mathrm{li}}{\mathrm{m}} - \left(\frac{\mathrm{di}}{2\mathrm{m}}\right)^2 \quad (2)$$

where C_i+N_c indicates the cluster with the node N_c and C_i-N_c indicates the cluster without the node N_c. This function $\boldsymbol{q}$ is derived by the Modularity function [4,5], l_i represents the number of the edges in the cluster C_i and d_i represents the sum of the degrees of the nodes in the cluster C_i. Each node of the set N_c moves from its original cluster C_o to the cluster C_i if $CM\ (C_i,N_c) > CM\ (C_o,N_c)$, i.e. this relocation produces an increment of the Modularity function (the sum of all cluster contributes **q**).
Once all the four steps are repeated for all network clusters, the *Moving List* is emptied. A part of the nodes in it is used to create a new cluster while the remaining nodes are reallocated among the existing clusters, each node is assigned to the cluster maximizing the CM function. The whole process is described in Figure 3.
When the steady state is reached, i.e. when the partition resulted from the last iteration *i+1* is equal or almost equal to the previous partition (iteration *i)*, the process is stopped and the final partition is provided as result. The MTopGO output consists of the final partition and the set of GO terms associated to each cluster of the final partition, which identifies the functional modules (Figure 4 shows an example of MTGO output).

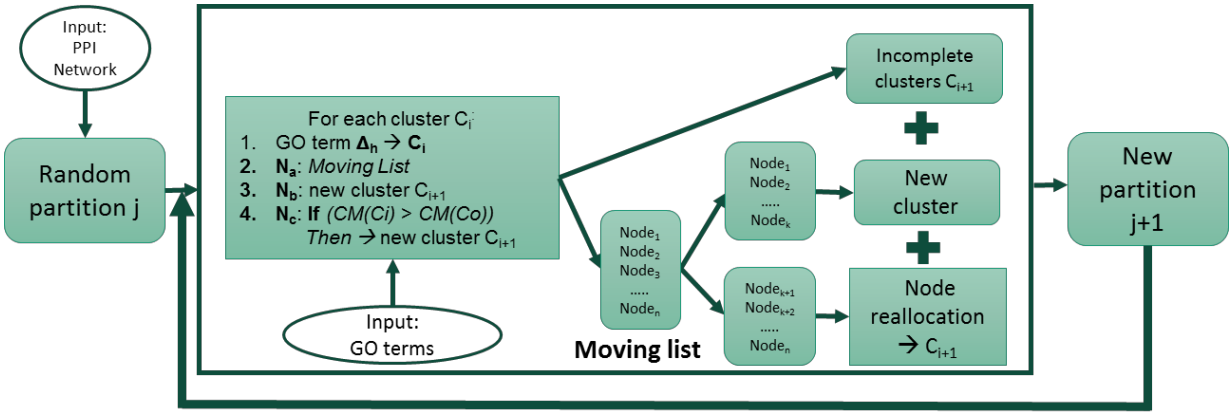

Figure 2 Workflow of MTopGO

Results

To evaluate the performance of the MTopGO algorithm, we compared it with another state-of-art algorithm for module identification in PPI network, DCAFP [9]. We decided to use DCAFP because, as MTopGO, it is based on both topological properties and GO information. The two algorithms applied on the PPI network from the Human organism, obtained downloading interactions from DIP database for human species [10]. This network is made of 2734 nodes and 4058 edges; the number of nodes covered by the GO terms is 2474. The list of GO terms for Human organism contains 7909 elements [8]. The predicted functional modules have been compared with a set of 1765 target complexes for Human organism, CORUM [11]. We analyzed the performance by six measures [12] showed in Table 1; F-measure is a combination of Recall and Precision, while Accuracy is a combination of Coverage and Positive Predictive Value. F-measure and Accuracy are two independent metrics used to evaluate the agreement between predicted and target complexes in terms of overlapping.

	Precision	Recall	F-measure	Coverage	Positive Predictive Value	Accuracy
MTopGO	0.101302	0.115014	0.107723739	0.481055	0.047131556	0.150575063
DCAFP	0.171806	0.026629	0.046110888	0.139808	0.143040823	0.141415307

Table 1 Results of MTopGO and DCAFP.

MTopGO shows better performance than DCAFP on 4/6 measures., In particular, it provides a better general quality. In fact, the F-measure is more than doubled (0.1 vs 0.04), and the Coverage more than tripled (0.481 vs. 0.139). However, DCAFP shows a better Precision, leading to a higher Positive Predictive Value. In fact, both Precision and PPV are metrics used to evaluate the proportion of predicted complexes but they work in a slightly different way. Precision evaluates the predicted complexes on the basis of a neighborhood affinity score between the predicted complex and real complex; while PPV on the basis of the number of proteins in common between the predicted complex and real complex [12].
Figure 4 shows an example of MTopGO applied to a human PPI Network built with String [6]. In this case, the network was built using as seed 54 genes muted in acute myeloid leukemia [7] and retrieving PPIs from String database, both known interactions (curated databases and experimentally determined) and predicted interactions (gene fusion, gene neighborhood, gene co-occurrence). Some other interacting nodes were added to the seeds

from String database to reach a final network of 78 nodes and 545 edges. To find the functional modules we retained only the GO terms related to the Human organism and tagged with Experimental evidence and/or computational analysis evidence Score (7909) [8]. To evaluate the MTopGO ability to detect a set of GO terms able to describe the network in terms of biological functions, the Fisher's exact test has been used to compute a p-value for each module and its corresponding GO term; the found p-values are all significant, under the 0.05 threshold (see Table 2).

	GO term	Description	P-value
1	GO:0005737	Cytoplasm	0.005033703
2	GO:0044822	RNA Binding	1.12E-12
3	GO:0005654	Nucleoplasm	0.000229789
4	GO:0003700	Transcription factor activity	0.003453057
5	GO:0005689	U12-type complex	6.17E-08

Table 2 The p-values computed by Fisher's exact test of the GO terms attached by MTopGO to each module.

Conclusions

MTopGo is a novel algorithm of module identification for PPI Network analysis, it is designed to consider two key aspects of these models, the topological properties of the network and the *apriori* knowledge about the proteins involved, represented by GO annotations. The final output provides both a PPI network partition and a set of GO terms describing the biological mechanisms involved. This represents a starting point for the model analysis and interpretation by biologists. MTopGo is therefore not just a clustering algorithm but also a tool to automatically analyze the biological phenomenon described by a PPI network and guide experts' research providing clinically interpretable results.

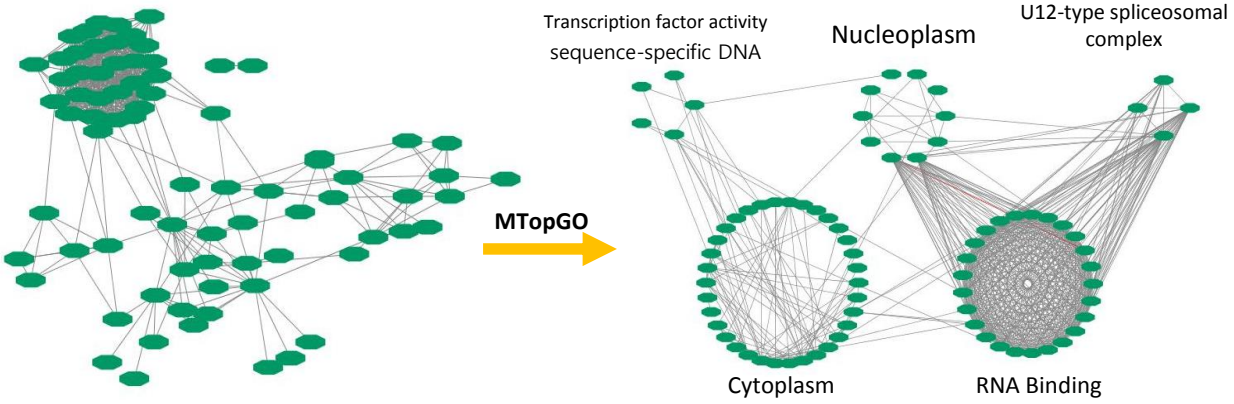

Figura 4 Example of MTopGO applied to a Human PPI network. The algorithm produces a partition of 5 clusters, each one tagged with a specific GO term.

References

1. Grindrod, Peter, and Milla Kibble. 2004. Review of uses of network and graph theory concepts within proteomics. Expert review of proteomics 1.2: 229-238.
2. Vella, D., Zoppis, I., Mauri, G., Mauri, P., & Di Silvestre, D. 2017. From protein-protein interactions to protein coexpression networks: a new perspective to evaluate large-scale proteomic data. EURASIP Journal on Bioinformatics and Systems Biology 2017(1):6.
3. Maayan, Avi. 2008. Network integration and graph analysis in mammalian molecular systems biology. IET systems biology 2.5: 206-221.
4. M.E.J. Newman, M. Girvan. 2004. Finding and evaluating community structure in networks, Physics Review
5. Fortunato, Santo. 2010. Community detection in graphs. *Physics reports* 486.3: 75-174.

6. Szklarczyk D, Morris JH, Cook H, Kuhn M, Wyder S, Simonovic M, Santos A, Doncheva NT, Roth A, Bork P, Jensen LJ, von Mering C. 2017. The STRING database in 2017: quality-controlled protein-protein association networks, made broadly accessible. Nucleic Acids Res. Jan; 45:D362-68.
7. Papaemmanuil, E., Gerstung, M., Bullinger, L., Gaidzik, V. I., Paschka, P., Roberts, N. D., & Gundem, G. 2016. Genomic classification and prognosis in acute myeloid leukemia. *New England Journal of Medicine* 374.23: 2209-2221.
8. http://geneontology.org/page/download-annotations
9. Hu, Lun, and Keith CC Chan. 2015. A density-based clustering approach for identifying overlapping protein complexes with functional preferences. BMC bioinformatics 16.1: 174.
10. Xenarios I, Salwnski L, Duan XJ, Higney P, Kim S, Eisenberg D. 2002. DIP, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. Nucleic Acids Res. 30(1):3035.
11. Ruepp A, Waegele B, Lechner M, Brauner B, Dunger-Kaltenbach I, Fobo G, Frishman G, Montrone C, Mewes HW. 2010. CORUM: the comprehensive resource of mammalian protein complexes2009. Nucleic Acids Res. 38(Database issue): D497-501. 19884131
12. Li, X., Wu, M., Kwoh, C. K., & Ng, S. K. 2010. Computational approaches for detecting protein complexes from protein interaction networks: a survey. *BMC genomics* 11.1: S3.