BIOINFORMATICS APPLICATIONS NOTE
Vol. 30 no. 19 2014, pages 2834–2836
doi:10.1093/bioinformatics/btu384

Systems biology
Advance Access publication June 12, 2014

NetBioV: an R package for visualizing large network data in biology and medicine

Shailesh Tripathi[1], Matthias Dehmer[2,3] and Frank Emmert-Streib[1,*]

[1]Computational Biology and Machine Learning Laboratory, Center for Cancer Research and Cell Biology, School of Medicine, Dentistry and Biomedical Sciences, Queen's University Belfast, Belfast BT9 7BL, UK, [2]Institute for Bioinformatics and Translational Research, UMIT, 6060, Hall in Tyrol, Austria and [3]Department of Computer Science, Universität der Bundeswehr München, 85577 Neubiberg, Germany

Associate Editor: Alfonso Valencia

ABSTRACT

Summary: NetBioV (Network Biology Visualization) is an R package that allows the visualization of large network data in biology and medicine. The purpose of NetBioV is to enable an organized and reproducible visualization of networks by emphasizing or highlighting specific structural properties that are of biological relevance.

Availability and implementation: NetBioV is freely available for academic use. The package has been tested for R 2.14.2 under Linux, Windows and Mac OS X. It is available from Bioconductor.

Contact: v@bio-complexity.com

Supplementary information: Supplementary data are available at Bioinformatics online.

Received on May 28, 2014; revised on May 14, 2014; accepted on June 9, 2014

1 INTRODUCTION

Modern research in the biological and biomedical sciences is driven by technological progress that routinely allows interrogating genetic or molecular entities on a genomic scale. For the analysis and interpretation of high-throughput genomics experiments, frequently networks are used because they enable a systems approach. However, the graphical representation of networks for reasons of their exploratory analysis, interpretation or merely as a visualization is far from being trivial, especially for large networks containing hundreds or thousands of genes. For this reason, several software visualization tools have been developed. In the Supplementary File, we compared our visualization software NetBioV with Cytoscape, VisANT and yEd (Hu *et al*., 2013; Shannon *et al*., 2003). In summary, NetBioV is fully integrated into R, the gold standard in the computational biology community, and hence can use all of its functionality for user-specific expansions. Furthermore, NetBioV provides (i) global, (ii) module, (iii) information flow and (iv) hierarchical layout styles, discussed in the following, not available in this form in other packages. Importantly, the provided layout styles run efficiently, even for large networks.

2 METHODS

The graphical representation of NetBioV is based on the igraph package (Csardi and Nepusz, 2006) by using elementary visualization capabilities, thereof allowing the composition of advanced graph layouts. NetBioV provides four principle types of layout styles—(i) global, (ii) modular, (iii) information flow and (iv) hierarchical layouts. These can be used either separately or combined with each other, in a nested way. The reason for this conceptual subdivision is that, first, networks contain a multitude of information that cannot be visualized assuming *just one* specific perspective. Second, it is generally acknowledged that biological gene networks have a hierarchical and modular organizational structure (Barabási and Oltvai, 2004). For this reason, selecting a layout from one of the four main categories allows to highlight a certain biological aspect of a network.

2.1 Global layout styles

NetBioV provides six different style functions for generating global layouts: *mst.plot*, *mst.plot.mod*, *plot.NetworkSpherical.startSet*, *plot.NetworkSpherical* and *plot.spiral.graph*. The first two functions extract, initially, a minimum spanning tree (MST) from a network and, then, use a forced-based algorithm (either Fruchterman–Reingold or Kamada–Kawai) for the MST to generate the coordinates for the nodes. Finally, all remaining edges are added. In Figure 1A, we show an example using the *mst.plot.mod* function. White edges correspond to the MST, all other edges are in shades of orange and a darker color corresponds to more distant nodes. Furthermore, the color of the nodes reflects the expression of the genes (blue low, red high). This approach is different to all other network visualization software because by using an MST in the first step one gains a considerable advantage in the computing time it takes to calculate the coordinates of the nodes in large networks (for a numerical comparison see Supplementary File Section S3.1 and S4.5). The function *plot.NetworkHubView* provides a hub-view of a network placing the nodes according to their degree in a circular order, and the remaining three functions produce a star-, spherical- and spiral-view of a network (see Supplementary File).

2.2 Modular layout styles

NetBioV offers four different style functions for modular layouts: *splitg.mst, plot.abstract.module, plot.abstract.nodes* and *plot.modules.* All of these functions require as an input, information about the partitioning of the nodes in the form of modules. Modules can be identified either with module detecting algorithms (included in NetBioV) or from biomedical databases, e.g. from gene ontology (GO). For instance, the function *split.mst* expands the global style function *mst.plot* to the module level maintaining the same computational advantages using an MST. The function *plot.modules* allows the module-wise specification of the

*To whom correspondence should be addressed.

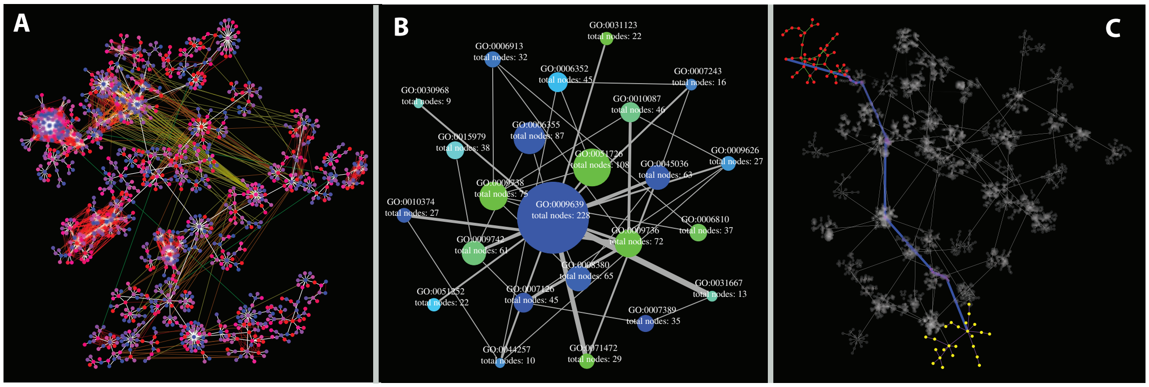

Fig. 1. (**A**) Global network view. (**B**) Abstract modular view. (**C**) Information flow between modules. A and B show the protein network of *A.thaliana* containing 1212 proteins and 2574 interactions. C is the gene regulatory network of B-cell lymphoma with 2498 genes and 2654 interactions

layout style representing each module, which means that it is possible to select a different layout style for each module in a network. The function *plot.abstract.nodes* provides an abstraction of a network, representing each module by a single node, and edges between modules are collapsed into a single edge. The size of the nodes and edges can be chosen to be proportional to the total number of nodes in a module and the edges between modules, respectively (see Fig. 1B). The resulting visualization is module-centric, and hence allows masking the connectivity on the gene level.

2.3 Information flow and hierarchical layout styles

An information flow layout style highlights the shortest paths between two or more modules, or gene sets. The available functions are *plot.modules* and *level.plot*. The available function plot.modules used in Figure 1C shows the information flow between two modules. In this view, the shortest path from an initial set of nodes (red) to a destination set of nodes (yellow) is color highlighted. A large variety of features can be user-specified to enhance a desired visualization effect.

Finally, we provide 2 functions (*level.plot, level.plot.spread*) allowing a hierarchal representation of networks. These functions assume an initial node set, $\mathcal{N}_i$, and plot their adjacent neighbors (measured by the Dijkstra distance) of these nodes iteratively on different levels considering the directionality of the edges (if available). For a directed network, ‘level.plot.spread’ shows all generations of ancestors and offsprings of $\mathcal{N}_i$.

2.4 Interfacing with R and object-oriented structure

For the presented examples in Figure 1, we used various R packages for (i) identifying the modules of the protein–protein interaction (PPI) network and (ii) obtaining information about the enrichment of the modules for GO terms. Owing to the integration of NetBioV into R, interfacing with the large package repositories CRAN or Bioconductor is naturally enabled allowing to use a multitude of different features that may be used as part of the visualization of the network.

Aside from this, NetBioV has an object-oriented structure allowing to combine different types of layouts seamlessly with each other. For example, it is possible to specify for each module in a network a different layout style or color scheme. That means, each of the global layout styles can be used independently as a layout style for a module. Figure 1C shows an example, where a different edge and node color are chosen for the two modules and the connecting path. For the visualization of the activity of genes or proteins, we provide a function that allows to map the expression levels onto the network by representing activity levels by different node colors to identify easily, e.g. differentially expressed genes (see Fig. 1A). Importantly, the final output of a plot can be saved as a vector graphics format (*eps* or *pdf*).

3 EXAMPLES

Figure 1 shows three different network visualizations, two for the PPI network of *Arabidopsis thaliana* (Breitkreutz *et al.*, 2008) (Fig. 1A and B) and one for the gene regulatory network of B-cell lymphoma, inferred with BC3NET (de Matos Simoes and Emmert-Streib, 2012) (Fig. 1C). Figure 1C can be reproduced by the following code. In the Supplementary File, we provide additional examples.

```
####### Figure 1 C ######
library("igraph")
library("netbiov")
data("gnet_bcell")
data("modules_bcell")
cl <- rgb(r=0.5, g=0.5, b=0.5, alpha=0.5)
sm <- rep(8, length(mod.list))
sm[c(23, 43)] <- 15
gparm <-  plot.modules(gnet, mod.list=mod.list, v.size=1.5,
layout.function=layout.graphopt,  nodeset=c(23,43), modules.color=cl,
path.col=c("blue", "green", "purple"), mod.lab=F, e.path.width=c(1,5),
lab.color="white", scale.module=sm, v.size.path=1.5, e.width=0.4)
```

4 CONCLUSION

NetBioV allows the efficient visualization of large biological networks by emphasizing important aspects of biological information processing, e.g. modularity, information flow or hierarchy. The organization of NetBioV is highly flexible enabling individualized network visualizations. Part of the functionality of NetBioV comes from its integration into R to borrow strength from existing package repositories and to ensure reproducible research by saving source code for the network visualization coordinates and parameters.

ACKNOWLEDGEMENT

M.D. thanks the Austrian Science Funds for supporting this work (project P26142).

Funding: ST is supported by a studentship from the National Institute of Immunology.

Conflicts of Interest: none declared.

REFERENCES

Barabási,A.L. and Oltvai,Z.N. (2004) Network biology: understanding the cell's functional organization. *Nat. Rev.*, **5**, 101–113.

Breitkreutz,B.J. *et al.* (2008) The BioGRID Interaction Database: 2008 update. *Nucleic Acids Res.*, **36** (**Suppl. 1**), D637–D640.

Csardi,G. and Nepusz,T. (2006) The igraph software package for complex network research. *InterJournal*, Complex Systems, 1695.

de Matos Simoes,R. and Emmert-Streib,F. (2012) Bagging statistical network inference from large-scale gene expression data. *PLoS One*, **7**, e33624.

Hu,Z. *et al.* (2013) Visant 4.0: integrative network platform to connect genes, drugs, diseases and therapies. *Nucleic Acids Res*, **41**, W225–W231.

Shannon,P. *et al.* (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.*, **13**, 2498–2504.