

Chapter 7

Network Visualization for Integrative Bioinformatics

Andreas Kerren and Falk Schreiber

Abstract Approaches to investigate biological processes have been of strong interest in the past few years and are the focus of several research areas like systems biology. Biological networks as representations of such processes are crucial for an extensive understanding of living beings. Due to their size and complexity, their growth and continuous change, as well as their compilation from databases on demand, researchers very often request novel network visualization, interaction, and exploration techniques. In this chapter, we first provide background information that is needed for the interactive visual analysis of various biological networks. Fields such as (information) visualization, visual analytics, and automatic layout of networks are highlighted and illustrated by a number of examples. Then, the state of the art in network visualization for the life sciences is presented together with a discussion of standards for the graphical representation of cellular networks and biological processes.

Keywords Biological networks • Visualization • Graph drawing • Visual analytics • Interaction • Exploration • SBGN • Visualization tools

A. Kerren (✉)

Department of Computer Science, Linnaeus University, Vejdes Plats 7, SE-351 95 Växjö, Sweden
e-mail: andreas.kerren@lnu.se

F. Schreiber

Martin Luther University Halle-Wittenberg, Von-Seckendorff-Platz 1, D-06120 Halle, Germany

IPK Gatersleben, Corrensstrasse 3, D-06466 Gatersleben, Germany

e-mail: schreibe@ipk-gatersleben.de

biological processes consists of a set of elements (called *nodes* or *vertices*) and their connections or interactions (called *edges*). These elements and connections often have a defined appearance and are placed in a specific layout. Due to the size and complexity of such networks, methods for their automatic visualization and interactive exploration are desired.

Network diagrams or maps have been produced manually for a long time. Examples are textbooks on biochemistry [8, 96], biological network posters [94, 99], and some electronic information systems such as *ExPASy* [4] and *KEGG* [61]. The drawings in these resources have been created manually long before their use and provide only a restricted view of the data. These maps represent the knowledge at the time of their generation and are static, hence cannot be changed by an end user. Therefore, this type of biological network visualization is often called *static visualization*.

Because of the size and complexity of biological networks, their steady growth and continuous change, as well as the compilation of user-specific networks from databases, novel automatic visualization, interaction, and exploration methods are desired. The generation of a network map on demand is called *dynamic visualization*. Such visualizations are automatically created by the end user from up-to-date data. Their advantages are, inter alia, that they can be modified to provide particular views at the data and often navigation and exploration methods are supported in interactive systems.

This review gives a brief introduction into (information) visualization, visual analytics, and automatic layout of networks, presents the state of the art in automatic network visualization for the life sciences, and standards for the graphical representation of cellular networks and biological processes. It is structured in two main parts as follows: Sect. 7.2 provides information about the foundations from computer science in general and looks into the subareas of *information visualization*, *graph drawing* (network visualization), and *visual analytics* in particular. Section 7.3 takes a closer look at the visualization of biological networks and discusses methods, some important tools, and the SBGN standard. It looks into the application and extension of computer science methods for the special requirements of the life sciences.

7.2 Background

The effective visualization of biological networks is influenced by research from many different fields. In the past, such networks were simply considered as large graphs (or hypergraphs), and a suitable visual representation was restricted to finding an appropriate (static) graph layout. Nowadays, research in the visualization of large and complex networks is more focused on interactive exploration and analysis that includes the consideration of additional data that might be attached to various graph elements or that might be the basis for the construction of biochemical networks. The process of such a data collection and storage will heavily increase in

the future. This is especially true in systems biology where, for example, the huge amount of *omics data automatically generated by high-throughput technologies [3, 39] lead to the challenge of interpreting all of these data sets in context of networks. The fundamental problem today is to transform the data—which is typically not preprocessed, erratic, stored in idiosyncratic formats, sometimes uncertain, and often composed of various types (multidimensional, time dependent, geospatial, etc.)—into information and make it useful/available/analyzable to analysts. Often, this challenge is called the *information overload problem*. Positive effects of such a transformation are then to discover something that is interesting (like patterns or outliers) or to monitor a huge data set in real time [70].

Because of this general view on the problem, we provide a more general background section. First, we discuss the field of information visualization in the next subsection. We highlight the most important definitions/aims and present a brief high-level overview of visual representations and interaction techniques. Then, we outline the field of graph drawing and discuss the most often used layout algorithms. Finally, a relatively new field, called visual analytics, is introduced. Due to page limitations, we cannot give a comprehensive overview of all aspects of the aforementioned research fields. Instead, we present a selection of fundamental ideas/approaches and refer to the literature including surveys.

7.2.1 *Information Visualization*

Information visualization (InfoVis) is a research area which focuses on the use of interactive visualization techniques to help people understand and analyze data. While related fields such as scientific visualization involve the presentation of data that has some physical or geometric correspondence, information visualization centers on abstract information without such correspondences, i.e., information that cannot be mapped into the physical world in most cases. Examples of such abstract data are symbolic, tabular, networked, hierarchical, or textual information sources. The ever-increasing amount of data generated or made available every day amplifies the urgent need for InfoVis tools. To give the field a firm base, InfoVis combines several aspects of different research areas, such as scientific visualization, human-computer interaction, data mining, information design, cognitive psychology, visual perception, cartography, graph drawing, and computer graphics [73, 74].

7.2.1.1 **The Importance of Human Visual Perception and Visual Metaphors**

Human information processing and the human capability of information reception have to be adequately taken into account when developing visualization tools. This should be reflected in an appropriate user interface design, a clean requirement analysis and modeling, and perhaps most important an efficient interaction between

the human analyst and the computer. Discussing the different features of our eye, the various process models of human visual perception (incl. preattentive perception and features) or our capabilities of pattern recognition would go beyond the scope of this background section. There are many good textbooks that deal with these topics in context of visualizations: we recommend the books of Ware [141], Kerren et al. [74], and Ward et al. [140].

Edward Tufte, one of the leaders in the field of visual data exploration, describes in his illustrated textbooks [131–133] how information can be prepared so that the visual representation depicts both the data and the data context. The use of suitable visual metaphors assists our brain in its endeavor to connect new information received through the visual input channels to existing information stored in short- or long-term memory [72]. Tufte inspired many InfoVis researchers in their ambition to develop novel visual representations for the data sets under consideration (the process of representing a concrete data set by an appropriate visual structure is called “visual mapping”) as well as interaction techniques which support a better understanding of the data.

7.2.1.2 Visual Representations

Visual mappings explain how data models can be expressed using visual metaphors and be converted into corresponding visual representations which are suitable for interaction. This is typically done in the 2D space, because 3D representations usually introduce unnecessary clutter and navigation problems. We highlight the most important visualization techniques for basic data types in the following paragraphs. Of course there are other types of data that have to be considered. We refer to the literature if the reader is interested to get more information, such as [27, 102] for geo-spatial data, [2] for time-series data, or [41, 126, 140] for a comprehensive discussion of visual representations in general.

Visualization Techniques for Multivariate Data

Multivariate (or multidimensional) data sets can mostly be described as data tables with n data objects and m attributes/features, i.e., for each object exists an attribute vector with m dimensions. The attribute values can be classified into nominal, ordinal, or quantitative. In practice, we often have a large amount of data objects and many attributes with different types. Finding a suitable visual representation is thus challenging, and the right choice might depend on further parameters like application domain, integration into a larger visualization environment, or support of specific interaction techniques. In general, visual mappings for multivariate data can roughly be categorized as follows:

Point-based approaches: This class of techniques projects n -dimensional objects from the data space to a lower-dimensional—typically 2D—display space [140].

There are different variations: *scatterplot matrices*, for instance, consist of a grid of 2D scatterplots each showing a possible pair of dimensions/attributes [19]; see Fig. 7.2b for an example. Dimensional reduction techniques, such as multidimensional scaling (*MDS*) [92, 145], principal component analysis (*PCA*) [53], or self-organizing maps (*SOMs*) [80], project n -dimensional data records into 2D/3D directly. The idea is to preserve properties of the multivariate data space during the projection, i.e., similar data objects in data space should also be similar in display space which is represented by neighborhood. Note that absolute positions in the display space are less important, in contrast to relative positions.

Axis-based approaches: Here, a multidimensional data object is usually represented by a polyline, and its attribute values are marked on coordinate axes which can be arranged in various ways. Thus, the user can read the attribute values from the intersections between the coordinate axes and the polyline. The most prominent examples are *parallel coordinate systems* [49] (cf. Fig. 7.2a) or *star plots* [16] (also called Kiviat diagrams).

Icon-based approaches: Icon- or glyph-based approaches are coherent graphical entities that represent the attribute values of a data record by modification of the entity's visual features, such as line thickness, size, color, and orientation. There are many different realizations, such as *stick figures* [106], *Chernoff faces* [18], or *shape coding* [7]. A variant of so-called rose diagrams [100] is shown in Fig. 7.2c.

Pixel-based approaches: Such approaches try to maximize the available display space by mapping attribute values to single pixels. There is only one degree of freedom to represent such a value by a pixel: its color. Therefore, the challenge in the development of pixel-based representations is to arrange the used pixels on the screen in a meaningful way. Well-known examples are *recursive patterns* [65] or the *VisDB* tool [66] for the analysis of databases. Figure 7.2d exemplifies the idea in context of the visualization of weather data collected over time.

Visualization Techniques for Hierarchical Data and Networks

Networks and trees are in the center of our interest in this chapter. Therefore, we provide an own Sect. 7.2.2 for a deeper discussion of suitable visualization possibilities for these data types and focus there on traditional node-link approaches. For the sake of completeness, we want to note that there are also so-called space-filling methods that try to solve some conceptual problems of node-link diagrams, such as the high space consumption and difficult inclusion of many (and complex) attributes into the drawing. *Treemaps* fall into this category in which the hierarchy is recursively mapped to rectangular areas [52]. Other examples are *Beamtrees* [134], *sunburst* approaches [108], or *network matrices* [1].

Visualization Techniques for Text and Documents

Today, the availability of texts and documents is overwhelming, and people want to actively deal with them to solve specific problems. Typical questions are as follows: what documents contain a text about a specific topic? Or are there similar documents to those that I already have? Information visualization is capable of supporting the aforementioned tasks in several ways.

Text visualization: First, we focus on approaches to the visualization of a single text document. *Tag Clouds* provide information about the frequency of words contained in a text [63]. The approach uses different font sizes for each word in the text to indicate how often a certain word is used in comparison with the other words as shown in Fig. 7.2e. Several extensions and related approaches exist, such as *Wordle* or *ManiWorldle* [77, 138]. *SparkClouds* extend the original tag cloud idea with a temporal variable by so-called sparklines [87]. Thus, trends can easily be identified and analyzed. An approach for visual literary analysis is called *Literature Fingerprinting* [67]. It supports the visual comparison of texts by calculating features (e.g., word/sentence length or measurement of vocabulary richness) for different hierarchy levels and by creating characteristic fingerprints of the texts.

Document visualization: Collections of text documents can be structured to some extent (software packages, wikis, etc.) or relatively unstructured (e-mails, patents, etc.). Early approaches, e.g., *Lifestreams* [34], simply arranged documents according to specific attribute values such as time tags. More recent works analyze the documents by metrics, such as similarity, and perform cluster analyses or compute SOMs. Conceptually similar (by looking at the resulting visual representation) is *ThemeScapes* [147] that follows a natural landscape metaphor. Single documents are categorized and then mapped to a document map as topic areas, whereas the documents themselves are shown as small dots. “Mountains” in the landscape represent document concentrations in a thematic environment (density), height lines connect concept domains, etc. There are many more recent approaches that make use of the same metaphor, such as [104]. In order to carry out comparisons of text documents using tag clouds, *Parallel Tag Clouds* [20] arrange tags on vertical lines for each document. Identical words are then highlighted by connection lines.

7.2.1.3 Interaction Techniques

Interaction techniques in information visualization are mechanisms “for modifying what the users see and how they see it” [140]. There are many taxonomies of interaction techniques in the literature which help to better understand the design space of interaction; a nice overview is provided by Yi et al. [148]. In the following, we present a simplified and shortened classification of interaction methods for information visualization from our paper [70] which is based on [43] of its own:

Data and view specification: This category focuses on the data space and how the data is visually represented (corresponds to data transformations and visual mappings in the InfoVis Reference Model [14]):

- *Encode/visualize:* Users can choose the visual representation of the data records including graphical features, such as color and shape. Visual representations typically depend on the data types as discussed in Sect. 7.2.1.2.
- *Reconfigure:* Some interaction techniques allow the user to map specific attributes to graphical entities. An example is the mapping of attributes in a multivariate data set to different axes in a scatterplot.
- *Filter:* This technique is of great importance as it allows the user to interactively reduce the data shown in a view. Popular methods are *dynamic queries* by using range sliders [146] or picking a set of nodes in a network visualization for further analyses by performing a “lasso” selection [44].
- *Sort:* Ordering of records according to their values is a fundamental operation in the visual analysis process. This is, for example, important in network analysis where nodes might be sorted based on specific centrality values [150].

View manipulation: Our second category addresses interacting with visual representations (view transformations in the InfoVis Reference Model).

- *Select:* Selection is often used in advance of a filter operation. The aim is to select an individual object or a set of objects in order to highlight, manipulate, or filter them out. Examples include putting a placemark on a virtual map to highlight a spatial area or the specification of attribute ranges in parallel coordinate systems as seen in Fig. 7.2a.
- *Navigate/explore:* This important class of interaction techniques typically modify the level of detail in visualizations following the mantra *overview first, zoom and filter, and details on demand* [121]. Well-known approaches are *focus and context* [111], *overview and detail* [51], *zooming and panning* [137], and *semantic zooming* [127].
- *Coordinate/connect:* Linking a set of views or windows together to enable the user to discover related items. Brushing and linking techniques (e.g., *histogram brushing* [89]) are used in almost all information visualizations, such as in [59].
- *Organize:* Large visualization systems often consist of several windows and workspaces that have to be organized on the screen. Adding and removing views can be confusing to the analyst. Some systems help the user to better overview and to preserve his/her mental map by grouping of views or by assigning specific places where they have to appear [50, 91].

Note that it is possible and also common practice to combine the aforementioned techniques. The given literature references only point to selected example works and make no claim to be complete.

7.2.2 Graph Drawing and Network Visualization

In this subsection, we distinguish between *graphs* and *multivariate networks*. A (simple) graph $G = (V, E)$ consists of a finite set of vertices (or nodes) V and a set of edges $E \subseteq \{(u, v) | u, v \in V, u \neq v\}$, whereas a multivariate network N consists of an underlying graph G plus additional attributes that are attached to the nodes and/or edges. To describe the fundamental ideas of graph visualization algorithms more efficiently, we have to provide some definitions:

- An edge $e = (u, v)$ with $u = v$ is called a *self-loop*.
- If an edge e exists several times in E , then it is called a *multiple edge*.
- A *simple graph* has no self-loops and no multiple edges. Here, we assume that all graphs are simple graphs for the sake of convenience.
- The *neighbors* of a node v are its adjacent nodes.
- The *degree* of a node v is the number of its neighbors.
- A *directed graph* (or digraph) is a graph with directed edges, i.e., (u, v) are ordered pairs of nodes.
- A directed graph is called *acyclic* if it has no directed cycles, i.e., there is no directed path where the same node is visited twice.
- A graph is *connected* if there is a path between u and v for each pair (u, v) of nodes.
- A graph is *planar* if it can be drawn in the 2D plane without intersections of edges (*edge crossings*).

7.2.2.1 Traditional Graph Drawing (GD)

Graph drawing algorithms compute a 2D/3D layout of the nodes and the edges, mainly based on so-called node-link diagrams [141]. They play a fundamental role in network visualization. Particular graph layout algorithms can give an insight into the topological structure of a network if properly chosen and implemented. The graph readability is affected by quantitative measurements called *aesthetic criteria* [24], such as:

- Minimization of edge crossings
- Minimization of the drawing area
- Displaying the symmetries of the graph topology
- Constraining edge lengths
- Constraining the number of edge bends
- Maximization of the resolution

Thus, graph drawing generally deals with the ways of drawing graphs according to the set of predefined aesthetic criteria [17]. A problem is that these criteria are often contradictory, and problems which aim to optimize the criteria are often NP-hard. Therefore, many GD algorithms are heuristics. Note that we only focus

on traditional GD approaches in this subsection. There are further possibilities to represent graphs, such as matrix representations [1] or hybridizations between both approaches [44] (cf. Sect. 7.2.1.2).

In the following paragraphs, a selection of drawing approaches is presented. These are layout methods for trees, force-based layout techniques, and hierarchical drawings. There are many more approaches not discussed here, for instance, orthogonal layouts [29], visualization of hypergraphs [9], or dynamic layouts for graphs that change over time [25] (a possible application of dynamic approaches is visualizing the evolution of biochemical networks [112], for instance). Implementing good graph drawing algorithms is usually complicated and time-consuming. Therefore, a number of different open source libraries were developed, such as *JUNG* [105] and many others, that allow to simply call predefined methods for the computation of a specific graph layout.

Tree Drawings

Trees are a special case of directed (acyclic) graphs that usually have a distinguished node called the *root* of the tree. We can regard a tree as a digraph with all edges oriented away from the root. A binary tree is a rooted tree where each node has at most two children (we assume here that binary trees are ordered). The graph drawing community developed a lot of different layout methods for binary and general trees. In this context, there is another set of more specified aesthetic criteria especially for (binary) trees:

- Nodes at the same level of the tree should lie along a straight line, and the straight lines defining the levels should be parallel.
- A left subtree should be positioned to the left of its parent node and a right subtree to the right.
- A parent node should be centered over its subtrees.
- Two isomorphic subtrees should be drawn equally. Graph isomorphism means that there is a bijection between two graphs, so that any two nodes u and v are adjacent in the first graph if and only if their bijections are adjacent in the second graph.
- A tree and its mirror image should produce drawings that are reflections of one another.
- Integer coordinates should be preferred which leads to a grid drawing at the end.

Many tree layout algorithms use a divide and conquer strategy, such as the well-known Reingold/Tilford algorithm for binary trees [107]. In a postorder traversal of the tree, the following simple steps are executed:

1. Draw the left subtree.
2. Draw the right subtree.
3. Combine both drawings with a specific minimum distance.

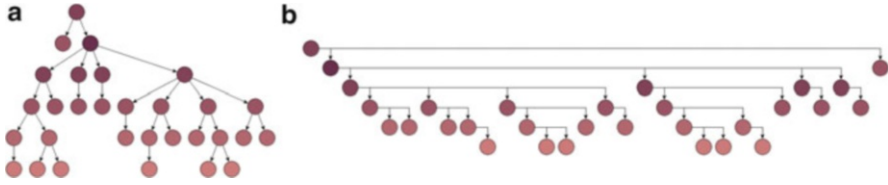


Fig. 7.3 Two sample tree layouts that were computed and displayed by the *yED* graph editor [149]. The identical input tree has 30 nodes and 29 edges. (a) A standard tree layout for general trees. (b) A so-called HV-drawing in which the layout algorithm switches between the *horizontal* and *vertical* orientation

4. Place the root of both subtrees at the next upper level exactly in the center of its subtrees.
5. In case the parent node has only one subtree, place the root in a specific horizontal distance.

Reingold/Tilford runs in linear time and can relatively easily be extended for the layout of general trees [13, 139]. Of course, there are further possibilities of drawing trees with the help of node-link diagrams, such as radial layouts, H-trees, or HV-trees. We refer the reader to the standard literature [24, 64]. Figure 7.3 shows two example layouts computed with the *yED* tool [149].

Force-Based Drawings

Force-based layout techniques use a physical analogy to draw graphs and are widely used in practice. This is because of several reasons: the physical metaphor makes them easy to understand and to code, the results are suitable for many application fields, they are easy to extend with additional constraints, and the process of obtaining an equilibrium state (see below) can be animated which looks pretty nice. A simple version of a force-based layout algorithm using spring and electrical repulsion forces is introduced in the following. Here, the edges between nodes are modeled as springs, and the nodes can be considered as charged particles that repel each other. For the x-component of the force vector on a node v , the following holds (y-component analogous):

$$\sum_{(u,v) \in E} (\text{sti}_{uv}(d_{uv} - l_{uv}))\hat{x}_{uv} + \sum_{(u,v) \in V \times V} \frac{\text{rep}_{uv}}{d_{uv}^2}\hat{x}_{uv} \quad (7.1)$$

Here, \hat{x}_{uv} denotes the unit vector of $(x_v - x_u)$. d_{uv} is the Euclidean distance between u and v , l_{uv} is the zero-energy (natural) length of the spring between u and v (i.e., no force if $d_{uv} = l_{uv}$), $\text{sti}_{uv} \in [0, 1]$ is the stiffness of the spring between u and v (i.e., the larger this parameter the more the tendency for d_{uv} to be close to l_{uv}), and finally rep_{uv} is the strength of the electrical repulsion between the two nodes. In Eq. 7.1,

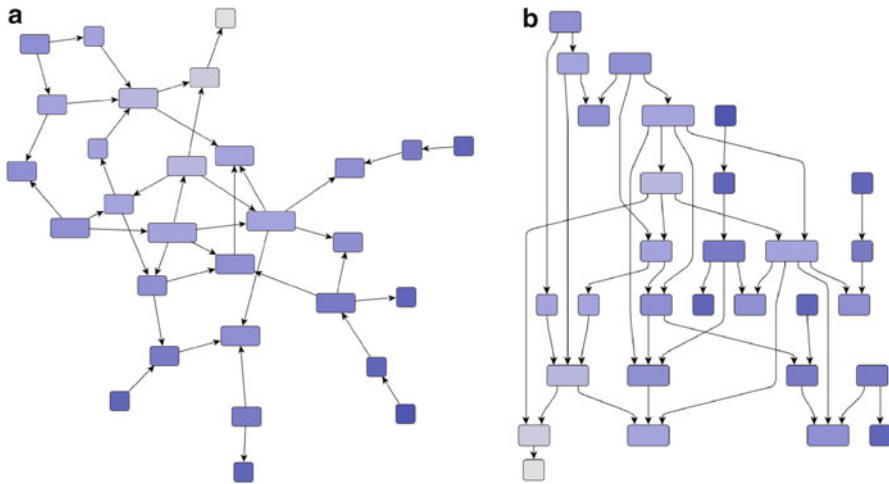


Fig. 7.4 Two sample graph layouts that were computed and displayed by the *yED* graph editor [149]. The identical input digraph has 29 nodes and 39 edges. (a) Result of a force-based layout algorithm. (b) Layered (or hierarchical) drawing

the first sum represents the spring force between two nodes u and v connected with an edge and the second sum the repulsion force between v and other nodes. Both forces together build a complete force system for all graph elements. Depending on the underlying physical model, the repulsion forces avoid that nodes are getting too close, and the spring forces provide a uniform edge length, for instance. In the current formula, Hook's law is used to specify the spring force between two nodes, i.e., if the distance between the two nodes is larger than the natural length of the spring, then the nodes attract each other. And the strength of the attraction is proportional to the difference between distance and natural length.

A simple algorithm that computes a final graph layout consists of a loop which firstly computes the forces of all nodes and then moves each node a bit into the direction of its force vector computed in Eq. 7.1. At the beginning, all nodes are positioned randomly. The loop is left if the sum of all forces together is small enough (equilibrium state) or after a specific number of iterations. This strategy works for undirected and directed graphs, with and without cycles, cf. Fig. 7.4a.

Layered (Hierarchical) Drawings of Directed Graphs

A general aim for the layout of a directed graph is to compute a so-called *monotone* drawing in which all edges point into the same direction. Such a monotone drawing has some advantages in the interpretation of the digraph's topology [47]. Obviously, the input digraph must be acyclic in that case, otherwise we would get edges that flow backwards (called *feedback edges*). In practice this apparent hard condition is not really a problem, because we can use such a drawing method for general directed

graphs if we change the direction of a minimal number of the feedback edges. This step is known as *cycle removal*. By doing so, we get a directed acyclic graph (DAG) that is drawn by using a method for computing monotone layouts, such as a layered drawing as explained in this paragraph. If the final layout is ready, we simply reverse the feedback edges again.

Many people prefer a hierarchical structure of the final graph layout, i.e., the nodes of the graph are arranged on vertical or horizontal, parallel layers in the 2D plane. Often, such a structure is already given by the input data. For instance, if someone wants to visualize hyperlinks (edges) between the HTML pages (nodes) of a website, then usually the pages are already hierarchically organized. In the following, we briefly present a standard technique for layered drawings that is based on the fundamental work of Sugiyama et al. [129].

The basic idea is very simple and intuitive; it has three phases. In the first phase, the nodes of the graph are assigned to a number of layers (we can skip this phase if there is already a layering in the input graph). This layer assignment problem is NP-complete if we want to minimize the height and the width of the final layering. A further complication occurs if edges span over several layers: then we have to introduce the so-called dummy nodes that lie on the spanned layers, i.e., a long edge is thus subdivided by the dummy nodes. This strategy causes modified edges which only reach from one layer to the next one (the digraph is called *proper* in such cases) and is needed for the second phase. After the layer assignment, we have to eliminate the number of edge crossings. This is done by reordering the graph nodes and the dummy nodes within each layer. With the help of the dummy nodes, the algorithm gets control over the edge positioning, and in consequence, it is possible to avoid crossings of edges that span over several layers. Minimizing edge crossings in a proper layered digraph is NP-complete, even if there are only two layers. Note that the node positions (x -coordinates) on the layers are relative only up to now (the y -coordinates of the nodes are already specified by the node layers if we assume to have horizontal layers). The final phase is the real coordinate assignment of all nodes on the layers, i.e., we assign concrete x -coordinates for each (normal and dummy) node. Also this task leads to an optimization problem that can be solved, for instance, by linear programming (LP). Constraints of the LP are then the fixed orderings in the layers, and the target function is specified by the straightness of the edges. As a final step, we remove the dummy nodes and obtain the wished layered drawing as shown in Fig. 7.4b.

7.2.3 *Multivariate Network Visualization*

Good drawing algorithms as described in the previous subsection will not solely solve the problem of visualizing multivariate networks. There are several reasons for this statement. First, the most traditional graph drawings do not scale well, i.e., they are not able to represent huge data sets with many thousands of nodes and/or edges. Second, additional multivariate data cannot be intuitively embedded

into a standard drawing. The InfoVis community tried to address those issues by visualization approaches that provide filtering and interaction possibilities in order to reduce the number of graph elements under consideration as well as by methods to visually analyze attributes in context of the underlying graph topology. Several approaches can be found in the literature that attempt to offer solutions for the problem of visualizing multivariate networks: *multiple and coordinated views*, *integrated approaches*, *semantic substrates*, *attribute-driven layouts*, and *hybrid approaches* [57]. We will discuss these concepts in the following paragraphs:

Multiple and coordinated views: This category of solutions aims to combine several views and present them together. Coordinated views allow the use of the most powerful visualization techniques for each specific view and data set [41, 109]. As an application example, we highlight the work of Shannon et al. [120] who realized this idea in the network visualization domain. They use two distinct views: one view shows a parallel coordinate approach for the visual representation of the network attributes and the other view displays a node-link drawing of a graph. Their tool is equipped with a variety of visualization and interaction techniques; both views are coordinated by linking and brushing [126] techniques. The drawback of multiple views is that they split the displayed data because of the spatial separation of the visual elements.

Integrated approaches: To provide a combined picture, attributes and the underlying graph can be displayed in one single view. “Integrated views can save space on a display and may decrease the time a user needs to find out relations; all data is displayed in one place” [41]. One example is described in Borisjuk et al. [10] work on the visualization of experimental data in relation of a metabolic network. The authors used a straightforward approach by employing small diagrams instead of representing the nodes as simple circles or rectangles. Each diagram, e.g., a bar chart, shows experimental data that is related to the regarded node. This approach provides a view to all available information, but the embedding of the visualizations into the nodes causes the nodes to grow in size. This issue may affect the readability of the network due to the overlaps that may appear when the number of nodes and the attributes is high [71]. Thus, it does not scale well. However, the problem of space usage and clutter introduced by such approaches can be avoided by using focus and context techniques (cf. Sect. 7.2.1). *Magic lenses* are one of several possibilities that are able to interactively visualize the node attributes within the same view as exemplified in Fig. 7.5.

Semantic substrates: In order to further avoid clutter in multivariate network visualizations, some researchers realized the idea of so-called semantic substrates that “are non-overlapping regions in which node placement is based on node attributes”: Shneiderman and Aris [122] introduced this idea and combined it with sliders to control the edge visibility and thus to ensure comprehensibility of the edges’ end nodes. One conceptual drawback of such approaches is that the underlying graph topology is not (completely) visible.

Attribute-driven layouts: Those layouts use the display of the network elements to present insight about the attached multivariate data instead of visualizing the graph

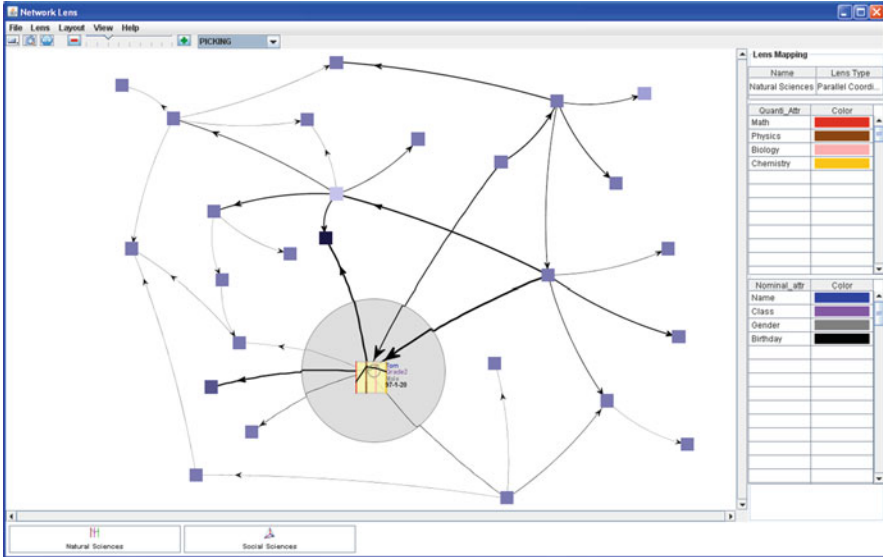


Fig. 7.5 Overview of the *Network Lens* tool [58]. The graphical user interface is divided into three distinctive parts: the main network visualization area, the lens information area on the *right-hand side*, and the *bottom* part where user-produced lenses are preserved. It offers a way to visualize additional network attributes (displayed inside of the circular lens), while preserving the overall network topology and context. The lens in the screenshot covers one node only and shows a small parallel coordinate diagram with four quantitative as well as four nominal attributes belonging to that node. The user is able to move the lens with the mouse or to translate the graph behind the lens

topology itself. While being similar to semantic substrates, this technique does not necessarily place the nodes into specific regions. Instead, it uses calculations based on node attributes to control the placement of a node in the graph layout. An example is *PivotGraph* [142] which uses a grid layout to show the relationship between (node) attributes and links.

Hybrid approaches: They combine at least two of the previously discussed techniques. The most common combinations are multiple coordinated views with any of the integrated approaches. For instance, Rohrschneider et al. [112] integrate additional attributes of a biological network inside the nodes and edges; see Fig. 7.6. The authors also use other visual metaphors for creating multiple coordinated views to show time-related data of the network.

7.2.4 Visual Analytics

Visual analytics (VA) “is the science of analytical reasoning facilitated by interactive visual interfaces” [130]. A crucial property of this research field is that computational methods of data analysis are combined with interactive visualization

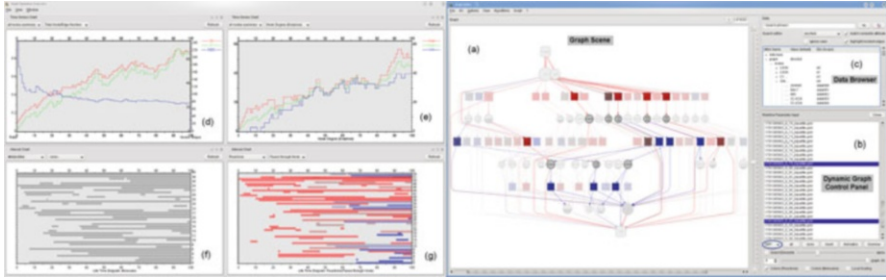


Fig. 7.6 The screenshot shows a tool for the visual analysis of dynamic metabolic networks [112]. On the *left-hand side*, two time-series charts of selected attributes display attribute dynamics over time. Interval charts represent the dynamic topology of the graph in terms of life times of metabolites, enzymes, and reactions. On the *right*, the graph scene shows the set union graph (= the super graph that summarizes all nodes/edges of the individual graphs that appear over time) with the applied node coloring scheme which supports distinguishing between older and newer nodes

techniques in order to analyze data more efficiently. *Automatic data analysis* covers various aspects from data storage and organization to automatic analysis algorithms, such as support vector machines, neural networks, and PCA. It might be classified among others into data management, data mining, and machine learning. For many data analysis problems, fully automated analysis methods only work for well-defined and well-understood problems, i.e., there has to exist a model of the underlying problem [68]. Otherwise, traditional data mining techniques will not work. Even if a model exists, then the results of the automated analyses have to be sufficiently communicated to and interpreted by analysts. Here, *interactive visualizations* come into the play as they are able to support the analyst to discover (possibly unexpected) patterns, trends, or relationships in the data. Interaction techniques (as presented in Sect. 7.2.1.3) are of particular importance to visually analyze large volumes of data. Interaction allows, among other things, to explore “unknown” data collections following Shneiderman’s mantra of information visualization [121] or to build hypotheses with the help of “What if?” questions and to verify them visually or with algorithmic methods. The need to combine interactive visualization with computational analysis methods is obvious and opens novel possibilities to address the information overload problem. A more detailed discussion on VA can be found in [68, 69, 130].

As an example from the field of visual network analysis, we have selected the *ViNCent* tool [75, 150] that combines exploratory data visualization with automatic analysis techniques, such as computing a variety of centrality values for network nodes as well as hierarchical clustering or node reordering based on centrality values. Automatic and interactive approaches are seamlessly integrated in one single analysis framework which provides insight into the importance of an individual node or groups of nodes and allows quantifying the network structure; see Fig. 7.7.



Fig. 7.7 Overview of the ViNCent user interface [150]. The center shows the radial centrality view of the input network. The right side displays the corresponding histograms of the network centralities as well as detailed values of the network centralities for the currently hovered node. Histograms can be used to filter the views. The left panel allows changing the render settings and displays an overview of the respective node-link layout of the network. A node group has been manually selected and is shown as a light-blue stripe along the outer circle in the centrality view as well as in the overview (bottom left) by using a background region of the same color (Color figure online)

7.3 Visualization of Biological Networks

Visual representations of biological networks are widely used in the life sciences. Examples are shown in textbooks, on pathway posters, in databases, and by a large number of tools for the analysis and visualization of biological processes. Well-known software tools are listed in Sect. 7.3.1.2. Software tools often use established layout methods as described in Sect. 7.2.2 to visualize biological networks automatically. Sometimes those algorithms are modified, for example, by adding extra forces to force-based approaches. However, often these methods do not or only partly take into account specific requirements for the visualization of a particular biological network, and hence these visualizations are usually difficult to understand, especially if large networks are visualized.

In the following subsections, we will introduce some typical solutions for common networks from molecular biology, discuss domain-adapted solutions for particular networks, list major tools for the visualization of biological networks, and finally discuss the *Systems Biology Graphical Notation* (SBGN) as the graphical standard for biological networks.

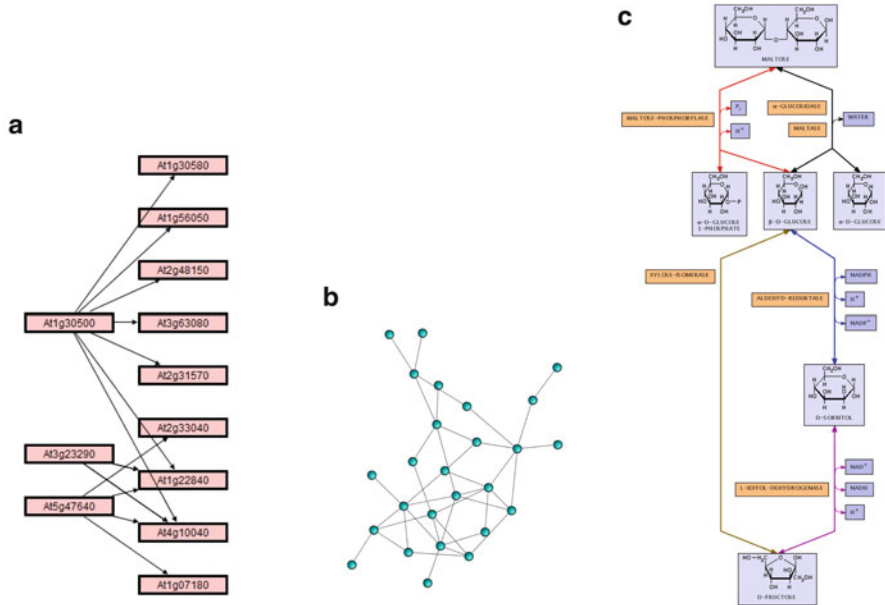


Fig. 7.8 Three sample layouts of biological networks. (a) and (b) were computed and displayed by the *Vanted* system [110]; (c) was computed by *BioPath* [33]. (a) A gene regulatory network (nodes represent genes, edges represent regulation, and labels show gene names). (b) A protein interaction network (nodes represent proteins; edges represent interaction). (c) A metabolic network (nodes represent metabolites, enzymes, and reactions; edges represent consumption and production)

7.3.1 Methods

7.3.1.1 Early Approaches

Driven by the emerging availability of biological networks from databases in the mid-1990s, several groups started to either use existing graph drawing algorithms or design extensions to these algorithms to automatically visualize biological networks. In the following, we present such early work for the three major types of networks from molecular biology.

Signal Transduction and Gene Regulatory Networks

These networks represent regulation or directed interaction between biological entities (such as genes) and are usually modeled as directed graphs; see Fig. 7.8a. There are two widely used methods to visualize such networks: force-based and layered drawings. Several systems provide force-based graph drawing methods for

the visualization of these networks, for example, *PATIKA* [23] and *GeNet* [118]. These tools typically use well-known force-based algorithms such as Eades' algorithm [28], often based on existing layout libraries and systems like *Pajek* [5] or *yFiles* [144]. There are some improvements of the general force-based method to consider application-specific requirements such as the representation of subcellular locations. One example is implemented in the *PATIKA* system.

Signal transduction and gene regulatory networks are directed graphs and, for example, the visualization of the main direction is important to understand the flow of information through the network. Therefore, layered drawing methods are often employed for the computation of maps of these networks. Some tools using this layout method are *TransPath* [85] and *BioConductor* [15]. Often layout libraries for layered drawings such as *dot* [84] are used.

Protein Interaction Networks

These networks represent proteins and their interactions and are modeled as undirected graphs; see Fig. 7.8b. Several systems which employ force-based graph drawing methods for their visualization have been presented, for instance [12,42,98,119]. Also some work on interactive exploration of protein interaction networks has been done, for example, by combining circular and force-based layouts and smooth transitions between subsequent drawings using animation [35].

Metabolic Networks

These networks represent the transformation of metabolites into each other and are usually modeled as directed graphs; see Fig. 7.8c. There are two common approaches to visualizing metabolic networks: force-based and layered drawing methods. Several network analysis tools support force-based layouts, for example, *BioJAKE* [113], *Cytoscape* [119], *PathwayAssist* [101], and *VisANT* [45]. Frequently they visualize not only metabolic but also other types of biological networks. However, force-based approaches mostly do not meet common application-specific requirements. Such requirements are, inter alia, different sizes of nodes, the special placement of co-substances and enzymes, and the general direction of pathways.

Layered drawings are often used as they emphasize the main direction in the network. Tools supporting layered drawings are largely based on existing software libraries. Such solutions show the main direction within networks and partly deal with different node sizes. However, there is no specific placement of co-substances or special pathways such as cycles. Examples are *PathFinder* [40] (which uses the *VCG* library [114]) and *BioMiner* [123] (which employs *yFiles* [144]). The earliest

approach to our knowledge is from Karp and Paley, where the complete network is separated into parts such as trees, paths, and circles, and the parts are laid out separately [62]. Although not a layered drawing algorithm as described in Sect. 7.2.2, it results in an overall layout with some layered structure. Extended layered drawings consider cyclic structures within the network or show pathways of different topology using different layouts, such as the algorithm by Becker and Rojas [6]. An advanced layered drawing algorithm for metabolic networks considering all relevant visualization requirements has been presented in [115].

7.3.1.2 Current Approaches and Tools

There are many challenges in current research of biological network visualization and visual analytics, such as visual analysis of integrated and correlated data, visual comparison of networks, integrated and overlapping networks, graphical representation of paths and flows, and hierarchical networks; see [3, 39]. Consequently, this field has become very research active and, for example, several special algorithms have been presented in the last few years concerning the layout of biological networks. Among them are grid-based methods [81], clustered circular layouts [38], and constraint-based methods [116]. The quality of these specialized layout algorithms is often much better than just applying standard methods, an example is shown in Fig. 7.1.

A broad range of more than 170 tools for the modeling, analysis, and visualization of biological networks is nowadays available on the Internet. These tools change often rapidly, new tools emerge, and old tools obtain new features or are not longer maintained. Therefore, only a small set of some important tools will be listed here. Other reviews are available, for example, Suderman and Hallett in 2007 compared more than 35 tools regarding network and data visualization [128]; Kono et al. compared tools for pathway representation, mapping and editing, and data exchange in 2009 [83]; and Gehlenborg et al. looked at visualization tools for interaction networks and biological pathways in 2010 [39].

The following tools may be of interest to the reader. As the functionality of the tools changes rapidly over time, we do not provide a feature list but encourage the reader to visit the respective tool websites given below:

- *BiNa* [86] (<http://bit.ly/y6ix9i>)
- *BioUML* [82] (<http://bit.ly/yIETIt>)
- *CellDesigner* [36, 37] (<http://bit.ly/A0FQiF>)
- *CellMicrocosmos* [125] (<http://bit.ly/WJ8cnE>)
- *Cytoscape* [119, 124] (<http://bit.ly/wY2sbG>)
- *Omix* [26] (<http://bit.ly/zL52vB>)
- *Ondex* [78, Chap. 5] (<http://bit.ly/AetZjz>)

- *Pathway Projector* [83] (<http://bit.ly/zo5x2M>)
- *PathVisio* [135] (<http://bit.ly/zunwxW>)
- *Vanted* [54, 110] (<http://bit.ly/Aigr0T>)
- *VisAnt* [45, 46] (<http://bit.ly/agZBni>)

7.3.2 SBGN Standard

Biological networks shown in books, articles, and online resources are often difficult to understand as the same biological concept can be shown by using different graphical representations. Therefore, it is time-consuming to get familiar with the graphical notation used, but this also carries the danger of misinterpretation. Consequently, particularly for molecular-biological networks such as gene regulatory, signal transduction, protein interaction, and metabolic networks, there were several attempts to define a uniform representation. This includes Kitano's Process Diagrams [76], Kohn's Molecular Interaction Maps [79], and Michal's representation of metabolic pathways [95]. However, a single map type is often not enough to adequately illustrate the complexity of biological processes, and none of the mentioned attempts has asserted itself as a widely used standard.

Since 2006, there is a new initiative which partly builds on earlier standardization attempts and is closely connected with the successful exchange format *SBML* (System Biology Markup Language) [48]: *SBGN*—the System Biology Graphical Notation [88]. Additional material can be found under <http://sbgn.org>, and formal specifications are available [93, 97, 103]; see the previously mentioned website for the latest version of the specification.

SBGN supports three corresponding views or maps on a biological process: *process description* which describes elements (cellular building blocks like molecules, and nucleic acid sequences but also other information like observable events) and interactions between these elements; *entity relationship* which presents the interaction between biological entities and the influence of entities on other elements; and *activity flow* which focuses on the flow of information from one activity to another. These different language types enable to show different aspects of biological processes. A process description contains, for example, a molecule often several times in different states, e.g., phosphorylated or unphosphorylated, while both other map types show in each case only one occurrence of such a molecule. Figure 7.9 shows two molecular-biological networks in SBGN notation.

There are several tools supporting SBGN, including *CellDesigner* [36], *EPE* (Edinburgh Pathway Editor) [30], *PathVisio* [135], and *SBGN-ED* [21] (an extension of *Vanted* [110]). A comparison has been done by Junker et al. [56]. There is also SBGN support for tool developers [136].

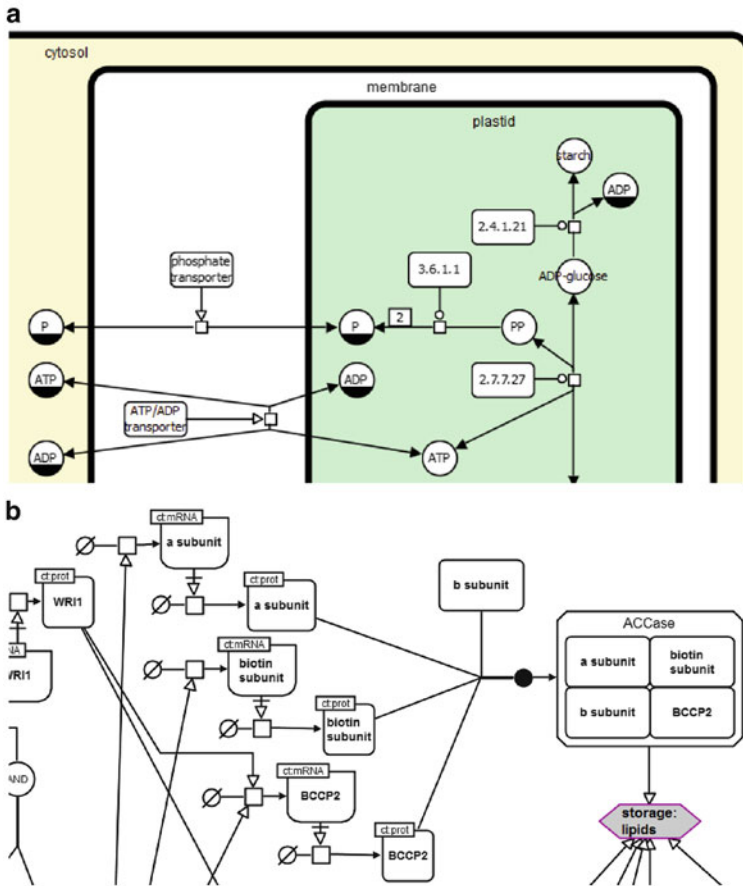


Fig. 7.9 Two examples of SBGN maps. **(a)** Part of a metabolic pathway in SBGN notation (pathway derived from *MetaCrop* [117], an information system based on *Meta-All* [143]). **(b)** Part of a gene regulatory network in SBGN notation (derived from *RIMAS* [55])

References

1. Abello J, van Ham F (2004) Matrix zoom: a visual interface to semi-external graphs. In: Proceedings of the IEEE symposium on information visualization, Austin. IEEE Computer Society, Los Alamitos, Texas, pp 183–190
2. Aigner W, Miksch S, Schumann H, Tominski C (2011) Visualization of time-oriented data. Springer, London/New York
3. Albrecht M, Kerren A, Klein K, Kohlbacher O, Mutzel P, Paul W, Schreiber F, Wybrow M (2010) On open problems in biological network visualization. In: Proceedings of the international symposium on graph drawing (GD '09), Chicago. LNCS, vol 5849. Springer, pp 256–267

4. Appel RD, Bairoch A, Hochstrasser DF (1994) A new generation of information retrieval tools for biologists: the example of the ExpASY WWW server. *Trends Biochem Sci* 19:258–260
5. Batagelj V, Mrvar A (2004) Pajek – analysis and visualization of large networks. In: Jünger M, Mutzel P (eds) *Graph drawing software*. Springer, Berlin/New York, pp 77–103
6. Becker MY, Rojas I (2001) A graph layout algorithm for drawing metabolic pathways. *Bioinformatics* 17(5):461–467
7. Beddow J (1990) Shape coding of multidimensional data on a microcomputer display. In: *Proceedings of the 1st conference on visualization '90, VIS '90, San Francisco*. IEEE Computer Society Press, Los Alamitos, pp 238–246
8. Berg JM, Tymoczko JL, Stryer L (2002) *Biochemistry*. W H Freeman, New York
9. Berge C (1989) *Hypergraphs: the theory of finite sets*. North-Holland, Amsterdam
10. Borisjuk L, Hajirezaei MR, Klukas C, Rolletschek H, Schreiber F (2005) Integrating data from biological experiments into metabolic networks with the DBE information system. *Silico Biol* 5(2):93–102
11. Bostock M Edgar Anderson's Iris data set scatter plot matrix. <http://mbostock.github.com/d3/talk/20111116/iris-splom.html>. Last accessed 13 Mar 2013
12. Breitzkreutz BJ, Stark C, Tyers M (2003) Osprey: a network visualization system. *Genome Biol* 4(3):R22
13. Buchheim C, Jünger M, Leipert S (2002) Improving walker's algorithm to run in linear time. In: *Revised papers from the 10th international symposium on graph drawing, GD '02, Irvine*. Springer, London, pp 344–353
14. Card S, Mackinlay J, Shneiderman B (eds) (1999) *Readings in information visualization: using vision to think*. Morgan Kaufmann, San Francisco
15. Carey VJ, Gentry J, Whalen E, Gentleman R (2005) Network structures and algorithms in BioConductor. *Bioinformatics* 21(1):135–136
16. Chambers JM, Cleveland WS, Kleiner B, Tukey PA (1983) *Graphical methods for data analysis*. Wadsworth, Belmont
17. Chen C (2004) *Information visualization: beyond the horizon*, 2nd edn. Springer, London/Berlin/Heidelberg
18. Chernoff H (1973) The use of faces to represent points in k-dimensional space graphically. *J Am Stat Assoc* 68:361–368
19. Cleveland WC, McGill ME (1988) *Dynamic graphics for statistics*. CRC, Boca Raton
20. Collins C, Viegas F, Wattenberg M (2009) Parallel tag clouds to explore and analyze faceted text corpora. In: *Proceedings of the IEEE symposium on visual analytics science and technology (VAST '09), Atlantic City*. IEEE Computer Society, pp 91–98
21. Czauderna T, Klukas C, Schreiber F (2010) Editing, validating and translating of SBGN maps. *Bioinformatics* 26(18):2340–2341
22. D3. Data-driven documents. <http://d3js.org>. Last accessed 13 Mar 2013
23. Demir E, Babur O, Dogrusöz U, Gürsoy A, Nisanci G, Çetin Atalay R, Ozturk M (2002) PATIKA: an integrated visual environment for collaborative construction and analysis of cellular pathways. *Bioinformatics* 18(7):996–1003
24. Di Battista G, Eades P, Tamassia R, Tollis IG (1999) *Graph drawing: algorithms for the visualization of graphs*. Prentice Hall, Upper Saddle River
25. Diehl S, Görg C, Kerren A (2001) Preserving the mental map using foresighted layout. In: Ebert DS, Favre JM, Peikert R (eds) *Data visualization 2001, Eurographics, Ascona*. Springer, Vienna, pp 175–184
26. Droste P, Miebach S, Niefenführ S, Wiechert W, Nöh K (2011) Visualizing multi-omics data in metabolic networks with the software Omix: a case study. *Biosystems* 105(2):154–161
27. Dykes J, MacEachren AM, Kraak MJ (2005) *Exploring geovisualization*. Pergamon, Oxford
28. Eades P (1984) A heuristic for graph drawing. *Congressus Numerantium* 42:149–160
29. Eiglsperger M, Fekete SP, Klau GW (2001) Orthogonal graph drawing. In: Kaufmann M, Wagner D (eds) *Drawing graphs. Lecture notes in computer science, vol 2025*. Springer, Berlin/Heidelberg, pp 121–171

30. EPE. EPE Edinburgh Pathway editor. <http://epe.sourceforge.net/SourceForge/EPE.html>. Last accessed 02 Aug 2012
31. ExposeData.com. Nutrient contents – parallel coordinates. <http://exposedata.com/parallel/>. Last accessed 13 Mar 2013
32. Feinberg J, Wordle. <http://www.wordle.net>. Last accessed 13 Mar 2013
33. Forster M, Pick A, Raitner M, Schreiber F, Brandenburg FJ (2002) The system architecture of the BioPath system. *Silico Biol* 2(3):415–426
34. Freeman E, Fertig S (1995) Lifestreams: organizing your electronic life. In: AAAI fall symposium on AI applications in knowledge navigation and retrieval, Cambridge. Association for the Advancement of Artificial Intelligence, pp 38–44
35. Friedrich C, Schreiber F (2003) Visualization and navigation methods for typed protein-protein interaction networks. *Appl Bioinform* 2(3 Suppl):19–24
36. Funahashi A, Morohashi M, Kitano H (2003) CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. *Biosilico* 1(5):159–162
37. Funahashi A, Matsuoka Y, Jouraku A, Kitano H, Kikuchi N (2006) CellDesigner: a modeling tool for biochemical networks. In: Proceedings of the 38th conference on winter simulation, winter simulation conference, Monterey, pp 1707–1712
38. Fung D, Wilkins M, Hart D, Hong S (2010) Using the clustered circular layout as an informative method for visualizing protein-protein interaction networks. *Proteomics* 10(14):2723–2727
39. Gehlenborg N, O'Donoghue SI, Baliga NS, Goesmann A, Hibbs MA, Kitano H, Kohlbacher O, Neuweger H, Schneider R, Tenenbaum D, Gavin AC (2010) Visualization of omics data for systems biology. *Nat Methods* 7:S56–S68
40. Goesmann A, Haubrock M, Meyer F, Kalinowski J, Giegerich R (2002) PathFinder: reconstruction and dynamic visualization of metabolic pathways. *Bioinformatics* 18(1):124–129
41. Görg C, Pohl M, Qeli E, Xu K (2007) Visual representations. In: Kerren A, Ebert A, Meyer J (eds) Human-centered visualization environments. LNCS, tutorial, vol 4417. Springer, Berlin, pp 163–230
42. Han K, Ju BH, Park JH (2002) InterViewer: dynamic visualization of protein-protein interactions. In: Kobourov SG, Goodrich MT (eds) Proceedings of the international symposium on graph drawing (GD '02), Irvine. LNCS, vol 2528. Springer, pp 364–365
43. Heer J, Shneiderman B (2012) Interactive dynamics for visual analysis. *Commun ACM* 55(4):45–54
44. Henry N, Fekete JD, McGuffin MJ (2007) Nodetrix: a hybrid visualization of social networks. *IEEE Trans Vis Comput Graph* 13:1302–1309
45. Hu Z, Mellor J, Wu J, DeLisi C (2004) VisANT: an online visualization and analysis tool for biological interaction data. *BMC Bioinform* 5(1):e17
46. Hu Z, Hung JH, Wang Y, Chang YC, Huang CL, Huyck M, DeLisi C (2009) VisANT 3.5: multi-scale network visualization, analysis and inference based on the gene ontology. *Nucl Acids Res* 37(Web Server issue):W115–W121
47. Huang W, Eades P, Hong SH (2009) A graph reading behavior: geodesic-path tendency. In: Proceedings of the IEEE Pacific visualization symposium, 2009 (PacificVis '09), Beijing, pp 137–144
48. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, Cuellar AA, Dronov S, Gilles ED, Ginkel M, Gor V, Goryanin II, Hedley WJ, Hodgman TC, Hofmeyr JH, Hunter PJ, Juty NS, Kasberger JL, Kremling A, Kummer U, Le Novère N, Loew LM, Lucio D, Mendes P, Minch E, Mjolsness ED, Nakayama Y, Nelson MR, Nielsen PF, Sakurada T, Schaff JC, Shapiro BE, Shimizu TS, Spence HD, Stelling J, Takahashi K, Tomita M, Wagner J, Wang J (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* 19:524–531
49. Inselberg A, Dimsdale B (1990) Parallel coordinates: a tool for visualizing multi-dimensional geometry. In: Proceedings of the IEEE conference on visualization (Vis '90), San Francisco. IEEE Computer Society, pp 361–378

50. Javed W, Elmqvist N (2012) Exploring the design space of composite visualization. In: Proceedings of the IEEE Pacific symposium on visualization (PacificVis '12), Songdo. IEEE Computer Society Press, pp 1–8
51. Jerding DF, Stasko JT (1998) The information mural: a technique for displaying and navigating large information spaces. *IEEE Trans Vis Comput Graph* 4(3):257–271
52. Johnson B, Shneiderman B (1991) Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In: Proceedings of the 2nd conference on visualization (Vis '91), San Diego. IEEE Computer Society Press, Los Alamitos, pp 284–291
53. Jolliffe I (2002) Principal component analysis. Springer, New York
54. Junker BH, Klukas C, Schreiber F (2006) VANTED: a system for advanced data analysis and visualization in the context of biological networks. *BMC Bioinform* 7:109
55. Junker A, Hartmann A, Schreiber F, Bäumlein H (2010) An engineer's view on regulation of seed development. *Trends Plant Sci* 15(6):303–307
56. Junker A, Rohn H, Czauderna T, Klukas C, Hartmann A, Schreiber F (2012) Creating interactive, web-based and data-enriched maps using the systems biology graphical notation. *Nat Protoc* 7:579–593
57. Jusufi I (2012) Towards the visualization of multivariate biochemical networks. Licentiate thesis, Linnaeus University
58. Jusufi I, Dingjie Y, Kerren A (2010) The network lens: interactive exploration of multivariate networks using visual filtering. In: Proceedings of the 14th international conference on information visualisation (IV '10), London. IEEE Computer Society Press, pp 35–42
59. Jusufi I, Kerren A, Aleksakhin V, Schreiber F (2012) Visualization of mappings between the gene ontology and cluster trees. In: Proceedings of the SPIE 2012 conference on visualization and data analysis (VDA '12), IS&T/SPIE, Burlingame. SPIE, vol 8294, pp 8294–20
60. Jusufi I, Klukas C, Kerren A, Schreiber F (2012) Guiding the interactive exploration of metabolic pathway interconnections. *Inf Vis* 11(2):136–150
61. Kanehisa M, Goto S, Kawashima S, Nakaya A (2002) The KEGG databases at GenomeNet. *Nucl Acids Res* 30(1):42–46
62. Karp PD, Paley SM (1994) Automated drawing of metabolic pathways. In: Lim H, Cantor C, Bobbins R (eds) Proceedings of the international conference on bioinformatics and genome research, Tallahassee, pp 225–238
63. Kaser O, Lemire D (2007) Tag-cloud drawing: algorithms for cloud visualization. In: Proceedings of tagging and metadata for social information organization (WWW '07), Banff
64. Kaufmann M, Wagner D (1999) Drawing graphs: methods and models. Lecture notes in computer science, tutorial, vol 2025. Springer, Berlin/Heidelberg
65. Keim DA (2002) Information visualization and visual data mining. *IEEE Trans Vis Comput Graph* 7(1):1–8
66. Keim D, Kriegel HP (1994) Visdb: database exploration using multidimensional visualization. *IEEE Comput Graph Appl* 14(5):40–49
67. Keim D, Oelke D (2007) Literature fingerprinting: a new method for visual literary analysis. In: Proceedings of the IEEE symposium on visual analytics science and technology (VAST '07), Sacramento. IEEE Computer Society Press, pp 115–122
68. Keim D, Andrienko G, Fekete JD, Görg C, Kohlhammer J, Melançon G (2008) Visual analytics: definition, process, and challenges. In: Kerren A, Stasko JT, Fekete JD, North C (eds) Information visualization: human-centered issues and perspectives. Lecture notes in computer science, vol 4950. Springer, Berlin/Heidelberg, pp 154–175
69. Keim D, Kohlhammer J, Ellis G, Mansmann F (eds) (2010) Mastering the information age – solving problems with visual analytics. Eurographics Digital Library, Goslar
70. Kerren A, Schreiber F (2012) Toward the role of interaction in visual analytics. In: Proceedings of the winter simulation conference, winter simulation conference, WSC '12, Berlin, pp 420:1–420:13
71. Kerren A, Ebert A, Meyer J (eds) (2007) Human-centered visualization environments. LNCS, tutorial, vol 4417. Springer, Berlin

72. Kerren A, Ebert A, Meyer J (2007) Introduction to human-centered visualization environments. In: Kerren A, Ebert A, Meyer J (eds) *Human-centered visualization environments*. LNCS, tutorial, vol 4417. Springer, Berlin, pp 1–9
73. Kerren A, Stasko JT, Fekete JD, North C (2007) Workshop report: information visualization human-centered issues in visual representation, interaction, and evaluation. *Inf Vis* 6(3):189–196
74. Kerren A, Stasko JT, Fekete JD, North C (eds) (2008) *Information visualization: human-centered issues and perspectives*. Lecture notes in computer science, vol 4950. Springer, Berlin/Heidelberg
75. Kerren A, Köstinger H, Zimmer B (2012) Vincent – visualisation of network centralities. In: *Proceedings of the international conference on information visualization theory and applications (IVAPP '12)*, INSTICC, Rome, pp 703–712
76. Kitano H (2003) A graphical notation for biochemical networks. *Biosilico* 1(5):169–176
77. Koh K, Lee B, Kim B, Seo J (2010) Maniwordle: providing flexible control over wordle. *IEEE Trans Vis Comput Graph* 16:1190–1197
78. Köhler J, Baumbach J, Taubert J, Specht M, Skusa A, Rüegg A, Rawlings C, Verrier P, Philippi S (2006) Graph-based analysis and visualization of experimental results with ONDEX. *Bioinformatics* 22(11):1383–1390
79. Kohn KW, Aladjem MI (2006) Circuit diagrams for biological networks. *Mol Syst Biol* 2:e2006.0002
80. Kohonen T, Schroeder MR, Huang TS (eds) (2001) *Self-organizing maps*, 3rd edn. Springer, New York/Secaucus
81. Kojima K, Nagasaki M, Jeong E, Kato M, Miyano S (2007) An efficient grid layout algorithm for biological networks utilizing various biological attributes. *BMC Bioinform* 8:76
82. Kolpakov FA (2002) BioUML – framework for visual modeling and simulation of biological systems. In: *Proceedings of the international conference on bioinformatics of genome regulation and structure*, Novosibirsk. Springer, pp 130–133
83. Kono N, Arakawa K, Ogawa R, Kido N, Oshita K, Ikegami K, Tamaki S, Tomit M (2009) Pathway projector: web-based zoomable pathway browser using KEGG Atlas and Google maps API. *PLoS ONE* 4(11):e7710
84. Koutsofios E, North S (1995) *Drawing graphs with dot*. Technical report, AT&T Bell Laboratories, Murray Hill
85. Krull M, Voss N, Choi C, Pistor S, Potapov A, Wingender E (2003) TRANSPATH: an integrated database on signal transduction and a tool for array analysis. *Nucl Acids Res* 31(1):97–100
86. Küntzer J, Backes C, Blum T, Gerasch A, Kaufmann M, Kohlbacher O, Lenhof HP (2007) BNDB – the biochemical network database. *BMC Bioinform* 8:367
87. Lee B, Riche N, Karlson A, Carpendale S (2010) Sparkclouds: visualizing trends in tag clouds. *IEEE Trans Vis Comput Graph* 16(6):1182–1189
88. Le Novère N, Hucka M, Mi H, Moodie S, Schreiber F, Sorokin A, Demir E, Wegner K, Aladjem MI, Wimalaratne SM, Bergman FT, Gauges R, Ghazal P, Kawaji H, Li L, Matsuoka Y, Villéger A, Boyd SE, Calzone L, Courtot M, Dogrusoz U, Freeman TC, Funahashi A, Ghosh S, Jouraku A, Kim S, Kolpakov F, Luna A, Sahle S, Schmidt E, Watterson S, Wu G, Goryanin I, Kell DB, Sander C, Sauro H, Snoep JL, Kohn K, Kitano H (2009) The systems biology graphical notation. *Nat Biotechnol* 27(8):735–741
89. Li Q, Bao X, Song C, Zhang J, North C (2003) Dynamic query sliders vs. brushing histograms. In: *CHI '03 extended abstracts on human factors in computing systems*, CHI EA '03, Fort Lauderdale. ACM, New York, pp 834–835
90. Liu J (2012) *Visualization of weather data: temperature trend visualization*. Bachelor's thesis, School of Computer Science, Physics and Mathematics, Linnaeus University, Växjö
91. MacNeil S, Elmqvist N (2013) Visualization mosaics for multivariate visual exploration. *Comput Graph Forum* 32:38–50
92. Mardia KV (1979) *Multivariate analysis*. Academic, London/New York

93. Mi H, Schreiber F, Novère NL, Moodie S, Sorokin A (2009) Systems biology graphical notation: activity flow language level 1. *Nat Preced.* doi:10.1038/npre.2009.3724.1
94. Michal G (1993) Biochemical pathways (Poster). Boehringer Mannheim, Mannheim
95. Michal G (1998) On representation of metabolic pathways. *BioSystems* 47:1–7
96. Michal G (1999) Biochemical pathways. Spektrum Akademischer Verlag, Heidelberg
97. Moodie S, Novère NL, Sorokin A, Mi H, Schreiber F (2009) Systems biology graphical notation: process description language level 1. *Nat Preced.* doi:10.1038/npre.2009.3721.1
98. Mrowka R (2001) A Java applet for visualizing protein-protein interaction. *Bioinformatics* 17(7):669–670
99. Nicholson DE (1997) Metabolic pathways map (Poster). Sigma Chemical Co., St. Louis
100. Nightingale F (1858) Notes on matters affecting the health, efficiency, and hospital administration of the British Army. Harrison & Sons, London
101. Nikitin A, Egorov S, Daraselia N, Mazo I (2003) Pathway studio – the analysis and navigation of molecular networks. *Bioinformatics* 19(16):2155–2157
102. Nöllenburg M (2007) Geographic visualization. In: Kerren A, Ebert A, Meyer J (eds) Human-centered visualization environments. LNCS, tutorial, vol 4417. Springer, Berlin, pp 257–294
103. Novère NL, Moodie S, Sorokin A, Schreiber F, Mi H (2009) Systems biology graphical notation: entity relationship language level 1. *Nat Preced.* doi:10.1038/npre.2009.3719.1
104. Oesterling P, Scheuermann G, Teresniak S, Heyer G, Koch S, Ertl T, Weber G (2010) Two-stage framework for a topology-based projection and visualization of classified document collections. In: Proceedings of the IEEE symposium on visual analytics science and technology (VAST '10), Salt Lake City. IEEE Computer Society, pp 91–98
105. O'Madadhain J, Fisher D, Nelson T. JUNG – Java universal network/graph framework. <http://jung.sourceforge.net/>. Last accessed 27 Jan 2013
106. Pickett RM, Grinstein GG (1988) Iconographic displays for visualizing multidimensional data. In: Proceedings of the 1988 IEEE international conference on systems, man, and cybernetics, Beijing, vol 1, pp 514–519
107. Reingold EM, Tilford JS (1981) Tidier drawing of trees. *IEEE Trans Softw Eng* 7(2):223–228
108. Richard JS, Catrambone R, Guzdial M, McDonald K (2000) An evaluation of space-filling information visualizations for depicting hierarchical structures. *Int J Hum Comput Stud* 53:663–694
109. Roberts JC (2004) Exploratory visualization with multiple linked views. In: MacEachren A, Kraak MJ, Dykes J (eds) Exploring geovisualization. Elsevier, Amsterdam
110. Rohn H, Junker A, Hartmann A, Grafahrend-Belau E, Treutler H, Klapperstück M, Czauderna T, Klukas C, Schreiber F (2012) VANTED v2: a framework for systems biology applications. *BMC Syst Biol* 6:139
111. Rohrschneider M, Heine C, Reichenbach A, Kerren A, Scheuermann G (2010) A novel grid-based visualization approach for metabolic networks with advanced focus & context view. In: Proceedings of the international symposium on graph drawing (GD '09), Chicago. LNCS, vol 5849. Springer, pp 268–279
112. Rohrschneider M, Ullrich A, Kerren A, Stadler PF, Scheuermann G (2010) Visual network analysis of dynamic metabolic pathways. In: Proceedings of the 6th international conference on advances in visual computing – volume Part I (ISVC '10), Las Vegas. Springer, Berlin/Heidelberg, pp 316–327
113. Salamonsen W, Mok KY, Kolatkar P, Subbiah S (1999) BioJAKE: a tool for the creation, visualization and manipulation of metabolic pathways. In: Proceedings of the Pacific symposium on biocomputing, Big Island, pp 392–400
114. Sander G (1994) Graph layout through the VCG tool. In: Tamassia R, Tollis IG (eds) Proceedings of the DIMACS international workshop on graph drawing (GD '94), Princeton. Springer, pp 194–205
115. Schreiber F (2002) High quality visualization of biochemical pathways in BioPath. *Silico Biol* 2(2):59–73
116. Schreiber F, Dwyer T, Marriott K, Wybrow M (2009) A generic algorithm for layout of biological networks. *BMC Bioinform* 10:375

117. Schreiber F, Colmsee C, Czauderna T, Grafahrend-Belau E, Hartmann A, Junker A, Junker BH, Klapperstück M, Scholz U, Weise S (2012) MetaCrop 2.0: managing and exploring information about crop plant metabolism. *Nucl Acids Res* 40(1):D1173–D1177
118. Serov VN, Spirov AV, Samsonova MG (1998) Graphical interface to the genetic network database GeNet. *Bioinformatics* 14(6):546–547
119. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res* 13(11):2498–2504
120. Shannon R, Holland T, Quigley A (2008) Multivariate graph drawing using parallel coordinate visualisations. Technical report 2008-6, School of Computer Science and Informatics, University College Dublin
121. Shneiderman B (1996) The eyes have it: a task by data type taxonomy for information visualizations. In: *Proceedings of the IEEE symposium on visual languages (VL '96)*, Boulder. IEEE Computer Society, pp 336–343
122. Shneiderman B, Aris A (2006) Network visualization by semantic substrates. *IEEE Trans Vis Comput Graph* 12:733–740
123. Sirava M, Schäfer T, Eiglsperger M, Kaufmann M, Kohlbacher O, Bornberg-Bauer E, Lenhof HP (2002) BioMiner – modeling, analyzing, and visualizing biochemical pathways and networks. *Bioinformatics* 18(Suppl. 2):S219–S230
124. Smoot ME, Ono K, Ruscheinski J, Wang PL, Ideker T (2011) Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics* 27(3):431–432
125. Sommer B, Künsemöller J, Sand N, Husemann A, Rummig M, Kormeier B (2010) Cellmicrocosmos 4.1 – an interactive approach to integrating spatially localized metabolic networks into a virtual 3d cell environment. In: Fred ALN, Filipe J, Gamboa H (eds) *Proceedings of the first international conference on bioinformatics (BIOINFORMATICS '10)*, Valencia, pp 90–95
126. Spence R (2007) *Information visualization: design for interaction*, 2nd edn. Prentice Hall, Harlow
127. Stasko J, Muthukumarasamy J (1996) Visualizing program executions on large data sets. In: *Proceedings of the IEEE symposium on visual languages (VL '96)*, Boulder. IEEE Computer Society, pp 166–173
128. Suderman M, Hallett MT (2007) Tools for visually exploring biological networks. *Bioinformatics* 23(20):2651–2659
129. Sugiyama K, Tagawa S, Toda M (1981) Methods for visual understanding of hierarchical system structures. *IEEE Trans Syst Man Cybern SMC-11(2)*:109–125
130. Thomas JJ, Cook KA (2006) A visual analytics agenda. *IEEE Comput Graph Appl* 26(1):10–13
131. Tufte ER (1990) *Envisioning information*. Graphics Press, Cheshire
132. Tufte ER (1997) *Visual explanations: images and quantities, evidence and narrative*. Graphic Press, Cheshire
133. Tufte ER (2001) *The visual display of quantitative information*, 2nd edn. Graphics Press, Cheshire
134. van Ham F, van Wijk JJ (2002) Beamtrees: compact visualization of large hierarchies. In: *Proceedings of the IEEE symposium on information visualization (InfoVis '02)*, Boston. IEEE Computer Society, pp 93–100
135. van Iersel MP, Kelder T, Pico AR, Hanspers K, Coort S, Conklin BR, Evelo C (2008) Presenting and exploring biological pathways with PathVisio. *BMC Bioinform* 9:399. 1–9
136. van Iersel MP, Villéger A, Czauderna T, Boyd SE, Bergmann FT, Luna A, Demir E, Sorokin AA, Dogrusöz U, Matsuoka Y, Funahashi A, Aladjem MI, Mi H, Moodie SL, Kitano H, Novère NL, Schreiber F (2012) Software support for SBGN maps: SBGN-ML and LibSBGN. *Bioinformatics* 28(15):2016–2021
137. Van Wijk JJ, Nuij WAA (2003) Smooth and efficient zooming and panning. In: *Proceedings of the IEEE conference on information visualization (InfoVis '03)*, Seattle. IEEE Computer Society, Washington, DC, pp 15–22

138. Viegas FB, Wattenberg M, Feinberg J (2009) Participatory visualization with wordle. *IEEE Trans Vis Comput Graph* 15:1137–1144
139. Walker JQ (1990) A node-positioning algorithm for general trees. *Softw Pract Exp* 20(7):685–705
140. Ward M, Grinstein G, Keim DA (2010) *Interactive data visualization: foundations, techniques, and application*. A.K. Peters, Natick
141. Ware C (2004) *Information visualization: perception for design*, 2nd edn. Morgan Kaufmann, San Francisco
142. Wattenberg M (2006) Visual exploration of multivariate graphs. In: *Proceedings of the SIGCHI conference on human factors in computing systems (CHI '06)*, Montreal. ACM, New York, pp 811–819
143. Weise S, Grosse I, Klukas C, Koschützki D, Scholz U, Schreiber F, Junker BH (2006) Meta-All: a system for managing metabolic pathway information. *BMC Bioinform* 7:465
144. Wiese R, Eiglsperger M, Kaufmann M (2001) yFiles: visualization and automatic layout of graphs. In: Mutzel P, Jünger M, Leipert S (eds) *Proceedings of the international symposium on graph drawing (GD '01)*, Vienna. LNCS, vol 2265. Springer, pp 453–454
145. Williams M, Munzner T (2004) Steerable, progressive multidimensional scaling. In: *Proceedings of the IEEE symposium on information visualization (InfoVis '04)*, Austin. IEEE Computer Society Press, pp 57–64
146. Williamson C, Shneiderman B (1992) The dynamic homefinder: evaluating dynamic queries in a real-estate information exploration system. In: *Proceedings of the international ACM conference on research and development in information retrieval (SIGIR '92)*, Copenhagen. ACM, New York, pp 338–346
147. Wise J, Thomas J, Pennock K, Lantrip D, Pottier M, Schur A, Crow V (1995) Visualizing the non-visual: spatial analysis and interaction with information from text documents. In: *Proceedings of the IEEE symposium on information visualization (InfoVis '95)*, Atlanta. IEEE Computer Society, pp 51–58
148. Yi JS, Kang YA, Stasko J, Jacko J (2007) Toward a deeper understanding of the role of interaction in information visualization. *IEEE Trans Vis Comput Graph* 13(6):1224–1231
149. yWorks. yEd graph editor. http://www.yworks.com/en/products_yed_about.html. Last accessed 02 Aug 2012
150. Zimmer B, Jusufi I, Kerren A (2012) Analyzing multiple network centralities with ViNCent. In: *Proceedings of SIGRAD 2012: interactive visual analysis of data*, Växjö, 29–30 Nov 2012. Number 81 in Linköping electronic conference proceedings. Linköping University Electronic Press, pp 87–90