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

Shailesh Triapthi, Matthias Dehmer and Frank Emmert-Streib

1 Computational Biology and Machine Learning Laboratory, Center for Cancer Research and Cell Biology, Faculty of Medicine, Health and Life Sciences, Queen’s University Belfast, 97 Lisburn Road, Belfast BT9 7BL, UK; 2 Institute for Bioinformatics and Translational Research, UMIT, Eduard Wallnoefer Zentrum 1, 6060, Hall in Tyrol, Austria; 3 Department of Computer Science, Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany

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: 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 (after the review). (Preliminary link at www.bio-complexity.com/netbiov_2.6.tar.gz)

Contact: v@bio-complexity.com

1 INTRODUCTION

Modern research in the biological and biomedical sciences is driven by technological progress that routinely allows to interrogate genetic or molecular entities on a genomic-scale. For the analysis and interpretation of high-throughput genomics experiments, frequently, networks are utilized, 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 (Shannon et al., 2003; Hu et al., 2013). In summary, NetBioV is fully integrated into R, the gold standard in the computational biology community, and, hence, can utilize all of its functionality for user-specific expansions. Furthermore, NetBioV provides (1) global, (2) modular, (3) information flow and (4) hierarchical layouts. These can be either used 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 6 different style functions for generating global layouts: mst.plot, mst.plot.mod, plot.NetworkHubView, plot.NetworkSpherical, startSet, plot.NetworkSpherical, plot.spherical.graph. The first two functions extract, first, 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 Fig. 1 A, we show an example using the mst.plot.mod function. White edges correspond to the MST, all other edges are in shades of orange, whereas 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 a 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 Sec. 3.1 and Sec. 4.5). 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 4 different style functions for modular layouts: split.mst, plot.abstract.module, plot.abstract.nodes, 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 utilizing a minimum spanning tree. The function plot.modules allows the module-wise specification of the 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 their degree.
to the total number of nodes in a module respectively the edges between modules (see Fig. 1 B). The resulting visualization is module-centric and, hence, allows to 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.module and level.plot. Figure 1 C, 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 hierarchical representation of networks. These functions assume an initial node set, \( N_i \), 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 \( N_i \).

2.4 Interfacing with R and object oriented structure

For the presented examples in Fig. 1, we utilized various R packages for (a) identifying the modules of the PPI (protein-protein interaction) network and for (b) obtaining information about the enrichment of the modules for gene ontology (GO) terms. Due to the integration of NetBioV into R, interfacing with the large package repositories CRAN or Bioconductor is naturally enabled allowing to utilize 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 1 C shows an example, where a different edge and node color is 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. 1 A). Importantly, the final output of a plot can be saved as a vector graphics format (eps or pdf).

3 EXAMPLES

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

### Figure 1 C

```r
library("igraph")
library("netbiov")
data("modules_bcell")
si <- rgb(r=0, g=0, b=0.5, alpha=0.5)
sn <- rep(8, length(mod.list))
sm[23, 43] <- 15
sm <- rep(8, length(mod.list))
cl <- rgb(r=0.5, g=0.5, b=0.5, alpha=0.5)
gparm <- plot.modules(gnet, mod.list=mod.list, v.size=1.5,
layout.function=layout.graphopt, nodeset=c(23, 43), modules.color=cl,
pch.color=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.

Funding: ST is supported by a studentship from the National Institute of Immunology. MD thanks the Austrian Science Funds for supporting this work (project P22029-N13).

REFERENCES


