

## Systems biology

## Tools for visually exploring biological networks

Matthew Suderman\* and Michael Hallett

McGill Centre for Bioinformatics, 3775 University Street, Montreal, QCH3A 2B4, Canada

Received on February 27, 2007; revised on June 4, 2007; accepted on August 3, 2007

Advance Access publication August 25, 2007

Associate Editor: Jonathan Wren

**ABSTRACT**

Many tools exist for visually exploring biological networks including well-known examples such as Cytoscape, VisANT, Pathway Studio and Patika. These systems play a key role in the development of integrative biology, systems biology and integrative bioinformatics. The trend in the development of these tools is to go beyond 'static' representations of cellular state, towards a more dynamic model of cellular processes through the incorporation of gene expression data, subcellular localization information and time-dependent behavior. We provide a comprehensive review of the relative advantages and disadvantages of existing systems with two goals in mind: to aid researchers in efficiently identifying the appropriate existing tools for data visualization; to describe the necessary and realistic goals for the next generation of visualization tools. In view of the first goal, we provide in the Supplementary Material a systematic comparison of more than 35 existing tools in terms of over 25 different features.

**Contact:** msuder@mcb.mcgill.ca

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

**1 INTRODUCTION**

Networks are used ubiquitously throughout biology to represent the relationships between genes and gene products. Perhaps due to their simple discrete nature and to their amenability to visual representation, biological networks allow for the most salient properties of complex systems to be highlighted in a succinct and powerful manner. Visualization of biological networks range from small-scale descriptions of specific metabolic, regulatory or signalling pathways appearing in countless journal articles, seminars and courses, to classic efforts such as the Roche Applied Science Biochemical Pathways (Michal, 1993) and the Biochemical Pathway Atlas (Michal, 1998) that attempt to provide broad maps of cellular organization. Whereas these latter efforts required extensive manual curation to iteratively refine the maps as new information became available, the advent of high-throughput genetic-based assays that permit (near) cell-wide exploration of relationships between genes and gene products has motivated the development of software systems capable of constructing visualizations of these datasets 'on the fly'. Assays such as genetic-based protein–protein interaction

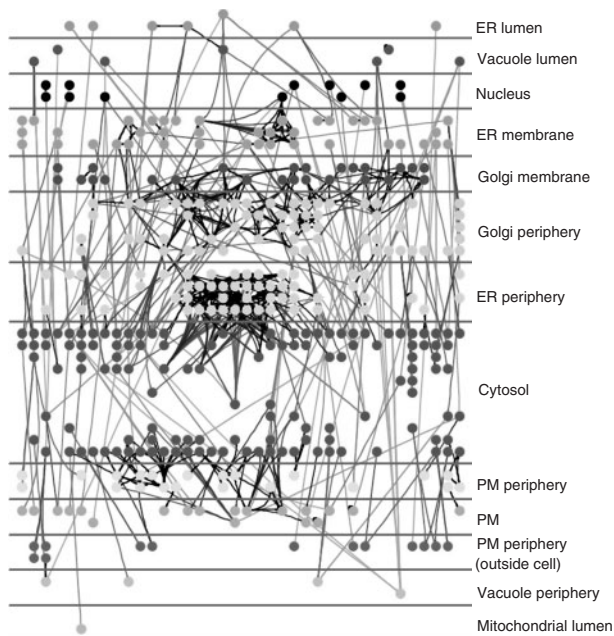
screens [e.g. (Fields and kyu Song, 1989; Rigaut *et al.*, 1999; Selbach and Mann, 2006)], RNAi-based genetic interaction screens (Echeverri and Perrimon, 2006; Fire *et al.*, 1998) gene expression (Lockhart *et al.*, 1996) and ChIP-Chip arrays (Blat and Kleckner, 1999; Ren *et al.*, 2000) are readily visualized when nodes represent genes, gene products or small molecules and, links (edges) between pairs of nodes can model metabolic events, protein–protein/protein–nucleotide interactions, regulatory relationships or signaling pathways.

Such simple networks, that capture a static 'snapshot' of cellular state, have been shown capable of yielding insight into the underlying biology. Network analysis, e.g. has shown that many types of biological networks across a broad spectrum of organisms share important graph-theoretic properties. These properties suggest testable biological hypotheses that may in turn reveal the fundamental design principles of biological systems. For example, there is evidence that many types of biological networks are scale-free (Barabási and Oltvai, 2004), containing a few highly connected nodes ('hubs') and a majority of nodes linked to only a few neighbors (Barabási and Albert, 1999). The fundamental hypothesis here is that scale-free networks are 'robust' to random node removals, since a majority of the nodes are not essential to cellular function and simultaneously 'fragile' because targeted removal of a few hubs will disrupt cellular function. This 'robust yet fragile' network property is thought by some to be responsible for the ability of biological systems to survive a wide variety of attacks while being susceptible to a few specific targeted attacks (Albert *et al.*, 2000).

Network models have facilitated a shift from the study of evolutionary conservation between individual gene and gene products towards the study of conservation at the level of pathways and complexes. In particular, Kelley *et al.* (2003) show that the protein–protein interaction networks of two distantly related species, *Saccharomyces cerevisiae* and *Helicobacter pylori*, contain many of the same or very similar pathways. In many of these conserved pathways, the *S.cerevisiae* network appears to be more specialized. Resultant tools such as PathBLAST (Kelley *et al.*, 2004) are now commonly used to automate the discovery of similarities ('alignments') within and between protein–protein interaction networks.

These and other successes indicate that network modeling is a promising approach, however, visualization in this domain remains a difficult challenge for several reasons. First, the increasing number of interactions available for even small genomes/proteomes presents significant visualization challenges.

\*To whom correspondence should be addressed.



**Fig. 1.** Protein–protein interaction map showing predicted subcellular-localizations of all proteins in the secretory pathway (Scott *et al.*, 2005a). The drawing was created using the Cytoscape plugin Cerebral (Barsky *et al.*, 2007). Subcellular compartments are ordered so that compartments with the most inter-compartmental links are next to one another.

Most available biological network tools make use of generic visualization packages (and algorithms) from the graph drawing community to address problems of scalability. However, such generic layouts often produce network drawings that resemble the infamous ‘hair ball’, since the layout optimization criteria fail to take into consideration important domain-specific knowledge such as subcellular localizations, molecular functions, protein complexes or pathways (see Fig. 1). Second, we lack robust query languages embedded directly within the visualization software. These querying languages allow users to filter cell-wide networks, thus restricting their attention to a core set of nodes of particular interest and to organize this information in an intuitive manner. Third, the increasing number of *types* of interactions raises many practical and theoretical problems related to how these datasets should be integrated into network models, how this information is best visualized and how the network can be explored. For instance, consider the visualization of a biological network containing both physical and genetic interactions. The fact that protein–protein interactions occur between molecules that are at least transiently physically co-located in the cell may suggest an approach that seeks to generate a visualization that mimics cellular or organellar location. However, such an approach may be unsuitable for genetic interactions where the interacting proteins may be implicated in two distinct, distally located pathways. Other types of biological interactions are not necessarily well modeled by simple binary, pairwise graphs (i.e. 2-hypergraphs). Consider, e.g. expressing that a certain ternary complex forms if and only if each of the three components of the complex are present.

For practical reasons, including simplicity and computational tractability, most network models cannot handle such  $n$ -ary relations. Consequently, general methods of extending models such as by using  $n$ -ary relations are not practical. Extensions based on domain-specific knowledge is required. Finally, there is a general need for the biological network tools to incorporate more dynamic information within the network. A first step in this direction is the inclusion of gene expression data within several existing visualization tools. The mapping of gene expression (represented typically by color) onto network nodes can be viewed as our first glimpse of large-scale systems-wide dynamics in biological systems. However, current functionality is not capable of addressing the computational and statistical challenges that will arise as researchers begin to construct large-scale probabilistic and dynamic models of complicated, distributed biological processes, nor is it capable of addressing how dynamic models may be visually represented.

The purpose of this review is 2-fold. First, it is designed to provide readers with an overview of existing software systems for biological network visualization (Section 2). The existence of at least 35 such systems precludes an in-depth review of each, instead we have sought to single out systems that contain unique functionality that have broad applicability in many different studies. Furthermore, we discuss some of the most important features currently available in network visualization packages and summarize the tools that support them (Section 3). Second, we discuss the future of network visualization and how current tools can be improved to satisfy future demands (Section 4).

## 2 TOOL SURVEY

There are many network visualization tools and each meets some specific visualization need. In fact, during the preparation of this manuscript, we identified no less than 35 such tools, and the number continues to grow. A similar review (Saraiya *et al.*, 2005) published only a few years ago identifies barely half that number and, due to recent progress in this area, is silent on some of what are currently the most important visualization challenges. We focus here on six tools which together contain many of the important features necessary for network visualization in a broad range of studies. A more complete list can be found in the Supplementary Material.

Pathway Studio (Nikitin *et al.*, 2003) (formerly PathwayAssist) is a Microsoft Windows application that combines an extensive set of features with a polished graphic user interface. It includes, most notably, customizable network display styles for assigning visual attributes such as node color, size and shape, multi-user support, subcellular localization visualization and tight integration with several database including RESNET (<http://www.ariadnegenomics.com/>), KEGG (Kanehisa, 2002), BIND (Alfarano *et al.*, 2005), GO (Ashburner *et al.*, 2000), DIP (Xenarios *et al.*, 2000), ERGO (<http://www.integratedgenomics.com/>) and PathArt (<http://jubilantbiosys.com/>). Importantly, this package also contains an SQL-like language that allows users to query the network using some simple topological contractions with node and link attributes. In comparison to other packages, this query language provides a much more flexible tool for quickly filtering nodes of

interest from large networks. Pathway Studio is a commercial product although academic licenses are available.

Cytoscape (Shannon *et al.*, 2003) is a well-known network visualization tool supporting a core set of features including standard and customizable network display styles, ability to import a large variety of interaction files, and zoomable network views. Release 2.3 claims a high-performance rendering engine supporting networks with as many as 100 K nodes and edges. One of the most positive aspects of Cytoscape is the large user and developer base. In particular, since Cytoscape is a Java application whose source code is released under the Lesser General Public License (LGPL), it is straightforward for third-party developers to construct new plugins for the system. In fact, more than 40 plugins are currently available available for tasks including importing networks from various data formats, analyzing networks and generating networks from literature searches.

Osprey (Breitkreutz *et al.*, 2003) was one of the first tools specifically designed to visualize and analyze large networks. Consequently, all but one network layout in Osprey are elaborations of the circular layout (described subsequently) because they can be quickly computed. Other tools that specialize in large networks include Interviewer (Han *et al.*, 2004) and ProViz (Iragne *et al.*, 2005). Osprey was also one of the first tools to support functional comparisons between different networks. In particular, Osprey is able to superimpose one network additionally on top of another in order to show similarities and differences. Osprey is a Java application and can be used free of charge.

The PATIKA Project (Demir *et al.*, 2002) provides a WWW-based visual editor, PATIKAweb, for accessing a central database containing pathway data from several sources including Reactome (<http://www.reactome.org/>). The project is built on top of an extensive ontology supporting representation of biological objects at different levels of detail, and graph types that facilitate visualizations of molecular complexes, pathways and black-box reactions. PATIKAweb is capable of producing high-quality visualizations using the Tom Sawyer Visualization (<http://www.tomsawyer.com/>) software. It also supports SQL-like queries on node and edge properties. It is implemented using Java Server Pages and is publicly available for non-profit use.

VisANT (Hu *et al.*, 2005) not only provides network drawing capabilities, including support for very large networks, but it is one of the first such packages to support creation, visualization and analysis of mixed networks, i.e. networks containing both directed and undirected links. The ability to use nodes to model more complex entities such as protein complexes or pathways allow for more informative visualizations. VisANT implements algorithms for analyzing node degrees, clusters, path lengths, network motifs and network randomizations. Like Cytoscape, VisANT is a Java application that can be extended using plugins and is freely available.

ProViz (Iragne *et al.*, 2005) leverages the power of the graph drawing package Tulip (David, 2001) for handling graphs containing millions of nodes and edges, while maintaining a guaranteed response time (i.e. ProViz will never make the user wait longer than predefined length of time). Tulip supports of a sophisticated plugin architecture allowing third-party

developers to extend the system. Many plugins are currently available and shipped with the system including plugins for importing/exporting various network file formats, obtaining many different two and three-dimensional network layouts, computing various network metrics like connectivity and eccentricity, and selecting subgraphs such as spanning subtrees and paths between nodes. ProViz adds to these features the ability to import and export PSI-MI (Hermjakob *et al.*, 2004b) and IntAct (Hermjakob *et al.*, 2004a) data formats, and interfaces for exploring the GO (Ashburner *et al.*, 2000) and IntAct controlled vocabularies. There is functionality to define filters for large networks. Both ProViz and Tulip are implemented in C++ and can be installed in Windows, Linux or MacOSX. The source code is released under the GNU General Public License (GPL).

BiologicalNetworks/PathSys (Baitaluk *et al.*, 2006a) extends Cytoscape and includes much of the functionality of Pathway Studio. In fact, Pathway Studio users will recognize the dialog box used to create pathways from a set of interactions. BiologicalNetworks is a user-interface to PathSys (Baitaluk *et al.*, 2006b), a graph-based system for creating a combined database of biological pathways, gene regulatory networks and protein interaction maps. It integrates over 14 curated and publicly available data sources including BIND (Alfarano *et al.*, 2005), GO (Ashburner *et al.*, 2000) and KEGG (Kanehisa, 2002) for eight representative organisms. PathSys supports SQL-like queries that can explore network properties such connectivity and node degree. BiologicalNetworks improves on previous tools by better integrating expression data with network visualization and analysis. In particular, after importing expression data, users can apply sorting, normalization and clustering algorithms on the data and then create various tables, heat maps and network views of the data. It is implemented using Java Server Pages and is publicly available on the WWW.

### 3 FEATURE SURVEY

The utility of any biological network visualization tool ultimately depends on the supported features. In this section, we consider some of the most important features including available network layout routines, supported graphical notation, integration of analysis into the visualization, variety of user input methods, integration of external biological data sources, integration of third-party software and finally availability (licensing restrictions and platform limitations).

#### 3.1 Network layouts

Clearly, one of the most important aspects of any biological network visualization is its ability to automatically construct network drawings (or, layouts). Most tools automatically create static layouts of networks, using methods that roughly fall into one of the following categories:

*3.1.1 Circular* Nearly every tool produces circular layouts. In its simplest form, each node is placed on the circumference of a circle and links are drawn as straight-line segments between them. More complicated versions attempt to order the nodes to

uncover network symmetries and other versions place nodes on multiple concentric circles. Osprey (Breitkreutz *et al.*, 2003) implements a total of six different elaborations on the circular layout. The most complex, called ‘Spoked Dual Ring’, creates circular layouts of highly connected parts of the network inside a circle and places the remaining vertices on the circle circumference. The purpose of this layout is to highlight the highly connected parts of the network and show how they relate to the remainder of the network (e.g. see Fig. S1a–c in the Supplementary Material).

**3.1.2 Hierarchical** Directed edges are particularly important when visualizing regulatory networks. One approach to visualizing draws nodes on a series of horizontal lines so that edges are directed from nodes on lower horizontal lines to nodes on higher horizontal lines. This approach, invented by Sugiyama *et al.* (1981), is included in several systems including Pathway Studio (Nikitin *et al.*, 2003), BioPath (Schreiber, 2002), ROSPath (Paek *et al.*, 2004), CellDesigner (Kitano, 2003) and Virtual Cell (<http://www.vcell.org/>). ProViz (Iragne *et al.*, 2005) also produces hierarchical drawings but uses a different algorithm (Messinger *et al.*, 1991). An example from CellDesigner is shown in Figure S2b.

**3.1.3 Force-directed** The drawings are also known as *spring embeddings* (Eades, 1984; Frick *et al.*, 1994; Fruchterman and Reingold, 1991; Kamada and Kawai, 1989) since edges are modeled as springs that pull linked nodes together, or push unlinked nodes apart, until the layout reaches a state of equilibrium (e.g. see Figure S1a). Force-directed algorithms attempt to place nodes so that all forces are in equilibrium. This approach is quite popular because the algorithms are simple to implement, produce relatively good drawings, and are easy to tweak for specific applications. In fact, nearly every network visualization tool implements the version described by Frick, Sander and Wang (Frick *et al.*, 1999).

The main drawback of force-directed algorithms is that they can require a significant amount of time before converging to equilibrium. Fortunately, these algorithms are often easy to visually animate so the user can watch the network model incrementally approach equilibrium and perhaps terminate the algorithm when a good drawing is obtained.

**3.1.4 Simulated annealing** As the name suggests, simulated annealing methods model problem space as a set of states each with an associated energy so that low energy states correspond to potential solutions. Simulated annealing algorithms find solutions by traversing the space, moving from one network layout to another until it finds a layout with ‘sufficiently low energy’. GeneWays (Rzhetsky *et al.*, 2004) generalizes one such algorithm by Davidson and Harel (Davidson and Harel, 1996) for obtaining 3-dimensional layouts. Grid Layout (Li and Kurata, 2005) uses a related method and shows how, for a yeast network, their method seems to spacially cluster functionally related nodes. The main drawback of simulated annealing algorithms is that they tend to be slow (even in comparison with force-directed methods). Consequently, there are practical limits on the size of networks to which they can be applied.

All of the above mentioned layouts assume a graphical network model where interactions are between exactly two

interactors. Although these simple models have been shown to yield biological insight, they are incapable of modeling more complicated biological relationships that involve more than two interactors like protein complexes, relationships that depend on external factors like cell state, or regulatory circuits. For instance, although the ‘topology’ of the *cis*-regulatory network describing the set of transcription factors relevant for a particular promoter can be well described, the regulatory circuit describing the activation of the target cannot. Unfortunately, extending current visualizations to handle this added complexity is not straight-forward.

Consequently, successful visualization depends on exploiting domain-specific knowledge to reduce difficult general problems to something more manageable. There have been a few successful attempts in this direction:

**3.1.5 Subcellular localization** Both Pathway Studio (Nikitin *et al.*, 2003) and Patika (Demir *et al.*, 2002) are capable of using localization to influence network visualizations, if nodes are annotated with subcellular localizations. In particular, both systems partition the drawing space into regions corresponding to the subcellular localizations and then search for layouts where nodes are forcibly constrained to their respective locations. Both systems make use of modified force-directed algorithms to achieve this. Pathway Studio uses representative cartoons as backgrounds for each region in order to improve readability. Two other tools, Cell Illustrator (Nagasaki *et al.*, 2003) and Cerebral (Barsky *et al.*, 2007), support subcellular localization in drawings where nodes are restricted to positions on a grid. For an example, see Figure 1.

**3.1.6 Composite nodes** Pathway Studio (Nikitin *et al.*, 2003), Patika (Demir *et al.*, 2002) and VisANT (Hu *et al.*, 2005) visualize composite nodes representing molecular complexes and pathways as single nodes that can be interactively expanded to show individual members or collapsed. Some improvement to this functionality is warranted. VisANT, e.g. expands composite nodes by creating a small window on top of the current network view and then draws the composite node members inside the window. Unfortunately, these windows cover nearby neighbors making it difficult to see how the composite node members relate to the greater network. When the user attempts to expand a node in Patika (e.g. see Fig. S4), there must be space for the node to be expanded in the drawing, otherwise Patika will not expand the node.

**3.1.7 Hierarchical clusters** Hierarchical clusters of the nodes or edges can be very useful for obtaining simplified visualizations of large, complex networks. Schwikowski *et al.* (2000) show three views of a yeast protein–protein interaction network: either display all nodes and edges without any graph drawing constraints, collapse nodes of the same function into composite nodes, or collapse proteins of the same subcellular localization into composite nodes. Similar to quotient graphs (e.g. Bourqui *et al.*, 2006), an edge between two composite nodes in the last two views indicates the existence of a threshold number of edges between composite node members. More general hierarchical clusters of proteins exist, including, e.g. the Structural Classification of Proteins (SCOP) (Murzin *et al.*, 1995) and Gene Ontology (Ashburner *et al.*, 2000).

The SCOP hierarchy contains several levels of protein domain similarity differentiated by increasing specificity: class, fold, superfamily, family, protein and species. Lappe *et al.* (2001) show that a large yeast protein-protein interaction network at the SCOP level of superfamily has a very simple drawing. Edge thickness between superfamilies indicates the number of links between proteins in each superfamily.

A recent Cytoscape plugin, GenePro (Vlasblom *et al.*, 2006), partially supports interactive exploration of hierarchical network clusters (e.g. see Fig. S5). Given a protein interaction network and a predefined clustering on its nodes, GenePro initially presents the user with the most abstract view and then allows the user to expand clusters in a new window to see cluster members. GenePro can also render nodes as pie charts showing the fractions of proteins sharing a common feature.

**3.1.8 Time series** Most tools do not attempt to visualize time series data. Those that do, e.g. BioTapestry (Longabaugh *et al.* 2005), simply highlight links and nodes that are active during a given time period but otherwise present a static picture. Unfortunately, this approach fails to take advantage of the reduced network size at each time point and the small number of changes from one time point to the next.

**3.1.9 Neighbor expansion** Rather than show the entire network in one display, many tools including VisANT (Hu *et al.*, 2005), Osprey (Breitkreutz *et al.*, 2003) and MINT Viewer (Zanzoni *et al.*, 2002; Chatr-aryamontri *et al.*, 2007) initially show a small subnetwork and then allow the user to click on a node in order to add all of its neighbors to the current view (e.g. see Fig. S3). Later, the user may click the node again to hide those neighbors. The best tools provide animations from one network view to another so that the user can easily maintain a mental mapping from the previous to the current view.

**3.1.10 Three dimensions** Currently, only a few tools including GeneWays (Rzhetsky *et al.*, 2004) and ProViz (Iragne *et al.*, 2005) offer three-dimensional visualizations, and these are all very rudimentary. Making use of higher dimensions is difficult because users are generally viewing the network on a two-dimensional screen. Consequently, dynamic navigation is a necessity and is often awkward without specialized equipment. In addition, graphical complexity increases because network entities must not only represent data but also simulate distance from the user.

**3.1.11 Matrix representations** Although networks are typically visualized using the so-called ‘ball-and-stick’ representation, a complementary representation uses adjacency matrices and is often useful for uncovering topological patterns in, e.g. dense networks. In Bioinformatics, these are more commonly called clustergrams, heatmaps with dendrograms, or cluster maps. Here, each node corresponds to exactly one row and one column in the matrix and the intersection of a row and column is colored to represent the existence or strength of the link between the corresponding nodes. Drawing this matrix with different row and column orderings can uncover monochromatic patches indicating clusters of nodes with similar ‘behavior’ (e.g. Fig. 1 of Collins *et al.*, 2007). To create a matrix representation, researchers generally use one or more methods

for ‘optimally’ ordering rows and columns provided by their preferred statistical software package. Unfortunately, general-purpose orderings are rarely sufficient for specific biological applications so there has been some recent investigation of support for computing user-assisted orderings (e.g. Henry and Fekete, 2006).

## 3.2 Graphical notation

Until recently, most tools relied heavily on node/edge shape (e.g. to represent whether the node corresponds to a protein, gene or small molecule) and colors (e.g. gene expression, subcellular localization, molecular function) to visualize network properties. Unfortunately, the lack of standards describing typical biological objects results in the need for each visualization to come equipped with a legend describing symbols and colors. In some cases, the use of such node and edge attributes actually detracts from readability. This has motivated the development of informal standards that have evolved via imitation and the popularity of some tools. For example, elements of the Pathway Studio layout style has migrated into the BiologicalNetworks package. However, concern about a lack of formal standards has been increasing (Klipp *et al.*, 2007), and there now exist several efforts towards a fully standardized graphics vocabulary (Cook *et al.*, 2001; Kitano, 2003; Kohn, 1999, 2001; Kohn *et al.*, 2006; Kurata *et al.*, 2005; Pirson *et al.*, 2000). Cook *et al.* (2001) outline a carefully reasoned approach that can describe complex biological systems unambiguously. The authors hope that such a standard would be amenable to various automations including simulation and resolution of queries like ‘Find models where a potassium channel blocker affects cell-cycle progression’. Unfortunately, according to Kohn *et al.* (2006), this proposal is too cumbersome for biologists to actually use so they propose their own molecular interaction map (MIM) notation for regulatory networks that deliberately permits some forms of ambiguity. For example, in the interest of visual compactness, MIM diagrams do not specify an order for steps in a reaction. Kohn *et al.* (2006) downplay concerns for automation claiming that the process of manually creating MIM diagrams is in itself an important tool for exposing gaps in our knowledge of biological processes. Others however (Kurata *et al.*, 2003, 2005) have implemented a tool called CADLIVE that supports a slight modification of MIM. Kitano (2003) defines an alternative graphical notation called Systems Biology Graphical Notation (SBGN) that uses state-transition diagrams to model regulatory networks. In contrast to MIM diagrams, SBGN diagrams do enforce an event sequence and are consequently less compact because they may contain multiple nodes for a single molecule. In addition, they have implemented a tool called CellDesigner that generates SBGN diagrams (Funahashi *et al.*, 2003) as well as created a website (<http://sbgn.org/>) in order to promote wider community use and development of SBGN. Figure S2 shows an example from CellDesigner.

## 3.3 Integration of analysis

There are currently few network tools designed for both the visualization of biological networks and the analysis of these

networks. Consequently, users are forced to switch between tools, resulting in the need to continually import/export and reformat data. Clearly, a more optimal choice would be a network tool that supports both visualization and analysis, with a seamless integration between these two procedures. BiologicalNetworks is an early leader here allowing pathway visualizations from queries that combine gene expression data analysis with simple topological patterns. Several specialized tools exist for the integration of very specific types of visualization and data. Wilmascope (Dwyer *et al.*, 2004) shows time series data using '2.5 dimensional layouts' (networks are stacked one on top of the other by time points in order to highlight changes in the network). MAVisto (Schreiber and Schwöbbermeyer, 2005) searches for over-represented network motifs and highlights these in drawings of the network. GridLayout (Li and Kurata, 2005) highlights functionally related nodes by placing them in roughly the same regions of the drawing. A large number of tools (Dahlquist *et al.*, 2002; Grosu *et al.*, 2002; Karp *et al.*, 2002; Khatri *et al.*, 2005; Mlecnik *et al.*, 2005; Pan *et al.*, 2003; Thimm *et al.*, 2004) are designed mainly for mapping gene or protein expression profiles onto existing network diagrams.

### 3.4 User input and customization

Most of the major tools support a graphic user interface where mouse clicks/movements, simple dialog boxes and data imports allow most functionality to be accessed. However, in many cases the functionality offered by the biological network tool is insufficient or cumbersome for the task at hand. Several tools offer various means for third parties to develop new functionality and integrate it directly within the tool.

Plugins are an important way for advanced users to customize and extend an application. Of the major tools, Cytoscape, VisANT and ProViz support plugins. In fact, all three of these tools are based on a somewhat generic software design in order to stimulate a community of third party developers capable of expanding their system to address a greater range of biological applications. Plugins require however, a significant amount of effort and expertise to create, and thus do not represent a feasible avenue for tailoring systems to meet the needs of a particular scientific application for most labs.

Scripting and query languages can provide a convenient trade-off between the power and flexibility of plugins whilst conserving the convenience of features available through graphic user interfaces. Currently, Pathway Studio offers a wizard interface for creating very simple network and data queries and only BiologicalNetworks provides a language interface for expressing such queries. A recent tool, GUESS (Adar, 2006), provides the user with a powerful scripting language based on Python (<http://www.python.org/>) that includes a convenient notation for networks. The GUESS user interface consists of a component for network visualization and a component for entering commands in the scripting language. The language allows for the analysis and modification of network layouts, however, it is sufficiently simple that users do not require previous programming experience. In addition, it is relatively easy to extend its functionality by creating interfaces to other libraries such as the R statistical

library. Currently, GUESS is more widely used in the social network community but has been used for biological networks.

To the best of our knowledge, only Pathway Studio supports a multi-user environment that allows users to set sharing permissions on data and computed results.

### 3.5 Incorporation of external data sources

Most of the biological network tools are distributed with pre-formatted versions of popular interaction databases [e.g. BIND (Alfarano *et al.*, 2005) and DIP (Xenarios *et al.*, 2000)], metabolic pathway databases [e.g. KEGG (Kanehisa, 2002)], gene ontologies [GO (Ashburner *et al.*, 2000)] and molecular sequence databases [e.g. UniProt (Bairoch *et al.*, 2005), Entrez Gene <http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Gene>]. There are of course many additional bioinformatics public repositories and datasets produced by individual labs that may be relevant to the user's current study. The translation of such datasets into the format required by the specific network tool is in many cases non-trivial, especially for laboratories without significant bioinformatics expertise. Partial solutions to this ubiquitous problem can be found in several biological network tools.

In VisANT, Pathway Studio and BiologicalNetworks, all data are ported to a central repository and represented in an application-specific format. This forces the end-user to build parsers for each new data resource and reformat the data to the precise format expected by the tool. This also requires that the end-users rely on developers to maintain up-to-date versions of each imported database.

Fortunately, a small number of data format standards are becoming wide spread. When both the dataset is expressed in a standardized format and the network visualization tool supports the importation of this format, the incorporation of these datasets requires neither programming expertise nor an understanding of the format itself. The three most popular examples, SBML, PSI-MI and BioPAX, are open, XML-based (<http://www.w3.org/XML/>) standards that use a *leveled approach*, meaning that each standard is described at various levels of complexity and specificity. Users may then choose the simplest level sufficient to represent all of the necessary information in their dataset. The SBML (Systems Biology Markup Language) (Hucka and Finney, 2003; Hucka *et al.*, 2003) is supported by over 100 software systems, including Cytoscape, PATIKA and BiologicalNetworks. SBML is designed for modeling biochemical reaction networks at a level that admits automated simulation. Currently, SBML levels 1.0 and 2.0 are defined but a third, more detailed level is planned that would support the composition of models from component submodels, rule-based generation of states and interactions, and descriptions of cell geometries (Hucka and Finney, 2003).

Unlike SBML, PSI-MI (Proteomics Standards Initiative-Molecular Interactions) (Hermjakob *et al.*, 2004b) has the more pragmatic mandate of describing molecular interactions rather than complete cellular models. Several visualization tools including Cytoscape and VisANT can import and export PSI-MI formatted data and several public databases accept submissions in PSI-MI format.

A third standard, BioPAX (<http://www.biopax.org/>), has released two levels. Level 1 is used for representing metabolic pathways, while level 2 is for representing molecular interactions. PATIKA can export data in BioPAX format whereas VisANT and Cytoscape (using the BioPAX plugin) can import BioPAX files. In a comparative study of PSI-MI, SBML and BioPAX, Strömbäck and Lambrix (2005) claim that BioPAX is the most general and expressive of the three while PSI-MI is most appropriate for describing molecular interactions and SBML is best suited for simulation models of molecular pathways.

### 3.6 Integration of third-party software

Like external datasources, interfacing with external software tools can be very difficult or even impossible. However, overcoming this hurdle is key to contributing to the trend toward integrated visualization and analysis. There have been a few early attempts. For example, GUESS (described earlier) contains a nearly transparent interface to the R statistical package (<http://www.r-project.org/>) in its scripting language. Use of the CytoTalk plugin for Cytoscape is a quick way to add network visualization facilities to analysis software. The plugin simply allows external processes (including, e.g. R, Python and Perl) to remotely manipulate networks displayed in Cytoscape.

## 4 DISCUSSION

High-throughput experimental biology has already managed to create nearly cell-wide maps of protein–protein, protein–nucleotide and genetic interactions. As the cost, efficiency and accuracy of these assays improve, we can expect datasets several orders of magnitude larger than we currently have at our disposal, over a greater number of organisms. Furthermore, the variety of data types will also increase to include, e.g. a greater emphasis on protein–small molecule interactions, more comprehensive time-series gene and protein expression data and more comprehensive genetic interaction studies. There is a natural trend towards combining this information within probabilistic or dynamical models, capable of capturing the most salient aspects of complex biological processes. In order to extract as much information as we can from these meta-datasets, current biological network tools must evolve in functionality to address issues of scalability, integration and visualization.

With respect to scalability, visualization tools will need to employ novel approaches to providing access to massive datasets while respecting end-user limitations. More than ever before, users will depend on sophisticated network analysis algorithms to uncover interesting biological stories that they used to find ‘by eye’ in much smaller networks. There are several positive examples in this direction. Given a pathway in some organism, PathBLAST is able to identify evolutionarily conserved pathways in a second organism, by solving a restricted version of the subgraph isomorphism problem. Given a list of distinguished genes or gene products corresponding to, e.g. differentially expressed genes from a microarray experiment, the Steiner approach of Scott *et al.* (2005b) searches large interaction networks for the minimal connecting

set of nodes that ‘hook up’ the members of the input list. The resultant subnetwork is typically of a size that is more amenable to visualization and analysis.

Unfortunately, the size of cell-wide interaction networks renders many network analysis problems intractable. For instance, MAVisto (Schreiber and Schwöbbermeyer, 2005) determines whether or not certain motifs are over-represented in a network. Since determining the presence of a motif is equivalent to solving the subgraph isomorphism problem, it is very unlikely that any computer will ever be able to search for motifs of more than a few nodes in any cell-wide interaction network. Fortunately, the computational intractability of a problem in general does not imply intractability in a specific biological context. Creatively rephrasing the question can lead to tractable solutions, e.g. by restricting attention to the Steiner tree rather than the original network, or to a single cell state or location.

With respect to integration, visualization tools will need to go beyond simplistic graphical models and mere compliance with accepted standards in order to truly integrate new data types. Whereas interaction network models assume a static list of interacting pairs, there are many examples of proteins whose function and therefore interaction types and partners differ depending on cell state or subcellular location. For example, in the absence of glucocorticoids, the glucocorticoid receptor (GR) is bound to the cytosolic chaperone Hsp90 in the cytosol. The introduction of glucocorticoids causes the release of GR from Hsp90 and its subsequent retrotranslocation into the nucleus, where it functions either as a transcription factor (protein–nucleotide interactions) or an adapter protein within large transcriptional complexes (protein–protein interactions). We are not yet, however, close to obtaining complete information about cell state so models will need to strike a delicate balance between being immediately useful and attempting to model true biological networks. Standards like SBML (Hucka and Finney, 2003; Hucka *et al.*, 2003] that aim at completeness can be used to measure progress.

With respect to visualization, single network views will provide little more than brief glimpses of the large datasets. Visualization tools will then need to support many different types of views, each network view at a different level of detail. Dynamic navigation from one view to another will be a key to showing the connection between different views. Navigating from one time series point to another, for instance, could involve a view showing only the differences between the two time points. If the time points are consecutive, the number of differences will tend to be quite small. A similar approach could be applied to subcellular localization information as well.

To adequately address each of these issues, active cooperation will be required between a variety of research fields including *graph drawing*, *information visualization*, *network analysis* and of course *biology*. It is unfortunate to witness the number of new tools designed for biology that have earlier analogs in other research areas. Pajek (Batagelj and Mrvar, 2001), for instance, is a general visualization and analysis tool that has been used extensively to study social networks but has seen limited use with biological networks [Ho *et al.*, 2002; Ludemann *et al.*, 2004; Tong *et al.*, 2001]. This waste of resources is caused by a simple lack of communication between

fields. For instance, graph drawing researchers ignorant of emerging biological models tend to tackle layout problems of little use in biology or, at best, express solutions to relevant problems in a way that is inaccessible to biologists. A biologically informed graph drawing community, however, would be capable of generating clever ideas for optimization criteria that produce layouts more easily interpreted by biologists. This would have tremendous benefits by better facilitating the evaluation of correctness of existing interaction networks. Consider, e.g. a visualization of protein-protein interactions where proteins are placed in the layout according to subcellular localizations. Edges that span any membrane would be identified as potential mistakes, either because they are false positives identified by the assay, or because the subcellular location may be improperly or incompletely described in the dataset.

The key to success of this endeavor is the involvement of all these communities towards standards, open access software, and distributed development. Tools developed in this environment will attract large user communities, avoid duplication of effort and ultimately lead the way towards the goals of systems biology.

## ACKNOWLEDGEMENT

Funding was provided by NSERC (Postdoctoral Fellowship to M.S.; Discovery to M.H.).

*Conflict of Interest:* none declared.

## REFERENCES

- Adar,E. (2006) Guess: a language and interface for graph exploration. In *CHI '06: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press, New York, USA, pp. 791–800 ISBN 1-59593-372-7.
- Albert,R. *et al.* (2000) Error and attack tolerance of complex networks. *Nature*, **406**, 378–382.
- Alfarano,C. *et al.* (2005) The biomolecular interaction network database and related tools 2005 update. *Nucleic Acids Res.*, **33**, 418–424.
- Ashburner,M. *et al.* (2000) Gene ontology: tool for the unification of biology. the Gene Ontology Consortium. *Nat. Genet.*, **25**, 25–29.
- Bairoch,A. *et al.* (2005) The universal protein resource (uniprot). *Nucleic Acids Res.*, **33** (Suppl. 1): D154–D159.
- Baitaluk,M. *et al.* (2006a) BiologicalNetworks: visualization and analysis tool for systems biology. *Nucleic Acids Res.*, **34** (Suppl. 2): W466–471.
- Baitaluk,M. *et al.* (2006b) Pathsys: integrating molecular interaction graphs for systems biology. *BMC Bioinformatics*, **7**, 55.
- Barsky,A. *et al.* (2007) Cerebral: a Cytoscape plugin for layout of and interaction with biological networks using sub-cellular localization annotation. *Bioinformatics*, **23**, 1040–1042. doi: 10.1093/bioinformatics/btm057.
- Barabasi,A.-L. and Albert,R. (1999) Emergence of scaling in random networks. *Science*, **286**, 509–512.
- Barabasi,A.-L. and Oltvai,Z.N. (2004) Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.*, **5**, 101–113.
- Batagelj,V. and Mrvar,A. (2001) Pajek—analysis and visualization of large networks. In Mutzel,P. *et al.* (eds.) *Graph Drawing, 9th International Symposium, GD 2001, volume 2265 of Lecture Notes in Computer Science*. Springer, pp. 477–478 ISBN 3-540-43309-0.
- Blat,Y. and Kleckner,N. (1999) Cohesins bind to preferential sites along yeast chromosome iii, with differential regulation along arms versus the centric region. *Cell*, **98**, 249–259.
- Bourqui,R. *et al.* (2006) Metabolic network visualization using constraint planar graph drawing algorithm. *Tenth International Conference on Information Visualisation (IV'06)*. Vol. IV, pp. 489–496.
- Breitkreutz,B.-J. *et al.* (2003) Osprey: a network visualization system. *Genome Biol.*, **4**, R22.
- Chatr-aryamontri,A. *et al.* (2007) MINT: the Molecular INTeraction database. *Nucleic Acids Res.*, **35** (Suppl. 1): D572–D574.
- Collins,S.R. *et al.* (2007) Functional dissection of protein complexes involved in yeast chromosome biology using a genetic interaction map. *Nature*, **446**, 806–810.
- Cook,D.L. *et al.* (2001) A basis for a visual language for describing, archiving and analyzing functional models of complex biological systems. *Genome Biol.*, **2**, research0012.1–0012.10.
- Dahlquist,K.D. *et al.* (2002) GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways. *Nat. Genet.*, **31**, 19–20.
- David,A. (2001) Tulip. In Mutzel,P. *et al.* (eds.) *Graph Drawing, 9th International Symposium, GD 2001, volume 2265 of Lecture Notes in Computer Science*. Springer, pp. 435–437. ISBN 3-540-43309-0.
- Davidson,R. and Harel,D. (1996) Drawing graphs nicely using simulated annealing. *ACM Trans. Graph.*, **15**, 301–331.
- Demir,E. *et al.* (2002) PATIKA: an integrated visual environment for collaborative construction and analysis of cellular pathways. *Bioinformatics*, **18**, 996–1003.
- Dwyer,T. *et al.* (2004) Representing experimental biological data in metabolic networks. In Chen,Y.-P.P. *APBC*, volume 29 of CRPIT. Australian Computer Society, pp. 13–20. ISBN 1-920682-11-2.
- Eades,P. (1984) A heuristic for graph drawing. *Congressus Numerantium*, **42**, 142–160.
- Echeverri,C.J. and Perrimon,N. (2006) High-throughput RNAi screening in cultured cells: a user's guide. *Nat. Rev. Genet.*, **7**, 373–384.
- Fields,S. and kyu Song,O. (1989) A novel genetic system to detect protein-protein interactions. *Nature*, **340**, 245–246.
- Fire,A. *et al.* (1998) Potent and specific genetic interference by double-stranded RNA in *Caenorhabditis elegans*. *Nature*, **391**, 806–811.
- Frick,A. *et al.* (1994) A fast adaptive layout algorithm for undirected graphs. In Tamassia,R. and Tollis,I.G. (eds.) *Graph Drawing, DIMACS International Workshop, GD '94*, volume 894 of Lecture Notes in Computer Science. Springer, pp. 388–403. ISBN 3-540-58950-3.
- Frick,A. *et al.* (1999) Simulating graphs as physical systems: a spring-embedder system for force-directed layout. *Dr. Dobbs Journal*.
- Fruchterman,T.M.J. and Reingold,E.M. (1991) Graph drawing by force-directed placement. *Soft. Pract. Exper.*, **21**, 1129–1164.
- Funahashi,A. *et al.* (2003) CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. *BIO-SILICO*, **1**, 159–162.
- Grosu,P. *et al.* (2002) Pathway Processor: a tool for integrating whole-genome expression results into metabolic networks. *Genome Res.*, **12**, 1121–1126.
- Han,K. *et al.* (2004) WebInterViewer: visualizing and analyzing molecular interaction networks. *Nucleic Acids Res.*, **32**, 89–95.
- Henry,N. and Fekete,J.D. (2006) Matrixexplorer: a dual-representation system to explore social networks. *IEEE Trans. Vis. Comput. Graph.*, **12**, 677–684.
- Hermjakob,H. *et al.* (2004a) IntAct: an open source molecular interaction database. *Nucleic Acids Res.*, **32** (Suppl. 1): 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.
- Ho,Y. *et al.* (2002) Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry. *Nature*, **415**, 180–183.
- Hu,Z. *et al.* (2005) VisANT: data-integrating visual framework for biological networks and modules. *Nucleic Acids Res.*, **33**, W352–W357.
- Hucka,M. and Finney,A. (2003) Systems biology markup language: Level 2 and beyond. *Biochem. Soc. Trans.*, **31**, 1472–1473.
- Hucka,M. *et al.* (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**, 524–531.
- Iragne,F. *et al.* (2005) ProViz: protein interaction visualization and exploration. *Bioinformatics*, **21**, 272–274.
- Kamada,T. and Kawai,S. (1989) An algorithm for drawing general undirected graphs. *Inf. Process. Lett.*, **31**, 7–15.
- Kanehisa,M. (2002) The KEGG database. *Novartis Found. Symp.*, 42–46.
- Karp,P.D. *et al.* (2002) The Pathway Tools software. *Bioinformatics*, **18**, S225–S232.
- Kelley,B.P. *et al.* (2003) Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proc. Natl Acad. Sci.*, **100**, 11394–11399.

- Kelley,B.P. *et al.* (2004) PathBLAST: a tool for alignment of protein interaction networks. *Nucleic Acids Res.*, **32** (Suppl. 2): W83–W88.
- Khatri,P. *et al.* (2005) Recent additions and improvements to the Onto-Tools. *Nucleic Acids Res.*, **33**, W762–W765.
- Kitano,H. (2003) A graphical notation for biological networks. *BioSilico*, **1**, 169–176.
- Klipp,E. *et al.* (2007) Systems biology standards-the community speaks. *Nat. Biotechnol.*, **25**, 390–391.
- Kohn,K.W. (1999) Molecular interaction map of the mammalian cell cycle control and DNA repair systems. *Mol. Biol. Cell*, **10**, 2703–2734.
- Kohn,K.W. (2001) Molecular interaction maps as information organizers and simulation guides. *Chaos*, **11**, 84–97.
- Kohn,K.W. *et al.* (2006) Molecular interaction maps of bioregulatory networks: a general rubric for systems biology. *Mol. Biol. Cell*, **17**, 1–13.
- Kurata,H. *et al.* (2003) CADLIVE for constructing a large-scale biochemical network based on a simulation-directed notation and its application to yeast cell cycle. *Nucleic Acids Res.*, **31**, 4071–4084.
- Kurata,H. *et al.* (2005) CADLIVE dynamic simulator: direct link of biochemical networks to dynamic models. *Genome Res.*, **15**, 590–600.
- Lappe,M. *et al.* (2001) Generating protein interaction maps from incomplete data: application to fold assignment. In *Proceedings of the Ninth International Conference on Intelligent Systems for Molecular Biology (ISMB)*. pp. 149–156.
- Li,W. and Kurata,H. (2005) A grid layout algorithm for automatic drawing of biochemical networks. *Bioinformatics*, **21**, 2036–2042.
- Lockhart,D.J. *et al.* (1996) Expression monitoring by hybridization to high-density oligo-nucleotide arrays. *Nat. Biotechnol.*, **14**, 1675–1680.
- Longabaugh,W.J. *et al.* (2005) Computational representation of developmental genetic regulatory networks. *Dev. Biol.*, **283**, 1–16.
- Ludemann,A. *et al.* (2004) PaVESy: pathway visualization and editing system. *Bioinformatics*, **20**, 2841–2844.
- Messinger,E.P. *et al.* (1991) A divide-and-conquer algorithm for the automatic layout of large directed graphs. *IEEE Trans. Syst. Man Cybern.*, **21**, 1–12.
- Michal,G. (1993) *Biochemical Pathways (Poster)*, Boehringer Mannheim.
- Michal,G. (1998) *Biochemical Pathways: An Atlas of Biochemistry and Molecular Biology*. John Wiley and Sons Ltd.
- Mlecnik,B. *et al.* (2005) PathwayExplorer: web service for visualizing high-throughput expression data on biological pathways. *Nucleic Acids Res.*, **33** (Suppl. 2): W633–W637.
- Murzin,A.G. *et al.* (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, **247**, 536–540.
- Nagasaki,M. *et al.* (2003) Genomic object net: I. a platform for modelling and simulating biopathways. *Appl. Bioinformatics*, **2**, 181–184.
- Nikitin,A. *et al.* (2003) Pathway studio — the analysis and navigation of molecular networks. *Bioinformatics*, **19**, 2155–2157.
- Paek,E. *et al.* (2004) Multi-layered representation for cell signaling pathways. *Mol. Cell Proteomics*, **3**, 1009–1022.
- Pan,D. *et al.* (2003) PathMAPA: a tool for displaying gene expression and performing statistical tests on metabolic pathways at multiple levels for Arabidopsis. *BMC Bioinformatics*, **4**, 56.
- Pirson,I. *et al.* (2000) The visual display of regulatory information and networks. *Trends in Cell Biol.*, **10**, 404–408.
- Ren,B. *et al.* (2000) Genome-wide location and function of DNA binding proteins. *Science*, **290**, 2306–2309.
- Rigaut,G. *et al.* (1999) A generic protein purification method for protein complex characterization and proteome exploration. *Nat. Biotechnol.*, **17**, 1030–1032.
- Rzhetsky,A. *et al.* (2004) GeneWays: a system for extracting, analyzing, visualizing, and integrating molecular pathway data. *J. Biomed. Inform.*, **37**, 43–53.
- Saraiya,P. *et al.* (2005) Visualizing biological pathways: requirements analysis, systems evaluation and research agenda. *Inf. Vis.*, **4**, 191–205.
- Schreiber,F. (2002) High quality visualization of biochemical pathways in BioPath. *In Silico Biol.*, **2**, 6.
- Schreiber,F. and Schwöbbermeyer,H. (2005) MAVisto: a tool for the exploration of network motifs. *Bioinformatics*, **21**, 3572–3574.
- Schwikowski,B. *et al.* (2000) A network of protein-protein interactions in yeast. *Nat. Biotechnol.*, **18**, 1257–1261.
- Scott,M.S. *et al.* (2005a) Refining protein subcellular localization. *PLoS Comput. Biol.*, **1**, e665. doi: 10.1371/journal.pcbi.0010066.
- Scott,M.S. *et al.* (2005b) Identifying regulatory subnetworks for a set of genes. *Mol. Cell Proteomics*, **4**, 683–692.
- Selbach,M. and Mann,M. (2006) Protein interaction screening by quantitative immunoprecipitation combined with knockdown (QUICK). *Nat. Methods*, **3**, 981–983.
- Shannon,P. *et al.* (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.*, **13**, 2498–2504.
- Strömbäck,L. and Lambrix,P. (2005) Representations of molecular pathways: an evaluation of SBML, PSI MI and BioPAX. *Bioinformatics*, **21**, 4401–4407.
- Sugiyama,K. *et al.* (1981) Methods for visual understanding of hierarchical system structures. *IEEE Trans. Syst. Cybern.*, **11**, 109–125.
- Thimm,O. *et al.* (2004) MAPMAN: a user-driven tool to display genomics data sets onto diagrams of metabolic pathways and other biological processes. *Plant J.*, **37**, 914–939.
- Tong,A.H.Y. *et al.* (2001) Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science*, **294**, 2364–2368.
- Vlasblom,J. *et al.* (2006) GenePro: a cytoscape plug-in for advanced visualization and analysis of interaction networks. *Bioinformatics*, **22**, 2178–2179.
- 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.