9

Heterogeneous Networks on Multiple Levels

Falk Schreiber, Andreas Kerren, Katy Börner, Hans Hagen, and Dirk Zeckzer

At any moment in time, we are driven by and are an integral part of many interconnected, dynamically changing networks. Our species has evolved as part of diverse ecological, biological, social, and other networks over thousands of years. As part of a complex food web, we learned how to find prey and to avoid predators. We have created advanced socio-technical environments in the shape of cities, water and power networks, streets, and airline systems. In 1969, people started to interlink computers leading to the largest and most widely used networked communication infrastructure in existence today: the Internet.

Often, the complex structure of networks is influenced by system-dependent local constraints on the node interconnectivity. Node characteristics may vary over time and there might be many different types of nodes. The links[1] between nodes might be directed or undirected, and might have weights and/or additional properties that might change over time. Many natural systems never reach a steady state and non-equilibrium models need to be applied to characterize their behavior.

As a result of this usually domain-specific complexity, analysts are not only confronted with large multivariate networks. In practice, those networks can be assigned to different *levels* (or *scales*), and it is absolutely possible that several different networks share the same level. In case a set of networks share the same level, several notions of those networks can be found in the literature: they reach from *multimodal networks*, *network of networks* to *heterogeneous networks*. For simplicity, we use the latter term for the remainder of this chapter.

Complex structures of heterogeneous networks distributed over various levels lead to considerable visualization and analysis problems. First, the possibly tremendous size and complexity (in terms of higher dimensionality of the node/edge attributes) of those networks form a challenge of itself that is also discussed by other chapters of this book. Second and more impor-

---

[1] We use the terms *link* and *edge* interchangeably.

In A. Kerren et al. (Eds.), Multivariate Network Visualization, LNCS 8380, Chapter 9, pages 175-206, Springer, 2014.

tant for this chapter, there are versatile relationships between the networks and/or between network elements across the different levels. In many application fields, it is essential to get a detailed understanding of such structures. In systems biology, for instance, networks are *the* key concept to structure and combine data that was generated by so-called high-throughput analysis methods. They can be arranged within a hierarchy of levels, from molecular-biological networks to evolutionary networks. The molecular-biological level, for example, contains a set of heterogeneous networks such as metabolic and gene regulatory networks. For a biologist, it is interesting to see how the different elements in those networks are connected with each other. Here, also multivariate data plays an important role as the network elements carry additional multidimensional information, such as experimental data that changes over time.

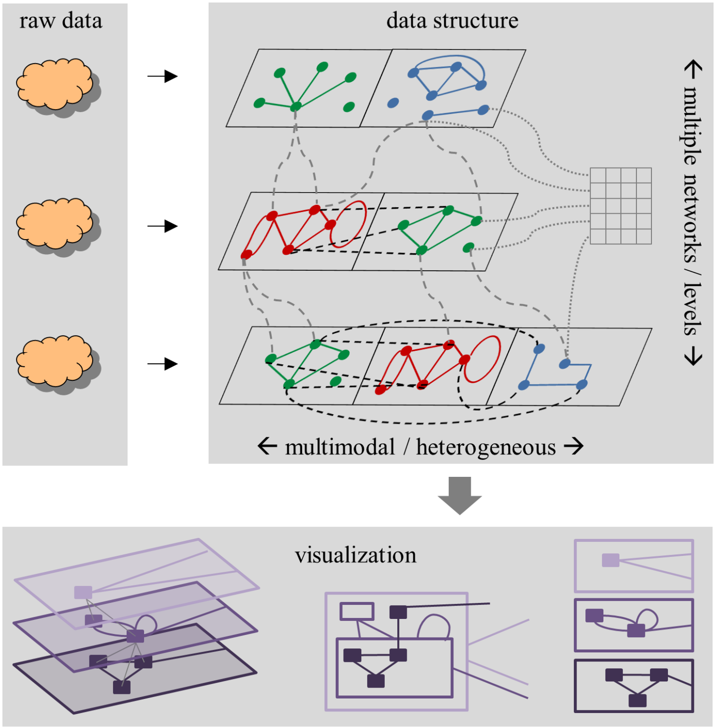


Fig. 9.1. An overview of the various topics of this chapter. Colors in the top right part are used to separate different graphs (e. g., within one level); colors in the bottom part separate different levels. Note that some nodes in the upper part have attached additional attributes symbolized by the small grey data matrix on the right hand side.

In terms of visualization, we want to navigate and explore through this world of networks, and a visualization tool should also provide techniques for the visual analysis of the multivariate data sets together with the underlying network topology. The upper part of Fig. 9.1 provides an overview of these concepts.

Many interactive visualization approaches and tools have been developed for the visual analysis of graphs/networks; the same is true for the visualization of multivariate data sets. We do not give a comprehensive overview of the literature here. Instead, we refer to textbooks, such as [38, 69, 70], surveys [24, 37, 46] and the introduction chapter of this book (Chapter 1). Note that the given literature references only point to selected example works and make no claim to be complete. However, a number of visualizations were especially developed for the analysis of multivariate networks [31]. Some of them are based on coordinated views, such as Jigsaw [67] or enRoute [56], others provide integrated solutions, for instance MobiVis [62], GEOMI [17], Vanted [60], or ViNCent [35, 72], and there are also approaches which realize so-called attribute-driven topologies like JauntyNets [32] and GraphDice [6]. We refer the reader back to Chapter 1 for a more detailed discussion of existing directions in multivariate network visualization. To the best of our knowledge there are no efficient visualization approaches for heterogeneous networks distributed over various levels. The aim of this chapter is to formalize and highlight the underlying problems and challenges by means of three application domains as well as to propose different solutions.

The remainder of this chapter is organized as follows: Sect. 9.1 provides a more formal specification of the data structures used in the chapter. Then, Sect. 9.2 exemplifies the visualization and analysis challenges by means of three important application fields: biology, social sciences, and software engineering. As there are almost no efficient visualization tools for multiple networks over levels available, we provide some ideas and visualization challenges in Sect. 9.3 (cf. the lower part of Fig. 9.1).

9.1 Formal Description of Used Data Structures

In order to facilitate the understanding and description of the following sections, we introduce a formal specification of the data structures used (a complete data structure sample is shown in Fig. 9.2). For this, let $G = (V, E, L)$ be a labeled graph with a finite set of nodes[2] $V = \{v_1, \ldots, v_n\}$, a finite set of edges $E = \{(v_i, v_j) | v_i, v_j \in V\}$, and a finite set of node and edge labels $L = \{l_1, \ldots, l_p\}$. Each node or edge of the graph is required to have a not necessarily unique label, and $l : V, E \rightarrow L$ gives the label for each node or edge. The label is used to encode a vector of additional node-/edge-specific data (i. e., the multivariate attributes).

---

[2] Often called *vertices*; therefore, the variable name v has been established to describe a node.

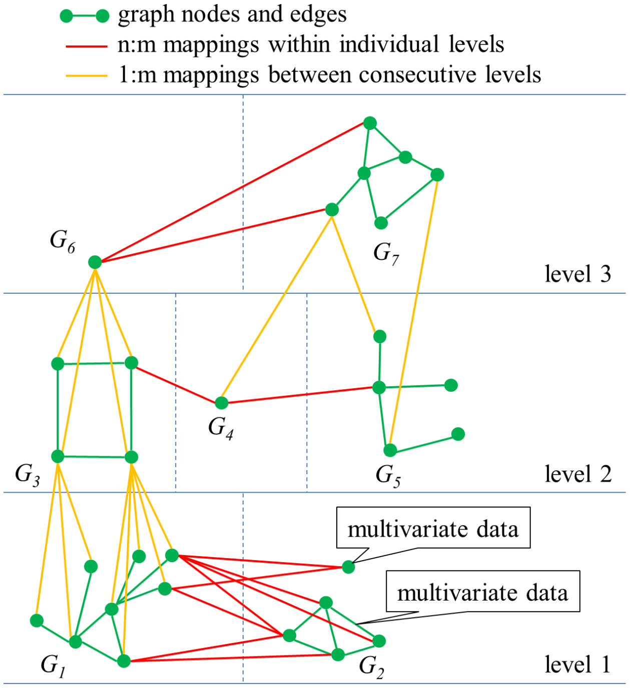


Fig. 9.2. The data structure described in Sect. 9.1. It shows three levels with two heterogeneous networks G_1, G_2 in level 1, three heterogeneous networks G_3, G_4, G_5 in level 2, and two heterogeneous networks G_6, G_7 in level 3.

Let $G_1, \ldots, G_m$ be a set of labeled graphs with $G_i = (V_i, E_i, L_i)$. Each graph may be directed, undirected, or mixed, and represents a specific network of the application domain (note that a graph might also be unconnected). For example, in the biological domain (see Sect. 9.2.1) a gene regulatory network may be represented by a directed graph, a protein interaction network by an undirected graph, and a metabolic network by a directed bipartite graph, respectively (cf. the green colored graphs in Fig. 9.2).

As mentioned in the introduction, we allow networks arranged at various levels. For modeling this property, let $S = \{s_1, \ldots, s_k\}$ be a set of consecutive levels. Each level can contain several graphs of $G_1, \ldots, G_m$, but each graph belongs to only one level. A level therefore groups graphs. The function $s : G, V, E \rightarrow S$ gives the level for each graph, node, or edge.

To connect graphs with each other, we introduce mappings between nodes of different graphs. Let $M = \{(v_i, v_j) | v_i \in V_i, v_j \in V_j\}$ be a mapping which connects a node from graph G_i with one node from a different graph G_j. Note that mappings within a graph are not allowed (those "intragraph mappings" are represented by the normal edges). Furthermore, the resulting structure could be seen as a new (global or union) graph G_G which merges all nodes and edges from $G_1, \ldots, G_m$ and the mappings (edges) M.

For simplicity, the mapping M will be restricted in the following way. For $m = (v_i, v_j)$ with $v_i \in V_i$ and $v_j \in V_j$: if both nodes belong to the same level—$s(v_i) = s(v_j)$; see the red links in the figure—there will be no restriction. However, if both nodes belong to different levels (yellow edges), a mapping is only allowed if the following two conditions hold:

1. both levels are consecutive (neighboring) levels, i. e., $s(v_i) = s(v_j) - 1$ or $s(v_j) = s(v_i) - 1$, and
2. there is a $1 : n$ mapping from the higher to the lower level—i. e., if $s(v_i)$ is the higher level ($s(v_i) = s(v_j) + 1$) then there is no other node v_k in level $s(v_i)$ with $(v_k, v_j) \in M$.

9.2 Application Domains

This section provides an overview of the most important visualization and analysis challenges by means of three application fields: life sciences / biology, social sciences, and software engineering.

9.2.1 Life Sciences / Biology

Biological processes are commonly represented as networks. Examples of biological networks are molecular biological networks such as protein interaction networks (showing the interaction possibilities of proteins) and metabolic pathways (representing the transformation of metabolites into other metabolites), food webs and ecological networks (showing the dependencies between prey and predators), and phylogenetic networks (representing the evolutionary relationships between species).

In the remaining of this section we first present a domain overview introducing relevant terminology along the way. The next subsection discusses major data sources and formats. Then we illuminate diverse network types and interlinkages together with examples. The next subsection presents concrete use cases that benefit from the formalization presented in the previous section and the visualization solutions discussed later. We conclude with a discussion of challenges for multi-level network analysis and visualization in the life / biological sciences.

Overview

A better understanding of biological networks helps in making sense of much of the complex data which is nowadays available in biology, biochemistry, medicine, and related areas of the life sciences. Visualization is a key method to foster exploration and understanding, and, therefore, biological network visualizations have existed for a long time. The importance of visualization and visual analysis of these networks is also evidenced by the large number of books, tools and databases that either contain manually produced drawings of biological processes and networks, or provide algorithms for their automatic layout. Many tools are available, some comparisons of tools have been presented (for example, in [22, 43, 60]), and a number of well known tools supporting network visualization and analysis are:[3]

- *BiNa* [45] (`http://bit.ly/y6ix9i`)
- *BioUML* [42] (`http://bit.ly/yIETIt`)
- *CellDesigner* [20] (`http://bit.ly/A0FQiF`)
- *CellMicrocosmos* [66] (`http://bit.ly/WJ8cnE`)
- *Cytoscape* [65] (`http://bit.ly/wY2sbG`)
- *Ondex* [41, Chapt. 5] (`http://bit.ly/AetZjz`)
- *Pathway Projector* [43] (`http://bit.ly/zo5x2M`)
- *PathVisio* [27] (`http://bit.ly/zunwxW`)
- *SBGN-ED* [13] (`http://bit.ly/17m7KfW`)
- *Vanted* [30] (`http://bit.ly/Aigr0T`)
- *VisAnt* [25] (`http://bit.ly/agZBni`)

Data sources

Biological networks may be directly derived from experimental data (such as protein interaction networks) or are built based on knowledge (such as metabolic networks). There are many data sources for biological networks: from databases covering a specific domain for a specific species (such as AraCyc [54] for metabolism in Arabidopsis) to a specific domain for a set of species (such as MetaCrop [61] for metabolism in crop plants) to several domains for several species (such as KEGG [34] for metabolic and signalling processes in a wide range of species). Another important criterion is the quality of the data which can range from completely manually curated, high-quality data to computationally derived, uncurated data. For overviews of databases for a range of data domains, see [4, 18], for instance.

To support exchange between tools and databases, a few standard representations are widely used such as SBML [26] and BioPAX [14]. Also the graphical representation of cellular processes and biological networks has been standardized with SBGN [48] which helps in understanding the complex processes due to the unambiguous use of glyphs. For details, see the specifications of the SBGN languages [51, 53, 55].

---

[3] Note that there are more than 170 tools available for network visualization and analysis, and a complete listing and comparison is beyond the scope of this article. We list some tools here which exists since several years and often allow easy extensions via plugin mechanisms.

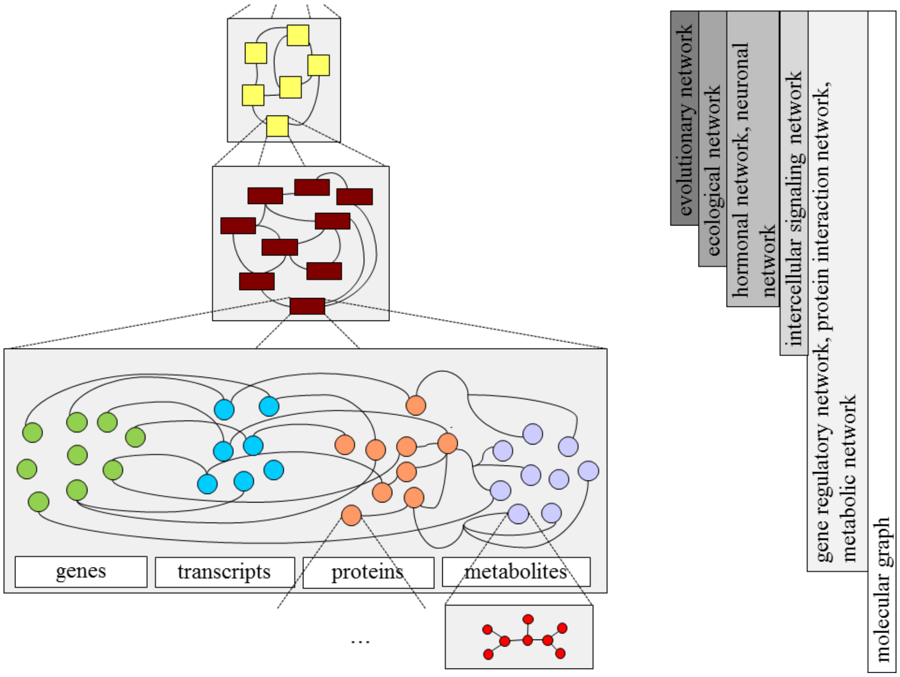


Fig. 9.3. A hierarchy of biological networks.

Network types and examples

Several types of graphs are used to represent biological networks and some typical examples are presented in the following. *Directed graphs*: gene regulatory and signaling networks which describe how genes can be activated or repressed and therefore which proteins are produced in a cell at a particular time; food webs which model the relationships between species in an ecological system. *Undirected or mixed graphs*: protein interaction networks which represent the interaction between proteins such as the building of protein complexes and the activation of one protein by another protein. *Hypergraphs or bipartite graphs*: metabolic networks which show how metabolites are transformed—for example, to produce energy or synthesize substances. *Trees:* phylogenetic trees which are commonly built on information from molecular biology such as DNA or protein sequences and which represent the ancestral relationships between different species. There exists a *hierarchy of biological networks* (see Fig. 9.3 and also Chap. 4).

An example of the data structure described in Sect. 9.1 is given in Fig. 9.4. Here, three molecular-biological networks (gene-regulatory network, protein interaction network, and metabolic network) are presented on level 1. Genes in the gene regulatory network may activate or inactivate the transcription of other genes. Genes are transcribed into proteins, therefore the protein interaction network does not only contain edges between nodes of the protein interaction network (for representing interaction), but also (red) edges between genes (gene regulatory network) and proteins (protein interaction network). Finally the metabolic network is a bipartite graph consisting of metabolites (circles) and enzymes (rectangles). Enzymes are proteins; therefore, there are (red) edges connecting the protein interaction graph with the metabolism graph. On the next level, two networks are represented: an association of all genes to their chromosomes and a clustering of all proteins into disjunct clusters. The yellow edges represent which gene of the gene regulatory network belongs to which chromosome, and which protein of the protein interaction network belongs to which protein cluster, respectively.

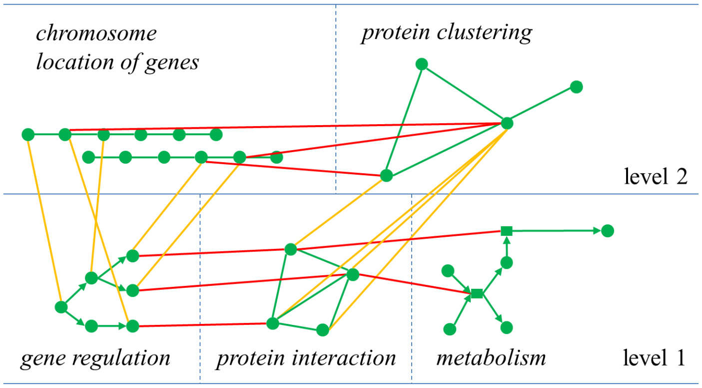
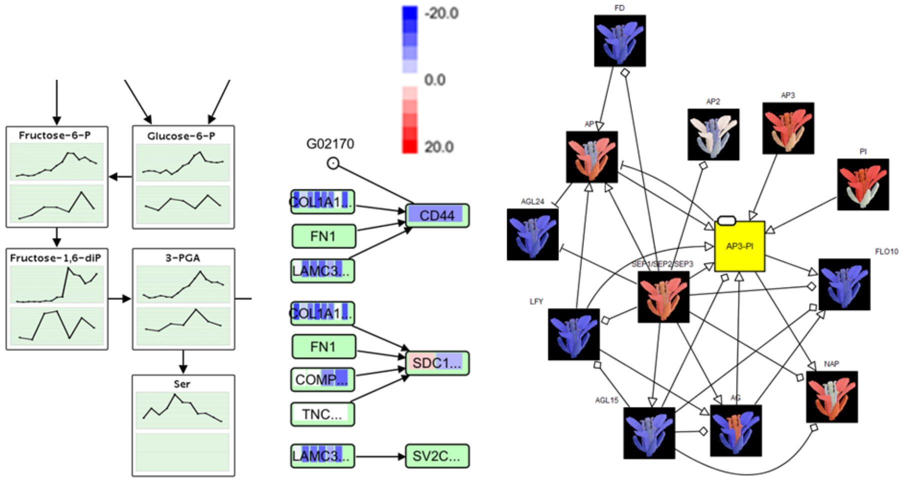


Fig. 9.4. An example instantiation of the data structure described in Sect. 9.1.

Typically multivariate data is connected to nodes and/or edges of the networks. Figure 9.5 gives some examples of such data. Here we will focus on the network structure, but it should be kept in mind that not only the visualization of the networks, but also of the additional multivariate data is often a challenge.

Use cases

Here we discuss use cases that involve the multi-level heterogeneous network shown in Fig. 9.4. Genes encode proteins, and proteins have many functions including catalyzing metabolic reactions in the form of enzymes. This major flow of information motivates the use of the three networks on level 1 in Fig. 9.4: gene regulatory, protein interaction, and metabolic network. These networks focus on different aspects of the same underlying biological system. In addition, level 2 gives further information which is either experimentally



(a) (b) (c)

Fig. 9.5. Examples of multivariate data in biological networks, (a) time-series of relative metabolite levels in a metabolic network (for two different conditions: day and night; from [7]); (b) up- (red) and down-regulated (blue) genes in a signalling pathway (from [39]), (c) spatial resolution of gene expression in the gene regulatory networks of Arabidopsis (from [29]).

obtained (such as the location of genes on the chromosomes) or computed (such as the clustering of proteins based on connectivity information).[4] Typical tasks involving these networks are:

Structural properties of the networks

Which elements of the networks are important nodes (such as regulatory genes), functional building blocks (such as feed-forward motifs [52]), or relevant paths through the network (such as the shortest path between two metabolites)? Such questions are usually answered using methods from network analysis: centrality analysis to evaluate the importance of nodes or edges in the network [44], network clustering which may structure the network into functional modules [5], shortest paths between nodes representing potentially preferred routes, and so on. The results of such analysis have to be visualized in the network context and may give new insights into how important a specific gene is for an organism or which metabolic pathway may be preferred by an organism.

---

[4] Note spatial location and aggregation or grouping of elements can also occur in other application domains; for example, see Fig. 9.6.

Networks and spatial information

Are genes with close proximity on the chromosome involved in the same biological processes? For example, if genes in close proximity on the chromosome are regulated by the same regulators in the gene regulatory network or if they fall in the same cluster in the protein clustering, then they are likely to belong to the same biological process.

Networks and multivariate data

Is experimental data in agreement with knowledge represented by the networks or not? Lots of experimental data can be attached to nodes and edges: data obtained under different experimental conditions such as treatments or temperature, time-series measurements, and so on. For example, the following attributes can be used: edge weights (e. g., to represent fluxes), node weights (e. g., to show concentrations), node presence/absence (e. g., to model knockouts), and node shape (e. g., to show different types of biological objects). Multivariate data has first to be integrated into the biological network. Data mapping deals with this integration of additional data into networks. An example is the integration of metabolomics, transcriptomics, and fluxomics measurement data into the metabolic network. This data can be mapped on different network elements (such as metabolites, enzymes, and reaction edges). The mapped data should behave in a way which could be explained by the underlying networks. If, for example, in the gene regulatory network, only some of the dependent genes get activated if a regulatory gene is active, this could be an indication for a yet undiscovered regulatory process.

Challenges

The compilation of heterogeneous networks requires the identification of the biological entities such as genes, proteins, metabolites, species and so on, and the interconnection between the networks with (different types of) edges. The biological entities are only partly known; therefore, the networks are not complete but change with increasing knowledge. Also, the interconnection between the heterogeneous networks is often difficult to obtain: identifiers for biological entities are often only unique in the context of one data source—for exmple, a database or an ontology. There are initiatives aiming towards the creation of globally unique and persistent URIs (e. g., the MIRIAM registry [47] and identifiers.org [33]), but they are still only used in some databases. A commonly used approach is identifier mapping (see [50] for an overview of tools capable of mapping biological database identifiers).

Tools for the visualization of analysis results often provide standard interaction techniques such as zooming (sometimes also semantic zooming), filtering, collapsing and expanding of structures, and highlighting. There are many tools available, but choosing the right tool is often difficult (for links to some more widely used tools, see the introduction of this section). Standard visualization tasks are often sufficiently solved, but more elaborated visualization, interaction, analytics, and layout methods are an open issue. An overview concerning open problems in biological network visualization can be found in [3].

9.2.2 Social Science

This section reviews heterogeneous, multi-level networks in the social sciences, drawing on research in sociology, information science, statistics, graph theory, and network science [9]. A special focus will be scholarly networks such as citation networks of papers or journals, and collaboration networks of authors, their institutions, and countries. We first present a domain overview introducing relevant terminology along the way. The next subsection discusses major data sources and formats. Then we illuminate diverse network types and interlinkages together with examples. In the next subsection we present concrete use cases that benefit from the formalization presented in Sect. 9.1 and the visualization solutions discussed in Sect. 9.3. We conclude with a discussion of challenges for multi-level network analysis and visualization in the social sciences.

Overview

Social networks exist from the individual (micro) to population (macro) levels. Examples are networks of friendship and hate, collaboration and competition, and trade and blockage between entity nodes such as individuals, organizations, cities, countries, geospatial regions, or areas of science. Typically, an entity node is part of multiple types of networks—for example, a person is part of friendship, collaboration, and family relationship networks. Nodes might contribute to information diffusion networks by receiving and sending emails, tweets, or commenting on digital objects. Nodes from different levels interact and influence each other—for example, individual authors of papers might be aggregated at the institution or even country level, see Fig. 9.6. Journal publications can be aggregated into journals. Undirected collaboration and directed citation edges are aggregated as well. Network edges can be mapped geospatially, supporting geospatial grouping by region, county, country, or continent (see lowest level in Fig. 9.6).

The movement of a highly cited author from one department or country to another will impact both the expertise profiles (entity properties) and collaboration patterns (entity linkages) of associated entity nodes. In other cases, networks from different levels are nested—for example, a person is part of local social networks that can be further aggregated to global, population level networks. Last but not least, the structure of multi-level networks impacts the utility of the network for, for example, information diffusion. The stronger the

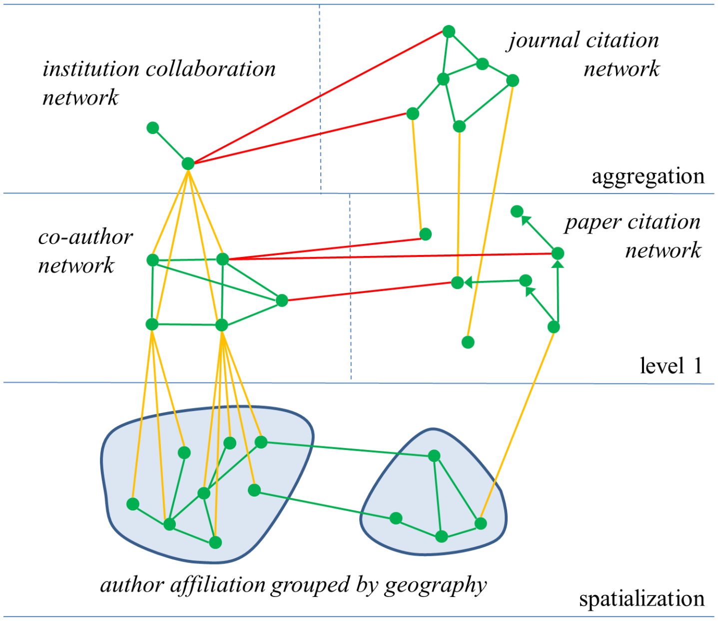


Fig. 9.6. Three-level network of different collaboration and citation networks.

edge between two nodes is, the more information can flow; the more often two nodes share information, the stronger their edge grows over time.

Understanding the evolution of network structure and dynamics has far-ranging applications. Among them are the prevention of disease spreading, for example, identifying highly connected individuals that should be vaccinated first when trying to fight a pandemic or using social networks to reducing the social diffusion of smoking or bad eating behaviors (see subsection "Use cases"); to increase the spreading of information—for example, in education or when designing effective (viral) marketing strategies; to help manage the extremely complex decision making space of professional career choices (see subsection "Use cases"); or to form and sustain productive research and development teams.

Data sources

Social network data might be qualitative or quantitative. Qualitative data is commonly acquired via surveys, interviews, direct observation, or by reviewing written documents. Quantitative data might be derived from existing databases (e. g., phone-address-data-revealing social networks; publication data for extracting co-author networks) or acquired via measurement. Common data sources are social media data (phone, email, blogs, Twitter, Facebook), scholarly data (scientific publications, patents, funding awards), or custom data collected in research studies. Commonly studied networks are social networks, friendship networks, collaboration networks, political networks, trading networks, and citation and other knowledge diffusion networks.

Nodes and their linkages can be represented as an adjacency matrix, an edge list, or lists of nodes and edges. Consequently, most network science tools[5] support Pajek .net or edge list formats as well as Graph/ML, XGMML formats. Many tools support the extraction of networks from common file formats—for example, publication data formats from the Web of Science, Scopus or personal bibliography tools such as EndNote or Latex's bibtex, or from generic tabular formats (e. g., .csv files).

Network types and examples

This subsection discusses different network types using examples from scholarly network analysis—the study of authors and their papers as we assume all readers are familiar with this data.

Networks might be directed or undirected, weighted or unweighted, valued or not. Linkages might be among units of the same type, such as friendship or co-authorship linkages, or between units of different types, such as authors and the papers they produce. In general, three types of linkages are distinguished: *direct linkages* such as paper citation linkages; *co-occurrence linkages* of words or references; and *co-citation linkages* (e. g., of authors or papers). Plus, units of the same type can be interlinked via different link types: for instance, teenagers might be linked via love and hate relationships; papers can be linked according to co-word, co-citation, or bibliographic coupling analysis. Linkages might be directed and/or weighted. Each non-symmetrical occurrence matrix has two associated (symmetrical) co-occurrence matrices; for instance, for each paper citation matrix exists a bibliographic coupling and a co-citation matrix.

An example of direct linkages are paper-paper (citation) linkages: Papers cite other papers via references, forming a non-weighted, directed paper citation graph. It is beneficial to indicate the direction of information flow from older to younger papers via arrows. References enable readers to search the citation graph backward in time. Citations to a paper support the forward traversal of the graph. Citing and being cited can be seen as two vital roles of a paper.

Co-occurrence linkages interconnect co-author networks. Having the names of two authors (or their institutions and countries) jointly listed on one paper, patent, or grant is an empirical manifestation of scholarly collaboration. The more often two authors collaborate, the greater the weight of their joint coauthor link. Weighted, undirected co-authorship networks appear to have a high correlation with social networks that are themselves shaped by geographic proximity.

---

[5] `http://sci2.wiki.cns.iu.edu/display/SCI2TUTORIAL/8.2+Network+Analysis+and+Other+Tools`

Word co-occurrence linkages are used to calculate the topic similarity of basic and aggregate units of science. Units that share more words are assumed to have higher topic overlap and are connected via linkages and/or placed in closer proximity on a topic map. Co-occurrence networks are weighted and undirected.

Co-citation linkages is as follows: Two basic or aggregate units of science are said to be co-cited if papers associated with them jointly appear in the list of references of a third paper. The more often two units are co-cited, the more they are expected to have something in common. Examples are document co-citation and author co-citation networks.

Given a data file with publication records retrieved from the Web of Science or Scopus database, more than 30 different networks can be extracted. Some of these networks—for example, co-author or paper-citation networks—have been studied extensively and their structure, distribution, and evolution are known. Other network types—particularly heterogeneous networks interlinking different levels—have not yet been studied in detail.

Use cases

This subsection exemplarily discusses two use cases that involve multi-level heterogeneous social networks. For each, we identify user needs/tasks, workflows, and insights gained.

Reducing social contagion of smoking behavior

Social peer pressure is powerful. The desire of an individual to be an accepted member of a group frequently leads to behavior that the individual would not show without the group. For example, it is well known that—all other things being equal—joining a well-organized team of experts will lead to higher professional performance than joining a team of less skilled, less organized individuals. Studies by Christakis and Fowler have evaluated a densely interconnected social network of 12,067 people assessed repeatedly from 1971 to 2003 as part of the Framingham Heart Study to show the relevance of social networks in the diffusion of not only smoking and obesity [10], but also loneliness. Relevant news stories were entitled: "Are Your Friends Making You Fat?"[6] While it is difficult and in some cases impossible to distinguish correlation vs. causation [11]—for example, some of the shown effects could be due to the "birds of a feather flock together" factor also known as homophily—networks seem to impact outcomes. A scientific approach that considers homophily, environment, and induction (e. g., by using multi-level heterogeneous networks) seems promising.

For example, it appears beneficial to study individuals in the context of multiple, interconnected networks. Level 1 might comprise the different social

---

[6] `http://www.nytimes.com/2009/09/13/magazine/13contagion-t.html?pagewanted=all&_r=0`

networks of a teenager node A. Among them are family, school, and out-of-school friendship networks. Level 2 aggregates these networks to families and clans (groups of teenagers) that have diverse profiles and interlinkages. Geography matters: the ease by which A can come into contact with teenagers and other individuals that have different demographic profiles is important—just like in Fig. 9.6 a spatialization level can be used to represent geospatial factors. Using this representation, the number of linkages to positive and negative influences can be calculated—for example, the strength of network ties to peers and family members, the rate of contact, and geospatial proximity can be determined for any individual node. Using computational models, it might be possible to predict the general impact of network changes on individual behavior. The big, open question is: How to change social networks and/or environments to cause positive change?

Designing successful career trajectories

Pathways that individuals choose from taking their very first job and then moving on to the next during their working years are called career trajectories. They might be plotted over time, geospatial space, or topic space. Individuals might change locations and jobs because of warfare, political problems, ethnic purging, or because of voluntary or volitional migration where individuals choose to relocate to new places because of opportunities offered in the new place. In the latter case, social factors (e. g., closeness to family and friends, standard of living), colored by cultural, historical, linguistic, or weather considerations, but also active encouragement by visa and immigration controls are key criteria. Education is another major factor—low education typically equates low paying jobs and little resources to pay for education or to move to a different place. High education commonly leads to better paying jobs and the generation of financial wealth that pays for personal education or the education of family members. Highly specialized expertise profiles might also mean that only a few jobs exist that match this expertise profile and international migration is required to find an appropriate job.

Coming back to the study of scholarly networks, much data exists to track career trajectories of scholars over time. The U.S. National Science Foundation has been conducting the Survey of Doctorate Recipients (SDR) biennially since 1973. The SDR follows a sample of U.S.-trained doctorates in science, engineering, and health fields throughout their careers from shortly after degree award by a U.S. institution through age 75. Multivariate data such as detailed information on professional position, salary, number of kids, etc. is available for each respondent in a longitudinal fashion. In addition, there exist extensive publication and funding databases that record funding intake (number of awards and funding amounts) and publication output (number of papers and their citations) over time together with information on not only co-investigators and coauthors, but also institutional affiliations, and acknowledged grant funding that interlinks funding and publications.

In general, career trajectories are best viewed as decisions over time. As it is much easier to change institutions than to change a topical area of expertise; researchers change geolocations frequently (particularly in the beginning of their career). However, they are less likely to change their topic area and, if they do, decide to venture into topically similar areas of science that benefit from the same skill set/expertise. Multi-level networks can be used to represent the impact of geospatial features—for example, car routes and air traffic networks as a proxy for reachability (spatialization level). Level 1 might show scholar node B and associated family links, mostly local friendship networks, (international) co-author relations, and collegial networks at the same institution. Level 2 might be used to represent the reputation and interlinkages of institutions and scientific disciplines. Family and friendship links, the reputation of another institution, and also attributes of the new geolocation (living costs, weather, etc.) have the power to make a person look for or accept another job—if a job offer is made. The new institutional environment together with new friendship and collaboration opportunities will change funding and publication patterns of an individual scholar and ultimately the conditions for the next career step. Note that not only individuals and families but also companies and organizations migrate in response to market changes, to maximize economic utility, or to be close to customers.

Challenges

The extraction of heterogeneous networks requires the identification of unique entities—for example, people, institutions, scientific areas, and their interlinkage via (different types of) edges. While publications, patents, and other digital documents have unique digital object identifiers, the development of digital object identifiers for authors, institutions, and scientific areas is still under active development. Similarly, the interlinkage of (heterogeneous) networks across levels poses serious unification and data mapping challenges. Many problems require $n : m$ mappings not only within one level but also between consecutive levels—implications for restricting the mapping to $1 : m$ are unknown. For example, Fig. 9.6 assumes that each author has exactly one affiliation and each institution is mapped to one country—this is not true for all authors nor institutions.

Few algorithms exist to analyze and visualize heterogeneous networks—most algorithms assume there is one node type and one edge type. Visualizations are important to communicate complex heterogeneous, multi-level networks. However, most readers hardly ever learned how to read and interpret a network and seeing multiple interlinked networks with different node and edge types is often overwhelming.

9.2.3 Software Engineering

A plethora of tasks in software engineering involves analyzing software artifacts that are best represented using networks, see also Chap. 2. We will focus on object-oriented programming languages like Java or C++ for implementation. Other languages will lead to similar structures. The different networks can be categorized into static software structure (e. g., classes in Java or C++ and their connections) and dynamic software structures (e. g., which classes are instantiated and when and how these instances are connected to each other). Furthermore, different structural connections between the same nodes (classes) are possible—for example, call graph, inheritance (both directed) or code clones (undirected). This forms a within-level set of graphs. Another within-level set of graphs is given by the change of the graphs over time. During the evolution of the software system (seen as snapshots in repositories like svn, cvs, or git), the graph changes: nodes (classes) are added, deleted, or changed, and also links are added or deleted, or they change their properties (e. g., the number of methods of one class called by another one). Finally, software entities are structured: methods belong to classes belong to modules. These modules can be represented by, e. g., packages in Java or namespaces in C++. This package or namespace tree can be directly mapped to the levels of Fig. 9.2.

In the following of this section we first present a domain overview introducing relevant terminology along the way. The next subsection discusses major data sources and formats. Then we present diverse network types and interlinkages together with examples. The following subsection presents concrete use cases. We conclude with a discussion of challenges for multi-level network analysis and visualization in software engineering.

Overview

The data produced and used in software engineering is manifold. First of all, software development processes can be mapped to directed graphs with loops, where each development step is mapped onto a node, and an edge designates the sequence of these steps. Further, software development artifacts like system designs, detailed designs, and the static and dynamic structure of programs can be mapped onto directed and undirected graphs. Finally, algorithms can be visualized. In this latter case, graphs are more rarely used, except that the algorithm requires the handling of graphs.

The networks can be time-dependent ($n : m$ mapping within a level)—for example, data from cvs, svn, or other repositories—and multi-level ($1 : n$ mapping between levels)—for example, package structure in Java. Further, they have different node and edge types, and to each of these nodes and edges additional information (multivariate data) can be associated. Thus, the networks have all properties described in Sect. 9.1 and shown in Fig. 9.2.

An exemplary instantiation of Fig. 9.2 is shown in Fig. 9.7. All nodes in the lowest-level—level 1—represent classes. These classes are related by edges (green) representing, for example, method calls. They are grouped according to their revision (release); three releases are shown. Classes (nodes) that are present in subsequent revisions are connected by red edges. Aggregating

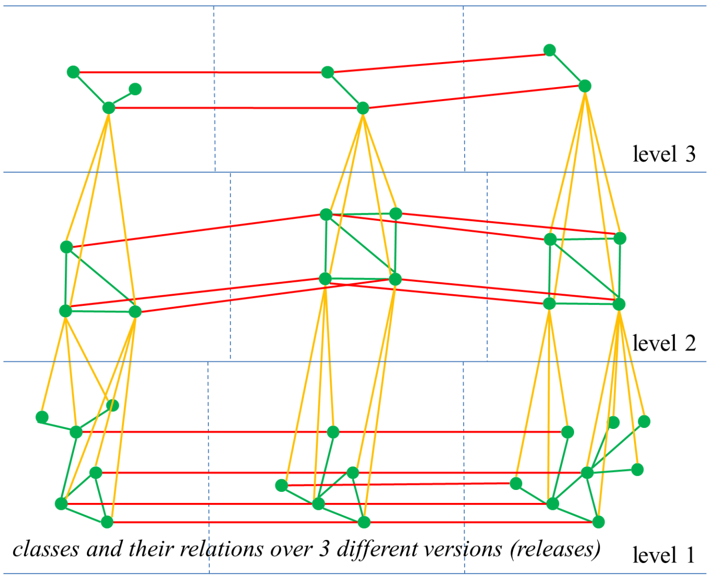


Fig. 9.7. An example instantiation of the data structure described in Section 9.1.

classes into packages (yellow edges) results in the graphs on level 2. Here, the nodes represent both nodes and packages. Packages are those nodes which are connected by yellow edges to level 1. The same holds for level 3.

Understanding software processes is important for assigning and managing resources during the software development process, while understanding the structure and the behavior of software products (software comprehension) is important for bug fixing and extending existing software systems (software maintenance).

Data sources

Networks in software engineering are either created manually or automatically. Manual creation occurs normally for describing software engineering processes and during software development in the design phase. However, the latter networks are only reliable in the case of automatic code generation. Whenever code is generated manually, the design is usually changed either intentionally or unintentionally. Therefore, software systems are commonly analyzed based on graphs extracted directly from the source code. The source code itself comes from software repositories like svn, cvs, or git. During the extraction, the graphs are generated. Some systems additionally extract so-called software metrics. In the latter case, these software metrics are used as attributes of the classes (nodes). The attributes for the edges can also be inferred from these metrics. On the other hand, the number of edges of the same type can be counted and used as an additional measurement (e. g., how many different method calls occur from class A to class B).

The granularity of the analysis influences which are the lowest-level entities (nodes). Most often, either classes, methods or member variables of classes are chosen as atomic entities. Methods and member variables of classes are a part of classes, which in turn are a part of packages. Packages themselves form a tree with one root package or a forest with several root packages. Edges might represent directed relations like method calls, inheritance, instantiation, or aggregation, or undirected relations like code clones or common fate.

Finally, execution logs (traces) of program runs are mined to create dynamic call graphs. In this case, nodes are mostly the instantiated classes, while edges are mostly the dynamic call relations between the class instances. However, static relations like aggregation or inheritance can help to understand the behavior of the programs. More data sources and how these sources are mined to obtain information are described in Sect. 2.3.3 of this book.

Network types and examples

Software processes are mapped onto networks similarly to other processes (e. g., business processes, schedules, or production processes). Each development step is mapped onto a node and can have additional information assigned (e. g., number of developers, time allocated for this step, a list of tasks, inputs, and deliverables). Each step is connected to the following step by an edge. This edge normally is guarded in the sense that it can be only followed if the deliverables of the previous step are ready (to a certain amount). As the processes can branch and can have loops, the graphs cannot be reduced to trees or DAGs (directed acyclic graphs). Mostly, the process comes in an abstract form that is instantiated for a specific project. Therefore, several similar networks exist that can be compared to each other (e. g., the instantiations among each other). Further, the process can be in the form of a hierarchy. Then, each node of the high-level process is refined into a number of steps that can be refined themselves.

The static structure of software artifacts can also be mapped to typically directed graphs. Taking Java source code, these networks have the following properties. The hierarchy of Java packages can be mapped to different levels. The lowest-level nodes can be either classes or methods, members, and subclasses of classes. The snapshots in time (typically saved in revision-control systems) form a sequence of hierarchical networks. Many revision-control systems allow branches and merges, such that the sequence becomes a directed acyclic graph. Possible links (edges) on class level are inheritance, method call, aggregation, implements relation, and usage relation (similar to aggregation, where the class is used as type of another class' member, it can also be used as type of return values, parameters, or exceptions). Besides these directed links (edges), also undirected links (edges) might also exist like code clones, common fate (evolution, classes that are often changed together), or semantic similarity. Aggregating the classes and sub-packages in packages directly implies the aggregation of the edges; thus, a hierarchy is built for one time-step.

The dynamic structure of a software system also leads to a sequence of networks. However, these networks typically have a $1 : n$ mapping as the underlying static structure does not change. Instead, individual edges are activated and deactivated according to the dynamic evolution of the networks. Indeed, the nodes of the dynamic networks are instances of the nodes of the static network, where each static node can lead to an arbitrary number of dynamic nodes. The same holds for the edges: call graph edges get instantiated during the evolution of the run. Of course, the granularity can be again on class or on any of the package levels.

Use cases

Use cases are typically directly derived from software engineering tasks (see also Sect. 2.3 in this book). Tasks that benefit from visualizations include

Software understanding

Understanding the functionality and the interplay of components necessitates the understanding of the static and dynamic structure. Further, it is the basis for the maintenance task.

Maintenance

To maintain a system, a thorough understanding of the individual components as well as their dependencies is needed. The dependencies are best analyzed based on visualizations of the static structure and dynamic runs.

Re-engineering

Re-engineering is needed for legacy systems or whenever the new developer can not directly access the knowledge of the original developers. Like Software understanding, the interplay of the different components plays an important role, which is best understood using static and dynamic networks.

Testing and bug fixing

For bug fixing, the interplay of components is very important, because side effects on other components should be minimized when changing an individual component. The same holds for adding new functionality. The current behavior should not change by adding additional functionality.

Product lines

Product lines incorporate basic functionality in a core of components, adding functionality by additional or changed components. For product lines, it is mandatory to understand the relationship between the different products. The goal is to maximize the amount of common components, minimizing the effort for creating additional functionality using additional components.

Challenges

There exist a lot of software visualizations, but only a few of them are scalable and comprehensive. Most of the existing solutions either focus on specific properties of software artifacts, like metrics, static structure (UML diagrams) at one time step with several types of edges, or static structure over time with one edge type.

Highly necessary for effective program comprehension are integrated views like the AreaView tool developed by Byelas at al. [8]. While this tool integrates UML diagrams with metrics and several areas of interests, it displays neither hierarchies nor several time steps. We need an integrated set of visualizations and interactions that allows us to look at static software artifacts from several points of view, showing or filtering information on demand to allow a focused analysis of the software artifacts. Such approaches should visualize networks with the following properties:

1. time-dependent static structures: e. g., svn or cvs snapshots;
2. structure hierarchies: e. g., packages and classes in Java;
3. distinct node types: e. g., classes and interfaces in Java;
4. distinct edge types: e. g., inheritance, aggregation, method calls (directed) and code clones, semantic similarity (undirected);
5. software metrics as additional node information: e. g., lines of code, number of methods, depth of inheritance;
6. metrics associated with edges: e. g., number of method call relations, similarity.

While all of these properties mainly map to visual structures, interaction has always to be considered as part of the solution—that is, at least the use of standard interaction techniques is mandatory. Only if users are allowed to interactively explore selected parts of the software artifacts, they will be able to gain new insights and find the information needed for solving specific tasks. Thus, interactive visualizations must be seamlessly embedded in the software analyst's work flow. Only then can analysts discover complex patterns in software. Specific challenges derived from this general challenge are:

Scalability: How to depict several levels of several revisions over time? Usually, the graphs are so large that a cut through the level hierarchy is needed showing the focused information in detail (lowest-level) and the context information reduced (on a higher level). Showing a series of cuts is then asked for to analyze the evolution of the software system for finding patterns or unusual changes, see also Chap. 10.

Comprehension: What is the best visualization-interaction combination for showing all relevant information?

- *Metrics:* On the one hand, metrics (i. e., attributes on the nodes and edges) have to be included in a non-obstructive way with respect to the structure.
- *Multiple Edge Types:* On the other hand—besides the hierarchy—each level might also represent different edge types. Then the yellow edges in Fig. 9.7 represent again the same node, while the levels in this case represent different structural information like call graphs, inheritance, or code clones. Possible solutions for this problem have already been proposed by Abuthawabeh et al. [1, 2] based on a matrix visualization. A similar approach using node-link diagrams has been proposed by Knodel et al. [40]. However, these two approaches need further improvements to become valuable in software comprehension tasks.

9.3 Visualization

The visualization of heterogeneous networks on multiple levels is still relatively unexplored in the literature. However, there are a number of visualization approaches that focus on solving specific analysis tasks or operate on a subset of the data structure introduced in Sect. 9.1, for instance on heterogeneous multivariate networks: some approaches abstain from explicit encoding of the network topology and visualize aggregated information only. This idea supports the analysis of very large data sets in terms of many heterogeneous networks and large multivariate attributes. ManyNets [19] represents networks as rows in a table together with their multivariate data (primary as well as secondary data [37]) similar to the well-known TableLens [58]. Several interaction possibilities support the visual analysis of the networks which might also be displayed as node-link diagram on demand. GraphTrail [16] has similar aims, but in contrast to showing the networks in a table, the developers have chosen to represent the network elements in an aggregated form. For doing this, standard charts like bar charts or tag clouds are employed that can be interactively arranged on a canvas. GraphTrail also supports the analysis process by providing a history functionality (Sect. 6.3.2 in Chap. 6 discusses GraphTrail in more detail). Other approaches abstract directly in the node-link representations, such as OntoVis [63]: an ontology graph, which describes the node categories/clusters and their relationships and serves as a vehicle to control the abstraction and navigation processes. In addition, layout methods have been proposed which try to preserve similar parts in the heterogenous networks such as the visualization of two or three heterogenous networks in parallel planes in three dimensions is discussed in [21]. All these tools and approaches have in common that they provide solutions for analyzing a set of heterogeneous multivariate networks, but *not* at multiple levels.

In the following, we provide a short overview of techniques and ideas that might partly solve the problem of representing a set of heterogeneous networks distributed in several levels. All figures refer back to the sample three-level networks in Fig. 9.2, with green indicating the lowest level, red indicating the middle level, and blue indicating the upper level.

9.3.1 Approaches for Networks at Multiple Levels

Stacking

The most obvious visualization metaphor for networks on different levels is stacking. All networks on the same level are laid out (by using any more or less smart graph drawing algorithm) on a 2D plane, and then these planes are stacked in 3D (cf. Fig. 9.8). Multivariate data attached to the nodes or edges might be displayed within the planes themselves, as additional layers below or above the individual 2D planes, or separated into multiple coordinated views [59]. One of the existing example tools is VisLink [12], which is a general approach to show relationships between visualizations. In our special case, networks on levels are displayed on multiple 2D planes that can be arranged in the third dimension in various ways (in parallel, book-like, etc.). Relationships are represented as links—that is, as inter-plane edges. Here also, multivariate data can be represented inside of the planes or on additional individual planes. In the latter case, inter-plane edges might be used to point to the attached multivariate data. Note that—according to our definitions—heterogeneous networks are usually placed by VisLink to different planes and not on one plane. The most obvious drawback of such 3D techniques is their low level of scalability as well as clutter and perspective distortions especially when showing multivariate data in combination with the networks themselves.

Nesting

Another thinkable visualization approach is to use nesting for the explicit encoding of inter-level edges. This requires that mappings across consecutive levels are of $1 : n$ type. Figure 9.9 shows an example of how such an approach might look. Multivariate data could be represented as additional graphical features of the nested boxes/circles or within separated, coordinated views. Advantages are the “flat” layout which might support finding answers for specific tasks, such as the analysis of the aggregation results between co-author networks and institute collaboration networks (cf. Fig. 9.6). Another benefit is the integration of various interaction techniques similar to Treemaps [28]. Disadvantages are the visual complexity of the approach—induced by the mixture of link and box elements—as well as the possibly high space consumption.

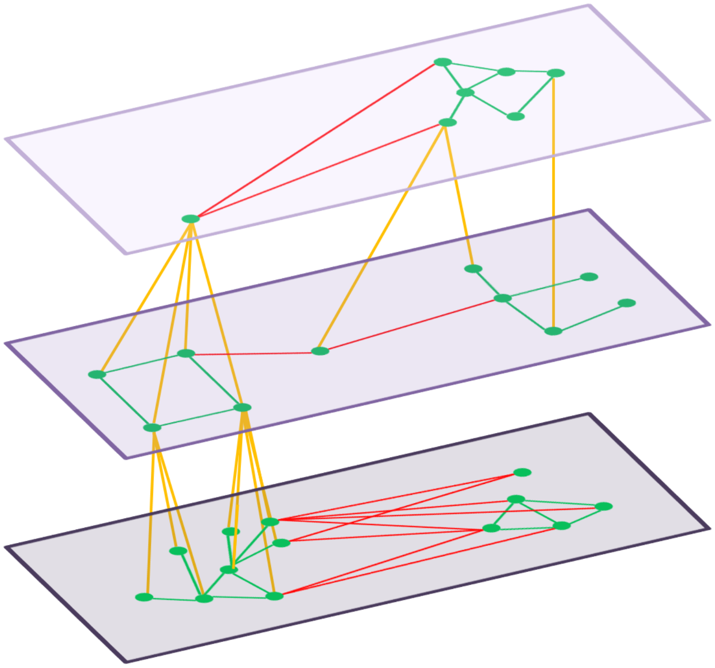


Fig. 9.8. Stacked 2D planes of network drawings which show the same networks as given in Fig. 9.2.

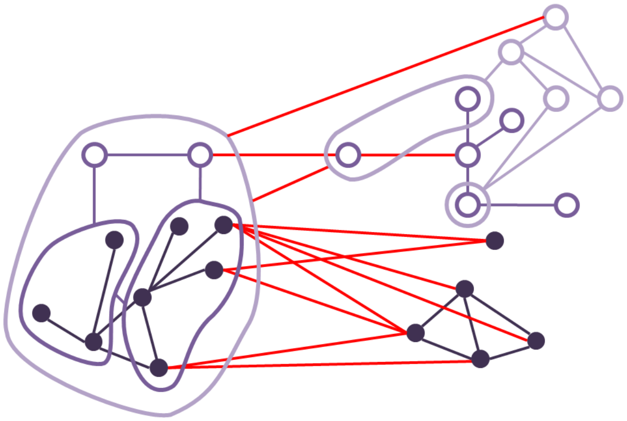


Fig. 9.9. Sketch of a nesting approach which shows the same networks as given in Fig. 9.2. Closed contours (incl. circles for nodes in level i that have no links to level $i-1$) are used to represent the $1:n$ mapping between levels.

Although this approach can be used for the hierarchical presentation of one single network (e. g., clustering) [15], we are not aware of related works in the visualization literature for heterogeneous networks at multiple levels.

Alignment

The next approach uses one view per level which are all aligned to each other (cf. Fig 9.10). If a user brings a network part at one specific level into his view by zooming and panning, the related nodes in the other networks are shown simultaneously within the remaining coordinated views. Brushing can be used to selected individual nodes together with their neighbors at other levels as exemplified in the figure. Advantages of this idea are the simple metaphor which can be implemented easily as well as the rich interaction possibilities. Negative aspects might be the large space consumption of the many views and the missing inter-level edges. However, so-called context-preserving visual links [68] could solve this issue. Multivariate data could be displayed within the views (e. g., by glyphs or similar) or within additional coordinated views. The Entourage tool [49] realizes a similar approach with a special focus on contextual subsets, but without explicit encoding of the different levels. Here, contextually relevant pathways are displayed side-by-

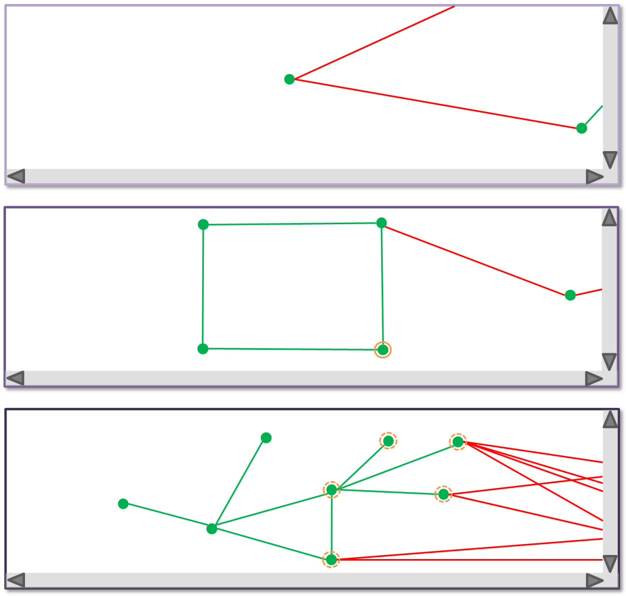


Fig. 9.10. Sketch of the alignment approach which shows the same networks as given in Fig. 9.2. Here, one node in the middle level was selected (orange halo); its neighbors in the lower level were highlighted accordingly (dashed orange halo).

side together with a focus pathway, and only important parts of those context pathways are visible depending on the current selection in the focus pathway.

9.3.2 Challenges and Future Directions

Heterogeneous networks on multiple levels are not easy to visualize even without attached multivariate data. One reason is the sheer size of the involved networks. Most graph drawing methods do not scale well. Another reason is the specific structure that is inherently given by the set of heterogeneous networks and the levels themselves: it is not efficiently reflected by most visualization approaches. Clever interaction techniques might help here, but in contrast to the visual analysis of single networks (or perhaps also heterogeneous networks within one level) we do not have a good understanding which interaction techniques and analytical methods work best in this context. More work has to be done to develop new visual representations and interaction metaphors to solve the specific problems and tasks described in the previous sections. This process has to be accompanied by user studies. Performing good and reliable evaluations is a challenge on its own, and we refer to the book [57] for further reading.

In this chapter, we restricted ourselves to $1 : n$ mappings across consecutive levels which can be sufficiently motivated by many concrete data sets and analysis tasks in our applications fields. However, there are—of course—situations in practice that demand universal $n : m$ relationships between network elements in different levels that must not be consecutive. Section 9.2.2 briefly exemplifies this issue. Because of the structural flexibility which comes with such general approaches, visualization experts have difficulties with the development of novel methods and tools that are able to handle those cases. From the perspective of visual analytics, more research has to be done in order to improve/facilitate the analysis processes. The analysis of heterogeneous networks on multiple levels is usually not done by just one analyst. Usually, several people work together—whether it be at one place (co-located) or several places (distributed), or whether it be at one specific period in time or at several different times. Our visualization and analysis tools should be able to support such collaborative work, record analysis sessions, support annotations by the users, and provide some guidance during the analysis process [37]—that is, a visualization should support "guided analytics to lead analysts through workflows for common tasks" [23].

For the integration of multivariate data into heterogeneous networks on multiple levels initial ideas have been proposed, but so far we have not really solved this problem. Both the network topologies and the attached multivariate data together are of great importance to the analysts, and in many tasks both is needed to solve specific questions and gain insights into the data. All ideas presented in Sect. 9.3.1 have the tendency to pay more regard to the network topology and not so much on the multivariate data. Such data can be added via glyphs or coordinated views, but this is not enough to discover patterns between the data and the network structure. Vice versa, if we focus on the multivariate data such as done by attribute-based network layouts [64, 71] and similar approaches [32], we run into the same problems. Thus, finding an appropriate bunch of techniques for the common analysis of multivariate data within networks of networks is still an unsolved challenge.

Acknowledgments

We would like to thank all participants of the Dagstuhl Seminar ♯13201 [36] for the fruitful discussions and Todd Theriault for carefully proof-reading our chapter. This research is supported in part by the U.S. National Institutes of Health under Grant No. U01 GM098959, the German Ministry of Education and Research under Grant 0101-31P7126, and the German Academic Exchange Service (DAAD) under Grand 54391720.

References

1. Abuthawabeh, A., Beck, F., Zeckzer, D., Diehl, S.: Finding Structures in Multi-Type Code Couplings with Node-Link and Matrix Visualizations. In: Proceedings of the first IEEE Working Conference on Software Visualization (VISSOFT '13) (2013)
2. Abuthawabeh, A., Zeckzer, D.: IMMV: An Interactive Multi-Matrix Visualization for Program Comprehension. In: Proceedings of the first IEEE Working Conference on Software Visualization (VISSOFT '13) (2013)
3. Albrecht, M., Kerren, A., Klein, K., Kohlbacher, O., Mutzel, P., Paul, W., Schreiber, F., Wybrow, M.: On open problems in biological network visualization. In: Eppstein, D., Gansner, E.R. (eds.) Graph Drawing, 17th International Symposium, GD 2009. Lecture Notes in Computer Science, vol. 5849, pp. 256–267. Springer (2010)
4. Bader, G.D., Cary, M.P., Sander, C.: Pathguide: a pathway resource list. Nucleic Acids Research 34, D504–D506 (2006)
5. Balasundaram, B., Butenko, S.: Network clustering. In: Junker, B.H., Schreiber, F. (eds.) Analysis of Biological Networks, pp. 113–138. Wiley Series on Bioinformatics, Computational Techniques and Engineering, Wiley (2008)
6. Bezerianos, A., Chevalier, F., Dragicevic, P., Elmqvist, N., Fekete, J.D.: Graphdice: A system for exploring multivariate social networks. Computer Graphics Forum (Proc. EuroVis 2010) 29(3), 863–872 (2010)
7. Borisjuk, L., Hajirezaei, M., Klukas, C., Rolletschek, H., Schreiber, F.: Integrating data from biological experiments into metabolic networks with the DBE information system. In Silico Biology 5, e11 (2004)
8. Byelas, H., Bondarev, E., Telea, A.: Visualization of areas of interest in component-based system architectures. In: Proceedings of the 32nd Euromicro Conference on Software Engineering and Advanced Applications. pp. 160–169. IEEE Computer Society Press (2006)

9. Brner, K., Sanyal, S., Vespignani, A.: Network science. In: Cronin, B. (ed.) Annual Review of Information Science and Technology. pp. 537–607. Information Today, Inc./American Society for Information Science and Technology (2007)
10. Christakis, N.A., , Fowler, J.H.: The spread of obesity in a large social network over 32 years. New England Journal of Medicine 357, 370–379 (2007)
11. Cohen-Cole, E., Fletcher, J.M.: Detecting implausible social network effects in acne, height, and headaches: longitudinal analysis. BMJ p. 337 (2008)
12. Collins, C., Carpendale, S.: Vislink: Revealing relationships amongst visualizations. IEEE Transactions on Visualization and Computer Graphics 13(6), 1192–1199 (2007)
13. Czauderna, T., Klukas, C., Schreiber, F.: Editing, validating and translating of SBGN maps. Bioinformatics 26(18), 2340–2341 (2010)
14. Demir, E., Cary, M.P., Paley, S., Fukuda, K., Lemer, C., Vastrik, I., Wu, G., D'Eustachio, P., Schaefer, C., Luciano, J., Schacherer, F., Martinez-Flores, I., Hu, Z., Jimenez-Jacinto, V., Joshi-Tope, G., Kandasamy, K., Lopez-Fuentes, A.C., Mi, H., Pichler, E., Rodchenkov, I., Splendiani, A., Tkachev, S., Zucker, J., Gopinath, G., Rajasimha, H., Ramakrishnan, R., Shah, I., Syed, M., Anwar, N., Babur, O., Blinov, M., Brauner, E., Corwin, D., Donaldson, S., Gibbons, F., Goldberg, R., Hornbeck, P., Luna, A., Murray-Rust, P., Neumann, E., Ruebenacker, O., Reubenacker, O., Samwald, M., van Iersel, M., Wimalaratne, S., Allen, K., Braun, B., Whirl-Carrillo, M., Cheung, K.H., Dahlquist, K., Finney, A., Gillespie, M., Glass, E., Gong, L., Haw, R., Honig, M., Hubaut, O., Kane, D., Krupa, S., Kutmon, M., Leonard, J., Marks, D., Merberg, D., Petri, V., Pico, A., Ravenscroft, D., Ren, L., Shah, N., Sunshine, M., Tang, R., Whaley, R., Letovksy, S., Buetow, K.H., Rzhetsky, A., Schachter, V., Sobral, B.S., Dogrusoz, U., McWeeney, S., Aladjem, M., Birney, E., Collado-Vides, J., Goto, S., Hucka, M., Novere, N.L., Maltsev, N., Pandey, A., Thomas, P., Wingender, E., Karp, P.D., Sander, C., Bader, G.D.: The BioPAX community standard for pathway data sharing. Nature Biotechnology 28(9), 935–942 (2010)
15. Dogrusoz, U., Giral, E., Cetintas, A., Civril, A., Demir, E.: A compound graph layout algorithm for biological pathways. In: Pach, J. (ed.) Proc. International Symposium on Graph Drawing (GD'04). pp. 442–447. LNCS (2004)
16. Dunne, C., Henry-Riche, N., Lee, B., Metoyer, R., Robertson, G.: Graphtrail: analyzing large multivariate, heterogeneous networks while supporting exploration history. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 1663–1672. CHI '12, ACM, New York, NY, USA (2012), `http://doi.acm.org/10.1145/2207676.2208293`
17. Dwyer, T., Hong, S.H., Koschützki, D., Schreiber, F., Xu, K.: Visual analysis of network centralities. In: Misue, K., Sugiyama, K., Tanaka, J. (eds.) Proc. Asia-Pacific Symposium on Information Visualization (APVis'06). CRPIT, vol. 60, pp. 189–198. ACS (2006)
18. Fernández-Suárez, X.M., Galperin, M.Y.: The 2013 Nucleic Acids Research database issue and the online molecular biology database collection. Nucleic Acids Research 41, D1–D7 (2013)
19. Freire, M., Plaisant, C., Shneiderman, B., Golbeck, J.: Manynets: an interface for multiple network analysis and visualization. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 213–222. CHI '10, ACM, New York, NY, USA (2010), `http://doi.acm.org/10.1145/1753326.1753358`

20. Funahashi, A., Matsuoka, Y., Jouraku, A., Kitano, H., Kikuchi, N.: CellDesigner: a modeling tool for biochemical networks. In: Proceedings of the 38th conference on Winter simulation. pp. 1707–1712. Winter Simulation Conference (2006)
21. Fung, D.C.Y., Hong, S.H., Koschützki, D., Schreiber, F., Xu, K.: Visual analysis of overlapping biological networks. In: Proceedings of the 13th International Conference on Information Visualisation (IV). pp. 337–342. IEEE Computer Society Press (2009)
22. Gehlenborg, N., O'Donoghue, S.I., Baliga, N.S., Goesmann, A., Hibbs, M.A., Kitano, H., Kohlbacher, O., Neuweger, H., Schneider, R., Tenenbaum, D., Gavin, A.C.: Visualization of omics data for systems biology. Nature Methods 7, S56–S68 (2010)
23. Heer, J., Shneiderman, B.: Interactive dynamics for visual analysis. Communication of the ACM 55(4), 45–54 (Apr 2012), `http://doi.acm.org/10.1145/2133806.2133821`
24. Herman, I., Melançon, G., Marshall, M.S.: Graph visualization and navigation in information visualization: A survey. IEEE Transactions on Visualization and Computer Graphics 6(1), 24–43 (2000)
25. Hu, Z., Hung, J.H., Wang, Y., Chang, Y.C., Huang, C.L., Huyck, M., DeLisi, C.: VisANT 3.5: Multi-scale network visualization, analysis and inference based on the gene ontology. Nucleic Acids Research 37(Web Server issue), W115–W121 (2009)
26. Hucka, M., Finney, A., Sauro, H.M., Bolouri, H., Doyle, J.C., Kitano, H., Arkin, A.P., Bornstein, B.J., Bray, D., Cornish-Bowden, A., Cuellar, A.A., Dronov, S., Gilles, E.D., Ginkel, M., Gor, V., Goryanin, I., Hedley, W.J., Hodgman, T.C., Hofmeyr, J.H., Hunter, P.J., Juty, N.S., Kasberger, J.L., Kremling, A., Kummer, U., Novere, N.L., Loew, L.M., Lucio, D., Mendes, P., Minch, E., Mjolsness, E.D., Nakayama, Y., Nelson, M.R., Nielsen, P.F., Sakurada, T., Schaff, J.C., Shapiro, B.E., Shimizu, T.S., Spence, H.D., Stelling, J., Takahashi, K., Tomita, M., Wagner, J., Wang, J.: The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19, 524–531 (2003)
27. van Iersel, M.P., Kelder, T., Pico, A.R., Hanspers, K., Coort, S., Conklin, B.R., Evelo, C.: Presenting and exploring biological pathways with PathVisio. BMC Bioinformatics 9, 399.1–9 (2008)
28. Johnson, B., Shneiderman, B.: Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In: Proceedings of the 2nd Conference on Visualization (Vis '91). pp. 284–291. IEEE Computer Society Press, Los Alamitos, CA, USA (1991), `http://portal.acm.org/citation.cfm?id=949607.949654`
29. Junker, A., Rohn, H., Schreiber, F.: Visual analysis of transcriptome data in the context of anatomical structures and biological networks. Frontiers in Plant Science 3, 252 (2012)
30. Junker, B.H., Klukas, C., Schreiber, F.: VANTED: A system for advanced data analysis and visualization in the context of biological networks. BMC Bioinformatics 7, 109 (2006)
31. Jusufi, I.: Multivariate Networks: Visualization and Interaction Techniques. Ph.D. Thesis, Linnaeus University, Växjö, Sweden (2013)

32. Jusufi, I., Kerren, A., Zimmer, B.: Multivariate network exploration with JauntyNets. In: Proceedings of the 17th International Conference on Information Visualisation (IV '13). pp. 19–27. IEEE Computer Society Press (2013)
33. Juty, N., Le Novère, N., Laibe, C.: Identifiers.org and MIRIAM registry: community resources to provide persistent identification. Nucleic Acids Research 40(1), D580–D586 (2012)
34. Kanehisa, M., Goto, S.: KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Research 28(1), 27–30 (2000)
35. Kerren, A., Köstinger, H., Zimmer, B.: Vincent – visualisation of network centralities. In: Proceedings of the International Conference on Information Visualization Theory and Applications (IVAPP '12). pp. 703–712. INSTICC (2012)
36. Kerren, A., Purchase, H., Ward, M.O.: Information Visualization – Towards Multivariate Network Visualization (Dagstuhl Seminar 13201). Dagstuhl Reports 3(5), 19–42 (2013), `http://drops.dagstuhl.de/opus/volltexte/2013/4177`
37. Kerren, A., Schreiber, F.: Toward the role of interaction in visual analytics. In: Proceedings of the Winter Simulation Conference. pp. 420:1–420:13. WSC '12, Winter Simulation Conference (2012), `http://dl.acm.org/citation.cfm?id=2429759.2430303`
38. Kerren, A., Stasko, J.T., Fekete, J.D., North, C. (eds.): Information Visualization: Human-Centered Issues and Perspectives, Lecture Notes in Computer Science, vol. 4950. Springer, Berlin, Heidelberg (2008)
39. Klukas, C., Schreiber, F.: Integration of -omics data and networks for biomedical research with Vanted. Journal of Integrative Bioinformatics 7(2), 112 (2010)
40. Knodel, J., Muthig, D., Naab, M.: Understanding software architectures by visualization–an experiment with graphical elements. Reverse Engineering, Working Conference on 0, 39–50 (2006)
41. Köhler, J., Baumbach, J., Taubert, J., Specht, M., Skusa, A., Rüegg, A., Rawlings, C., Verrier, P., Philippi, S.: Graph-based analysis and visualization of experimental results with ONDEX. Bioinformatics 22(11), 1383–1390 (2006)
42. Kolpakov, F.A.: BioUML – framework for visual modeling and simulation of biological systems. In: Proceedings of the International Conference on Bioinformatics of Genome Regulation and Structure. pp. 130–133. Springer (2002)
43. Kono, N., Arakawa, K., Ogawa, R., Kido, N., Oshita, K., Ikegami, K., Tamaki, S., Tomit, M.: Pathway Projector: Web-based zoomable pathway browser using KEGG atlas and Google maps API. PLoS ONE 4(11), e7710 (2009)
44. Koschützki, D.: Network centralities. In: Junker, B.H., Schreiber, F. (eds.) Analysis of Biological Networks, pp. 65–84. Wiley Series on Bioinformatics, Computational Techniques and Engineering, Wiley (2008)
45. Küntzer, J., Backes, C., Blum, T., Gerasch, A., Kaufmann, M., Kohlbacher, O., Lenhof, H.P.: Bndb - the biochemical network database. BMC Bioinformatics 8, 367 (2007)
46. von Landesberger, T., Kuijper, A., Schreck, T., Kohlhammer, J., van Wijk, J., Fekete, J.D., Fellner, D.: Visual analysis of large graphs: State-of-the-art and future research challenges. Computer Graphics Forum 30(6), 1719–1749 (2011), `http://dx.doi.org/10.1111/j.1467-8659.2011.01898.x`
47. Le Novère, N., Finney, A., Hucka, M., Bhalla, U.S., Campagne, F., Collado-Vides, J., Crampin, E.J., Halstead, M., Klipp, E., Mendes, P., Nielsen, P., Sauro, H., Shapiro, B., Snoep, J.L., Spence, H.D., Wanner, B.L.: Minimum information

requested in the annotation of biochemical models (MIRIAM). Nature Biotechnology 23(12), 1509–1515 (2005)

48. Le Novère, N., Hucka, M., Mi, H., Moodie, S., Schreiber, F., Sorokin, A., Demir, E., Wegner, K., Aladjem, M.I., Wimalaratne, S.M., Bergman, F.T., Gauges, R., Ghazal, P., Kawaji, H., Li, L., Matsuoka, Y., Villéger, A., Boyd, S.E., Calzone, L., Courtot, M., Dogrusoz, U., Freeman, T.C., Funahashi, A., Ghosh, S., Jouraku, A., Kim, S., Kolpakov, F., Luna, A., Sahle, S., Schmidt, E., Watterson, S., Wu, G., Goryanin, I., Kell, D.B., Sander, C., Sauro, H., Snoep, J.L., Kohn, K., Kitano, H.: The Systems Biology Graphical Notation. Nature Biotechnology 27(8), 735–741 (2009)
49. Lex, A., Partl, C., Kalkofen, D., Streit, M., Wasserman, A.M., Gratzl, S., Schmalstieg, D., Pfister, H.: Entourage: Visualizing relationships between biological pathways using contextual subsets. IEEE Transactions on Visualization and Computer Graphics (InfoVis '13) 19(12), 2536–2545 (2013)
50. Mehlhorn, H., Schreiber, F.: TransID – the flexible identifier mapping service. In: Proc. International Symposium on Integrative Bioinformatics. pp. 112–121 (2012)
51. Mi, H., Schreiber, F., Novère, N.L., Moodie, S., Sorokin, A.: Systems biology graphical notation: Activity flow language level 1. Nature Precedings (2009)
52. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: Simple building blocks of complex networks. Science 298(5594), 824–827 (2002)
53. Moodie, S., Novère, N.L., Sorokin, A., Mi, H., Schreiber, F.: Systems biology graphical notation: Process description language level 1. Nature Precedings (2009)
54. Mueller, L.A., Zhang, P., Rhee, S.Y.: AraCyc: a biochemical pathway database for Arabidopsis. Plant Physiology 132(2), 453–460 (2003)
55. Novère, N.L., Moodie, S., Sorokin, A., Schreiber, F., Mi, H.: Systems biology graphical notation: Entity relationship language level 1. Nature Precedings (2009)
56. Partl, C., Kalkofen, D., Lex, A., Kashofer, K., Streit, M., Schmalstieg, D.: enroute: Dynamic path extraction from biological pathway maps for in-depth experimental data analysis. In: Proceedings of the 2012 IEEE Symposium on Biological Data Visualization (BioVis). pp. 107–114. BIOVIS '12, IEEE Computer Society, Washington, DC, USA (2012), `http://dx.doi.org/10.1109/BioVis.2012.6378600`
57. Purchase, H.: Experimental Human-Computer Interaction: A Practical Guide With Visual Examples. Cambridge University Press, New York, NY, USA (2012), `http://eprints.gla.ac.uk/78680/`
58. Rao, R., Card, S.K.: The table lens: merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In: CHI '94: Conference companion on Human factors in computing systems. p. 222. ACM (1994)
59. Roberts, J.C.: Exploratory visualization with multiple linked views. In: MacEachren, A., Kraak, M.J., Dykes, J. (eds.) Exploring Geovisualization. Elseviers (2004), `http://www.cs.kent.ac.uk/pubs/2004/1822`
60. Rohn, H., Junker, A., Hartmann, A., Grafahrend-Belau, E., Treutler, H., Klapperstck, M., Czauderna, T., Klukas, C., Schreiber, F.: VANTED v2: a framework for systems biology applications. BMC Systems Biology 6(139) (2012)

61. Schreiber, F., Colmsee, C., Czauderna, T., Grafahrend-Belau, E., Hartmann, A., Junker, A., Junker, B.H., Klapperstück, M., Scholz, U., Weise, S.: MetaCrop 2.0: managing and exploring information about crop plant metabolism. Nucleic Acids Research 40(1), D1173–D1177 (2012)
62. Shen, Z., Ma, K.L.: Mobivis: A visualization system for exploring mobile data. In: Proceedings of IEEE Pacific Visualization Symposium. pp. 175–182. IEEE VGTC (2008)
63. Shen, Z., Ma, K.L., Eliassi-Rad, T.: Visual analysis of large heterogeneous social networks by semantic and structural abstraction. IEEE Transactions on Visualization and Computer Graphics 12(6), 1427–1439 (2006), `http://dx.doi.org/10.1109/TVCG.2006.107`
64. Shneiderman, B., Aris, A.: Network visualization by semantic substrates. IEEE Transaction on Visualization and Computer Graphics 12(5) (2006)
65. Smoot, M.E., Ono, K., Ruscheinski, J., Wang, P.L., Ideker, T.: Cytoscape 2.8: new features for data integration and network visualization. Bioinformatics 27(3), 431–432 (2011)
66. Sommer, B., Künsemöller, J., Sand, N., Husemann, A., Rumming, M., Kormeier, B.: Cellmicrocosmos 4.1 - an interactive approach to integrating spatially localized metabolic networks into a virtual 3d cell environment. In: Fred, A.L.N., Filipe, J., Gamboa, H. (eds.) Proc. International Conference on Bioinformatics. pp. 90–95 (2010)
67. Stasko, J., Görg, C., Liu, Z.: Jigsaw: supporting investigative analysis through interactive visualization. Information Visualization 7(2), 118–132 (2008), `http://dx.doi.org/10.1145/1466620.1466622`
68. Steinberger, M., Waldner, M., Streit, M., Lex, A., Schmalstieg, D.: Context-preserving visual links. IEEE Transactions on Visualization and Computer Graphics (InfoVis '11) 17(12), 2249–2258 (2011)
69. Ward, M., Grinstein, G., Keim, D.A.: Interactive Data Visualization: Foundations, Techniques, and Application. A.K. Peters, Ltd. (2010)
70. Ware, C.: Information Visualization: Perception for Design. Morgan Kaufmann, 2nd edn. (2004)
71. Wattenberg, M.: Visual exploration of multivariate graphs. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06). pp. 811–819. ACM, New York, NY, USA (2006)
72. Zimmer, B., Jusufi, I., Kerren, A.: Analyzing multiple network centralities with ViNCent. In: Proceedings of SIGRAD 2012: Interactive Visual Analysis of Data, November 29-30, 2012, Växjö, Sweden. pp. 87–90. No. 81 in Linköping Electronic Conference Proceedings, Linköping University Electronic Press (2012)