Stability Analysis of MTopGO for Module Identification in PPI Networks

Danila Vella¹, Allan Tucker², Riccardo Bellazzi¹,³
¹Clinical Scientific Institute Maugeri, Pavia, Italy ²Dept Computer Science, Brunel University London
³University of Pavia, Pavia, Italy (e-mail: danila.vella@icsmaugeri.it)

Introduction
MTopGo is a novel algorithm of module identification for PPI Network analysis [1], 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. Understanding and accounting for the stability of the clusters with respect to the presence of noise and uncertainty in the data is an important factor when evaluating an algorithm specific for PPI Networks [2]. In fact, the input graphs are obtained from high-throughput methods (e.g. yeast-two-hybrid, etc.) which are generally noisy with high false positive and false negative rates [3]. Moreover, many clustering algorithms rely on a random component, thus stability of the results across different runs is considered to be an asset of the algorithm [4]. As MTopGo is a non-deterministic algorithm, two different stability analyses have been executed to evaluate its performance. Firstly, one to evaluate the stability of the result over many runs starting from a same input, to consider the range of variability introduced by the random components of the algorithm; secondly, one to evaluate the robustness of the output clusters when the input is affected by noise and uncertainty.

Methods
MTopGO output consists of a network partition and a set of GO terms associated to each cluster of the partition, describing their biological role. Two different global functions are used to evaluate the final result: modularity (Q) [5] and Quality GO (QGO). Q evaluates the global quality of the partition; it ranges from -1 to 1, assuming positive values if there are more links within clusters than expected at random, and negative otherwise. QGO evaluates the agreement between the set of clusters and the set of GO terms selected, ranging from 0 to 1.

To evaluate the stability of the result over many runs starting from a same input, MTopGO has been executed 500 times on a PPI Network from human proteins [4]. While, to investigate the robustness of MTopGO on noisy PPI networks, the same network has been used to create eleven perturbed networks (keeping the same node set but changing the edges). In particular, combining three different percentages (0% - 5% - 10%) of random edge removal and addition, eight different networks have been obtained. Three additional networks have been obtained to explore more dramatic perturbations; two of them by removing and adding 50% of edges and the last one totally random (100% removal and 100% addition). Each network has been processed one hundred times.

Results
MTopGO stability results. In Figure 1 (left) the values of Q against QGO are shown for all 500 clusterings; it can be observed that the QGO values are more stable than Q values. It is interesting to note that about 95% of the total results can be identified in two main zones of high density, Zone 1 and Zone 2, suggesting that the Zone 1 correspond to a local Q maximum, while the Zone 2 corresponds to higher Q maximum. As regards the remaining 5% of the results, a third zone (Zone 3) of lower density can be observed, represented by points set along a diagonal line. Zone 3 suggests a negative correlation between Q and QGO, confirmed by a Pearson Correlation value of -0.78. In Figure 1 (right) we explore the use of Consensus Clustering (CC) [6] to remove lower quality clusterings. CC allows to compute a new clustering by maximizing agreement over all the 500 repeated clusterings. For each clustering, the Weighted Kappa (WK) measure [7] has been computed to measure the agreement with the CC found. The clusterings with bad WK value (less than a selected threshold 0.6) have been highlighted in Figure 3 (right) in red. It is interesting that the clusterings with low CC agreement are included in the lowest density area, Zone 3. As a result, the generation of CC can be used to filter out the less stable clusterings to ensure results that are closer to the more preferred Zone 1 and Zone 2.

Perturbed network results. Figure 2 shows the boxplots for the Q distributions for each of the perturbed networks. It can be observed in the first eight networks the modularity is quite preserved. The Q distributions clearly show the MTopGO algorithm is more robust against the edge removal than the edge addition. In fact, a Q decreasing can be observed when the edge addition percentage increases while the edge removal percentage is constant. While, the high similarity between the Q distributions for the two 50% edge-
addition and edge-removal networks, may be due to a high percentage of error introduced, causing a loss of the modular structure of the two networks. As expected, the random network shows the lower Q values as the modular structure is completely damaged. Interestingly, Fortunato et al. studying the modularity limits find that both for a random and scale-free graph the expected maximum modularity increase when the graphs get sparser, i.e. the edge number decrease [5]. As the PPI networks often shown scale-free properties, this could be the reason behind the fact that the networks 0A-5R and 0A-10R, those with higher number of removed edges, show the maximum values of modularity.

Conclusions

In this study, we have explored the stability of the MTopGO algorithm for detecting modules in PPI networks. Results demonstrated that the QGO variation was small from different repeated runs of MTopGO, whilst the Q variation was relatively high (due to the variation in modularity). The use of Consensus Clustering helped to identify reasonably good module allocations as well as filter out bad. The network perturbation results showed that MTopGO was more stable in case of false negative edges than false positive edges (adding false edges was more damaging than removing existing links).

References

5. Santo Fortunato. “Community detection in graphs”. In: Physics reports 486.3 (2010)