Systems biology

NAViGaTOR: Network Analysis, Visualization and Graphing Toronto

Kevin R. Brown[1], David Otasek[1], Muhammad Ali[1], Michael J. McGuffin[2], Wing Xie[1], Baiju Devani[1], Ian Lawson van Toch[1] and Igor Jurisica[1,3,*]

[1]Division of Signaling Biology, Ontario Cancer Institute and the University Health Network, Toronto, ON,
[2]Department of Software and IT Engineering, École de technologie supérieure, Montreal, QC and
[3]Departments of Medical Biophysics and Computer Science, University of Toronto, Toronto, ON, USA

ABSTRACT

Summary: NAViGaTOR is a powerful graphing application for the 2D and 3D visualization of biological networks. NAViGaTOR includes a rich suite of visual mark-up tools for manual and automated annotation, fast and scalable layout algorithms and OpenGL hardware acceleration to facilitate the visualization of large graphs. Publication-quality images can be rendered through SVG graphics export. NAViGaTOR supports community-developed data formats (PSI-XML, BioPax and GML), is platform-independent and is extensible through a plug-in architecture.

Availability: NAViGaTOR is freely available to the research community from http://ophid.utoronto.ca/navigator/. Installers and documentation are provided for 32- and 64-bit Windows, Mac, Linux and Unix.

Contact: juris@ai.utoronto.ca

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

1 INTRODUCTION

The availability of protein–protein interaction (PPI) data is increasing rapidly through literature-derived databases (Bader *et al.*, 2003; Breitkreutz *et al.*, 2002; Hermjakob *et al.*, 2004a; Peri *et al.*, 2004; Xenarios *et al.*, 2000; Zanzoni *et al.*, 2002), high-throughput detection methods (Barrios-Rodiles *et al.*, 2005; Rual *et al.*, 2005) and computational predictions (Brown and Jurisica, 2005; Persico *et al.*, 2005). These data, collectively referred to as the interactome, are critical to our understanding of both normal cellular processes and disease mechanisms. Visualizing the interactome, along with integrating orthogonal data types, may aid in the understanding of cell function, help elucidate hidden relationships within the data and help prioritize functional studies.

Several biological graph visualization tools are currently available, implementing a diverse range of approaches and algorithms (Breitkreutz *et al.*, 2003; Chin *et al.*, 2008; Hu *et al.*, 2004; Ju and Han, 2003; Macpherson *et al.*, 2009; Paananen and Wong, 2009). Cytoscape (Shannon *et al.*, 2003), in particular, has been widely adopted by the biological community for its ease of use and extensibility through open source plug-in development.

---

*To whom correspondence should be addressed.

While many of these tools are effective and widely used, there are several critical areas where these applications require improvement (reviewed in Suderman and Hallett, 2007). Scalability is essential to visualize the tens of thousands of known PPI, which is a challenge for current layout algorithms and software. Biological graph drawing software must also be able to handle richly annotated data, including genomic and proteomic profiles, KEGG pathways (Kanehisa and Goto, 2000), Gene Ontology (GO) annotations, data in PSI-MI (Hermjakob *et al.*, 2004b) and BioPAX formats (http://www.biopax.org/), in addition to the vast quantity of microarray data that is currently available.

NAViGaTOR builds upon these earlier efforts, addressing known issues in existing software. NAViGaTOR uses a combination of hardware-based graphics acceleration and highly optimized layout algorithms to enable interactive visualization of large networks. It supports community-based data interchange formats, such as PSI-MI, BioPAX and GML, facilitating interoperability with existing software tools. Additionally, NAViGaTOR includes a rich suite of built-in analysis and visualization functions, which can be extended through an application programming interface (API). Here, we describe the implementation of NAViGaTOR, and highlight how this tool improves upon existing network visualization packages.

2 SOFTWARE

2.1 Implementation

NAViGaTOR has been implemented in Java (v1.6), providing platform-independence, and uses JOGL (https://jogl.dev.java.net/) to enable OpenGL hardware-accelerated graphics rendering. At present, the core code-base is closed-source to ensure stability, but future enhancements will extend the plug-in API to an OSGi-compliant (http://www.osgi.org/Main/HomePage) framework that enables community-driven extensibility.

2.2 Features

NAViGaTOR enables interactive visualization and analysis of complex graphs that are typical in systems biology studies. Graphs can be loaded from PSI-MI XML, BioPax, GML and tab-delimited text files, or through online databases such as I2D (http://ophid.utoronto.ca/i2d) and cPATH (http://cbio.mskcc.org/cpath/). Both
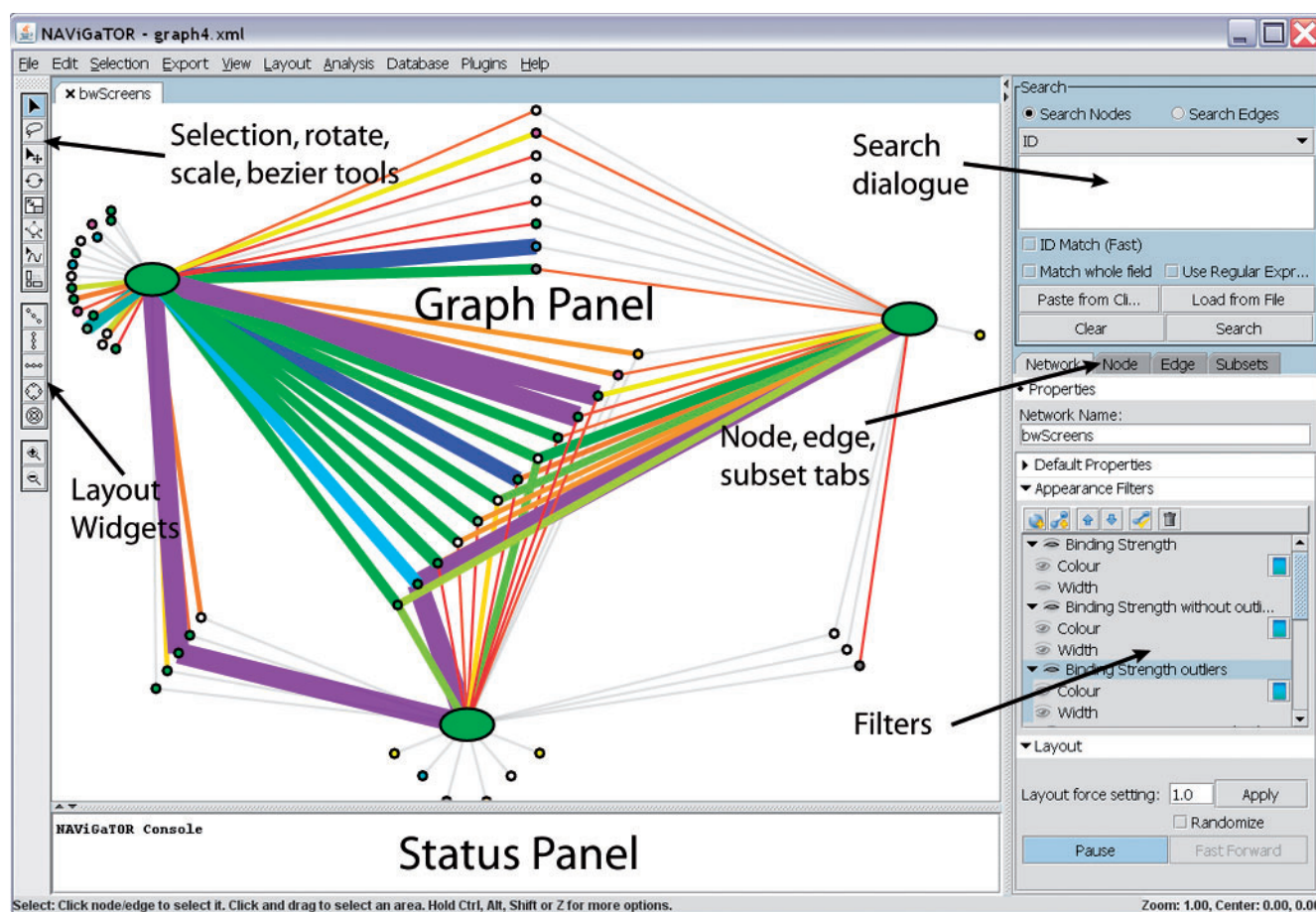
Fig. 1. Screen capture of the NAViGaTOR user interface. Labels indicate the various tools and descriptive regions of the interface. A graph is shown in the 'Graph Panel', with edges adjusted automatically by 'Edge Filters'. Filters can be used to automatically control visual attributes of both nodes and edges.

2D and 3D network views are fully interactive, allowing the user to manually manipulate the graph, or to use automated layouts such as circular, linear or concentric circular on subsets of nodes or entire graphs. A spreadsheet view supports selecting and deselecting nodes, edges and paths based on any attributes. Nodes and edges can be grouped into subsets, which can be collapsed or expanded to simplify views, or manipulated through set operations. Network analysis tools provide information about node and edge connectivity, shortest paths, identify hubs, cliques and articulation points and summarize network statistics. NAViGaTOR can also use a multi-threaded implementation to efficiently generate random networks for enrichment analyses. Fully annotated graphs can be exported to six different graphics formats, including PDF and SVG. In summary, NAViGaTOR provides a network analysis platform that is rich in the features essential to many biological applications, and yet is extensible through a plug-in interface to include additional features when required. See Figure 1 and the Supplementary Materials for examples of the NAViGaTOR interface and rendered networks.

2.3 Advances

NAViGaTOR's ability to handle larger datasets is facilitated through optimized layout algorithms, hardware-based graphics acceleration and a reduced memory footprint relative to other software. NAViGaTOR performs an initial layout using Graph Drawing with Intelligent Placement (GRIP; Gajer and Kobourov, 2002), which performs network layout in near linear time, and then continuously updates the layout of the graph using a multi-threaded grid-variant (Fruchterman and Reingold, 1991) of a force-directed layout algorithm. When benchmarked against the force-directed algorithms in Cytoscape and VisANT, NAViGaTOR consistently provided graphs rendered in significantly shorter time (Fig. 2; Supplementary Fig. 3.3). Only the yFiles Organic plug-in for Cytoscape rendered in similar time to NAViGaTOR, although the resulting graph was poorly optimized (compare Supplementary Fig. 3.5C to Supplementary Fig. 3.4C).

OpenGL enables NAViGaTOR to take advantage of hardware-based acceleration to render larger graphs in both 2D and 3D. Additionally, the data structures within NAViGaTOR were designed to maintain a small memory footprint in order to provide greater scalability for large datasets. When compared against Cytoscape and VisANT, NAViGaTOR had a memory footprint approximately half that of Cytoscape, although a 12–38% larger footprint than VisANT (Supplementary Fig. 5.1).

The NAViGaTOR user interface includes unique tools to help simplify the 'hairball', which is a common challenge in many PPI
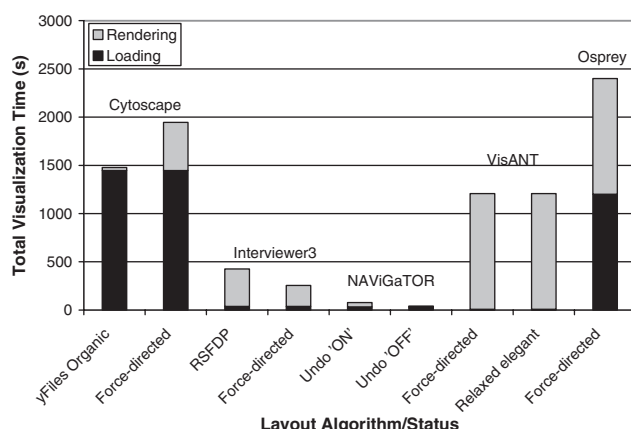
Fig. 2. Performance comparisons between applications. The Reactome *Caenorhabditis elegans* BioPax file was used to test the performance of several graph visualization applications in loading and visualizing the graph. Only Cytoscape and NAViGaTOR were able to load the BioPax file directly; Interviewer3 required a GML export from NAViGaTOR, VisANT required a PSI-MI XML v1.0 file, and Osprey required a tab-delimited text file. Stacked bars were used to show the cumulative loading and rendering time, or the total time to view a graph.

networks. Alpha blending is a technique to deemphasize unimportant areas of the network and focus on important areas by 'fading out' selected nodes and edges. Automated 'filters' let users map node and edge properties, such as color, size, shape and opacity to any numeric or text attribute. For instance, nodes can be scaled by degree or betweenness centrality, and colors can be mapped to GO ontology categories. Rectangle and lasso selection, and the unique ability to (de)select a connected neighborhood of nodes by dragging out its radius in edges, allow users to easily select specific subsets of nodes and edges, while other tools allow the selected subset to be rotated, scaled or laid out along a line or circle. Combined with pan/zoom navigation, users can quickly explore or simplify complicated networks.

3 FUTURE DEVELOPMENT

NAViGaTOR has evolved from an in-house visualization tool to a more versatile, comprehensive platform. While the current version of NAViGaTOR includes a plug-in API, NAViGaTOR 3.0 will adopt a more formal open plug-in interface compliant with the OSGi framework. This framework will allow for community-driven development through small, tightly coupled bundles while protecting the core code-base of the application. NAViGaTOR will also serve as a platform to explore novel ways for biologists to interact with graphs, as well as new ways to encode and display information in biological networks.

ACKNOWLEDGEMENTS

REFERENCES

Bader,G.D. *et al.* (2003) BIND: the biomolecular interaction network database. *Nucleic Acids Res.*, **31**, 248–250.

Barrios-Rodiles,M. *et al.* (2005) High-throughput mapping of a dynamic signaling network in mammalian cells. *Science*, **307**, 1621–1625.

Breitkreutz,B.J. *et al.* (2002) The GRID: the general repository for interaction datasets. *Genome Biol.*, **3**, PREPRINT0013.

Breitkreutz,B.J. *et al.* (2003) Osprey: a network visualization system. *Genome Biol.*, **4**, R22.

Brown,K.R. and Jurisica,I. (2005) Online predicted human interaction database. *Bioinformatics*, **21**, 2076–2082.

Chin,G., Jr *et al.* (2008) BioGraphE: high-performance bionetwork analysis using the Biological Graph Environment. *BMC Bioinformatics*, **9** (Suppl. 6), S6.

Fruchterman,T.M.J. and Reingold,E.M. (1991) Graph drawing by force-directed placement. *Software Pract. Exper.*, **21**, 1129–1164.

Gajer,P. and Kobourov,S.G. (2002) GRIP: graph drawing with intelligent placement. *J. Graph Algorithms Appl.*, **6**, 203–224.

Hermjakob,H. *et al.* (2004a) IntAct: an open source molecular interaction database. *Nucleic Acids Res.*, **32**, D452–D455.

Hermjakob,H. *et al.* (2004b) The HUPO PSI's molecular interaction format–a community standard for the representation of protein interaction data. *Nat. Biotechnol.*, **22**, 177–183.

Hu,Z. *et al.* (2004) VisANT: an online visualization and analysis tool for biological interaction data. *BMC Bioinformatics*, **5**, 17.

Ju,B.H. and Han,K. (2003) Complexity management in visualizing protein interaction networks. *Bioinformatics*, **19** (Suppl. 1), i177–i179.

Kanehisa,M. and Goto,S. (2000) KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, **28**, 27–30.

Macpherson,J.I. *et al.* (2009) JNets: exploring networks by integrating annotation. *BMC Bioinformatics*, **10**, 95.

Paananen,J. and Wong,G. (2009) FORG3D: force-directed 3D graph editor for visualization of integrated genome scale data. *BMC Syst. Biol.*, **3**, 26.

Peri,S. *et al.* (2004) Human protein reference database as a discovery resource for proteomics. *Nucleic Acids Res.*, **32**, D497–D501.

Persico,M. *et al.* (2005) HomoMINT: an inferred human network based on orthology mapping of protein interactions discovered in model organisms. *BMC Bioinformatics*, **6** (Suppl. 4), S21.

Rual,J.F. *et al.* (2005) Towards a proteome-scale map of the human protein-protein interaction network. *Nature*, **437**, 1173–1178.

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

Suderman,M. and Hallett,M. (2007) Tools for visually exploring biological networks. *Bioinformatics*, **23**, 2651–2659.

Xenarios,I. *et al.* (2000) DIP: the database of interacting proteins. *Nucleic Acids Res.*, **28**, 289–291.

Zanzoni,A. *et al.* (2002) MINT: a Molecular INTeraction database. *FEBS Lett.*, **513**, 135–140.