Molecular Network Analysis

**MONET**: a toolbox integrating top-performing methods for network modularisation

Mattia Tomasoni\(^1,2\)\(^*\), Sergio Gómez\(^3\), Jake Crawford\(^4,5\), WeiJia Zhang\(^6\), Sarvenaz Choobdar\(^1,2\), Daniel Marbach\(^1,2,7\) and Sven Bergmann\(^1,2,8\)\(^*\)

\(^1\)Department of Computational Biology, University of Lausanne, Lausanne, Switzerland
\(^2\)Swiss Institute of Bioinformatics, Lausanne, Switzerland
\(^3\)Departament d’Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Tarragona, Catalonia, Spain
\(^4\)Department of Computer Science, Tufts University, MA, USA
\(^5\)Graduate Group in Genomics and Computational Biology, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA
\(^6\)School of Information Technology and Mathematical Sciences, University of South Australia, Adelaide, Australia
\(^7\)Present address: Roche Pharma Research and Early Development, Pharmaceutical Sciences, Roche Innovation Center Basel, Hoffmann-La Roche Ltd, 4070 Basel, Switzerland
\(^8\)Department of Integrative Biomedical Sciences, University of Cape Town, Cape Town, South Africa

\(^*\)To whom correspondence should be addressed.

Received on 12/12/2019; revised on 03/09/2019; accepted on 31/03/2020

**Abstract**

**Summary:** We define a disease module as a partition of a molecular network whose components are jointly associated with one or several diseases or risk factors thereof. Identification of such modules, across different types of networks, has great potential for elucidating disease mechanisms and establishing new powerful bio-markers. To this end, we launched the “Disease Module Identification (DMI) DREAM Challenge”, a community effort to build and evaluate unsupervised molecular network modularisation algorithms (Choobdar, 2019). Here we present MONET, a toolbox providing easy and unified access to the three top-performing methods from the DMI DREAM Challenge for the bioinformatics community.

**Availability and Implementation:** MONET is a command line tool for Linux, based on Docker and Singularity virtualization technologies; the core algorithms were written in R, Python, Ada and C++. It is freely available for download at https://github.com/BergmannLab/MONET.git

**Contact:** mattia.tomasoni@unil.ch (MT); sven.bergmann@unil.ch (SB)

**Supplementary information:** Supplementary data are available at Bioinformatics online.

1 Introduction

Gene networks, such as protein interaction, signalling, gene co-expression and homology networks, provide scaffolds of linked genes. Sub-networks, or modules, include genes normally acting in concert but whose joint function may be disrupted, if any of its members is missing, or disregulated. For Disease Modules this disruption can lead to a disease phenotype. The identification of such modules is therefore useful for elucidating disease mechanisms and establishing new bio-markers and potential therapeutic targets. Yet, which methods work best to extract such modules from different types of networks is not well understood. This prompted us to initiate the “Disease Module Identification (DMI) DREAM Challenge” (Choobdar, 2019), providing an unbiased and critical assessment of 75 contributed module identification methods. Our method evaluation used summary statistics from more than 200 disease relevant Genome-wide Association Studies (GWAS) in conjunction with our Pascal tool (Lamparter et al., 2016), avoiding the bias of using annotated pathways.

The top-performing methods implemented novel algorithms that advanced the state of the art, clearly outperforming off-the-shelf tools. We therefore decided to make the top three methods available for the bioinformatics community in a single user-friendly package: MONET is a command line tool based on Docker and Singularity virtualization technologies, automatically installing the tool with all its dependencies inside a container, avoiding time-consuming and error-prone manual installations of computing environments and libraries. All computations then take place in this sandbox environment and once the output is ready, all resources can be fully released bringing the user’s machine back to its original state.
2 Methods and implementation

While our challenge was able to establish Kernel Clustering Optimization using the “Diffusion State Distance” metric by Cao et al. (2014) (hereafter K1) as the overall winner, there were several strong competitors using entirely different approaches for the network modularisation. Importantly, we observed that no single method was superior on all network types and that Disease Modules identified by different methods were often complementary (Choobdar, 2019).

2.1 K1: Top method using kernel clustering

K1 is based on the “Diffusion State Distance” (DSD), a novel graph metric which is built on the premise that paths through low-degree nodes are stronger indicators of functional similarity than paths that traverse high-degree nodes by Cao et al. (2014). The DSD metric is used to define a pairwise distance matrix between all nodes, on which a spectral clustering optimization is performed. Each point represents the number of non-overlapping clusters. Importantly, we observed that no single method was superior on all network types and that Disease Modules identified by different methods were often complementary (Choobdar, 2019).

2.2 M1: Top method using modularity optimization

M1 employs an original technique named Multiresolution introduced by Arenas et al. (2008) to explore all topological scales at which modules may be found. The novelty of this approach relies on the introduction of a parameter, called resistance, which controls the aversion of nodes to form modules. Modularity (Newman and Girvan, 2004; Arenas et al., 2007) is optimized using an ensemble of algorithms: Extremal optimization (Duch and Arenas, 2005), Spectral optimization (Newman, 2006), Fast algorithm (Newman, 2004), Tabu search (Arenas et al., 2008), and fine-tuning by iterative repositioning of individual nodes in adjacent modules. BLOG: https://www.synapse.org/#/Synapse:syn7352969/wiki/407384

2.3 R1: Top method using random walk

R1 is based on a variant of Markov Cluster Algorithm known as balanced Multi-layer Regularized Markov Cluster Algorithm (bMLRMC) (Satuluri et al., 2010) which scales well to large graphs and minimizes the number of oversized clusters. First, a pre-processing step is applied so that edges with low weights are discarded and all remaining edges are scaled to integer values. Then, bMLRMC is applied iteratively on modules of size greater than a user-defined threshold. BLOG: https://www.synapse.org/#/Synapse:syn7266597/wiki/406659

3 Performance

Figure 1 illustrates the performance of the MONET algorithms on simulated graphs with planted community structure, generated using the class of benchmark graphs proposed by Lancichinetti et al. (2008). Modularisation performance is measured using Normalized Mutual Information (NMI). Experiments were carried out on regular desktop hardware. All algorithms show great performance on the simulated benchmark. K1, winner of the DREAM Challenge, requires the most computational resources; this is in accordance with performance shown during the Challenge (it took a day to complete and required around 8GB RAM). M1, the second runner-up, has extremely low RAM requirements and displays excellent performance on the simulated benchmark (even superior to K1, especially in case of extremely high fraction of inter-module edges). R1, the third runner-up, has the most parameters, but it is very fast (it completed the Challenge in under an hour). Please, refer to the Supplementary Information for details about the execution time.

4 Installation and usage

MONET is extremely simple to install/uninstall and run. The only requirement is having installed either Docker (Merkel, 2014) or Singularity (Kurtzer et al., 2017). For detailed instructions and information about usage and I/O formats, please refer to the README file on the github repository.

$ git clone https://github.com/BergmannLab/MONET.git
$ cd MONET && ./install.sh
$ git clone https://github.com/BergmannLab/MONET.git
$ cd MONET && ./install.sh
$ monet --method=M1 --container=docker

Funding

This work was supported by the Swiss National Science Foundation grant no. 310030_152724/1.

References

Arenas, A., Duch, J., Fernández, A., and Gómez, S. (2007). Size reduction of complex networks preserving modularity. New Journal of Physics, 9(6), 176.
Arenas, A., Fernández, A., and Gómez, S. (2008). Analysis of the structure of complex networks at different resolution levels. New Journal of Physics, 10(5), 053039.
Cao, M., Pietras, C. M., Feng, X., Doroschak, K. J., Schulliner, T., Park, J., Zhang, H., Coren, L. J., and Hesscott, B. J. (2014). New directions for diffusion-based network prediction of protein function: incorporating pathways with confidence. Bioinformatics, 30(12), i219-i227.
Choobdar, S. et al. (2019). Assessment of network module identification across complex diseases. Nature Methods, 16(9), 843-852.
Duch, J. and Arenas, A. (2005). Community detection in complex networks using extremal optimization. Physical Review E, 72(2), 027104.
Kurtzer, G. M., Sochat, V., and Bauer, M. W. (2017). Singularity: Scientific containers for mobility of compute. PLOS ONE, 12(5), 1-20.
Lamparter, D., Marbach, D., Raveci, R., Katalik, Z., and Bengmann, S. (2016). Fast and rigourous computation of gene and pathway scores from snp-based summary statistics. PLOS Computational Biology, 12(1), 1-20.
Lancichinetti, A., Fortunato, S., and Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. Phys. Rev. E, 78, 046110.
Merkel, D. (2014). Docker: Lightweight linux containers for consistent development and deployment. Linux J, 2014(239).
Newman, M. E. (2004). Fast algorithm for detecting community structure in networks. Physical Review E, 69(6), 066133.
Newman, M. E. (2006). Modularity and community structure in networks. Proceedings of the National Academy of Sciences USA, 103(23), 8577-8582.
Newman, M. E. and Girvan, M. (2004). Finding and evaluating community structure in networks. Physical Review E, 69(2), 026113.
Satuluri, V., Parthasarathy, S., and Ucar, D. (2010). Markov clustering of protein interaction networks with improved balance and scalability. In Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology, BCB '10, pages 247-256, New York, NY, USA.