Dipl.-Ing. Christian Partl

Visual Analysis of Paths in Multivariate Graphs with Applications in Molecular Biology

DISSERTATION

zur Erlangung des akademischen Grades
Doktor der technischen Wissenschaften

eingereicht an der

Technischen Universität Graz

Betreuer

Prof. Dr. Dieter Schmalstieg
Technische Universität Graz

Zweitbetreuer

Prof. Dr. Alexander Lex
University of Utah

Prof. Dr. Marc Streit
Johannes Kepler Universität Linz

Graz, November 2018

Abstract

Multivariate graphs are commonplace in many application domains, such as social sciences, transport, and molecular biology. The analysis of such graphs can be challenging due to large quantities of data, the complexity of relationships, and data heterogeneity. However, visualization can help humans to gain insights into this complex data. In this thesis, we introduce four novel visualization techniques for multivariate graphs: EnRoute, Entourage, Pathfinder, and ConTour. Paths play a primary role in all four visualization techniques. Either, paths are utilized to alleviate graph exploration, or the analysis of paths is the designated goal. EnRoute is a scalable visualization for the investigation of graph attributes, where paths can be interactively extracted from a graph to show them side-by-side with large amounts of heterogeneous attributes. Entourage allows users to investigate multiple graph partitions simultaneously by showing one partition in detail, while showing only contextually relevant paths for other partitions. Pathfinder is a query-driven technique that uses ranking and alternative path representations for the analysis of multiple paths in large graphs. Finally, ConTour allows users to investigate path relationships in heterogeneous linked datasets in a list-based interface. While the core concepts of all proposed visualization techniques are domain-agnostic, the visualizations were primarily developed to help experts to investigate multivariate graphs in molecular biology and drug discovery. Therefore, our techniques are evaluated in case studies and usage scenarios that illustrate their fitness to support domain experts with their analysis tasks.

Kurzfassung

Multivariate Graphen findet man heute in vielen Anwendungsbereichen wie den Sozialwissenschaften, dem Transportwesen und der Molekularbiologie. Die Analyse solcher Graphen ist in vielen Fällen aufgrund großer Datenmengen, der Komplexität von Beziehungen und der Datenheterogenität schwierig. Visualisierung kann jedoch helfen, diese komplexen Daten für Menschen leichter verständlich zu machen. In dieser Arbeit stellen wir vier neuartige Visualisierungstechniken für multivariate Graphen vor: EnRoute, Entourage, Pathfinder und ConTour. In all unseren Visualisierungstechniken spielen Pfade eine zentrale Rolle. Entweder werden Pfade verwendet, um die Graphexploration zu erleichtern, oder das Hauptaugenmerk liegt auf der Analyse von Pfaden selbst. EnRoute ist eine skalierbare Visualisierung für die Analyse von Graphattributen, bei der Pfade interaktiv aus einem Graphen extrahiert werden können, um sie zusammen mit großen Mengen heterogener Attribute darzustellen. Entourage ermöglicht es Benutzern, mehrere Partitionen eines Graphen gleichzeitig zu untersuchen, indem eine Partition im Detail angezeigt wird, während nur kontextrelevante Pfade für andere Partitionen angezeigt werden. Pathfinder ist eine auf Abfragen basierende Visualisierungstechnik, die Ranking und alternative Pfaddarstellungen für die Analyse von multiplen Pfaden in großen Graphen verwendet. Unsere letzte Visualisierungstechnik, ConTour, ermöglicht die Untersuchung von Pfadbeziehungen in verknüpften Datensätzen mithilfe einer auf Listen basierenden Benutzerschnittstelle. Obwohl die Kernkonzepte unserer Visualisierungstechniken domänenunabhängig sind, wurden die Visualisierungen in erster Linie entwickelt, um Experten bei der Untersuchung von multivariaten Graphen in der Molekularbiologie und der Wirkstoffforschung zu unterstützen. Daher werden unsere Visualisierungstechniken großteils in Fallstudien und Nutzungsszenarien evaluiert, die ihre Eignung zur Unterstützung von Domänenexperten bei ihren Analyseaufgaben veranschaulichen.

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Dissertation identisch.

Datum

Unterschrift

Acknowledgments

I would like to give credit to all the people who supported me during the years I pursued my PhD. First and foremost I would like to thank my supervisor Dieter Schmalstieg, who let me choose my research focus freely and provided valuable input to all research I did. I am also very grateful for his endurance during the write-up of this thesis and his constant motivation to bring it to an end.

Though living and working in different places than I did, Alexander Lex and Marc Streit are probably the persons I worked together with closest over the course of my PhD. Many thanks to both of you for your great ideas, the fruitful discussions, the constructive and relaxed work atmosphere, and your very positive and motivational attitudes. Also, thank you Alex for hosting me during my stay in Boston, and thank you Marc for hosting me during my visit in Linz.

I would also like to thank Hanspeter Pfister for his inputs and enabling my stay at Harvard University. I am very grateful for the collaboration with Samuel Gratzl, who I learned a lot from in terms of coding. Thank you Anne Mai Wassermann and Karl Kashofer for being such great collaborators from the biological domain and patiently introducing me to your fields of expertise.

There are many other colleagues, co-authors and collaborators I also want to express my gratitude to, especially Denis Kalkofen, Hendrik Strobelt, Nils Gehlenborg, Hans-Jörg Schulz, Thomas Geymayer, Holger Stitz, Peter J. Park, Mark Borowsky, Felix Reisen, Gerald Höfler, Stephan Jahn, Christian Viertler, and Stefan Sauer. Also, thanks to all co-workers at the Institute for Computer and Vision at Graz University of Technology for creating such a good work environment and atmosphere. Furthermore, I want to thank my current bosses, Stefanie Lindstaedt and Vedran Sabol, for their support to finish my thesis. Special thanks go to Silvia Miksch for being a referee of this thesis.

Finally, I would like to thank the people that are most important in my life. I want to thank Markus, Patritz, and Tanja for their friendship and the awesome times we had

together. I would also like to thank my grandparents and sisters for being the dear and great persons they are. At last, I want to thank my parents for always being there for me when I needed them and for supporting me throughout my whole live.

I acknowledge funding by the Austrian Science Fund through the Caleydoplex project (P22902), and the Province of Styria through the HTI:SMApp project "Tumorheterogenität" (A3-22.M-5/2012-21).

Contents

1 Introduction

Contents

Linked data and graphs are prevalent in various application domains. For example, such data is found in the form of social networks that connect people through friendship relationships, or business data, where customers are linked to products via transactions. In biology, networks depict interactions between biomolecules and represent existing knowledge about biological processes in cells and whole organisms. To gain a deeper understanding of these processes, various kinds of data, such as measurements about the activity of genes or the responses of cells to drug treatment are analyzed in context of these biological networks. The impact of insights gained in this domain is potentially high, as they can ultimately lead to development of more effective drugs, improved diagnostics, or more targeted treatment of patients. However, due to the large amounts and diversity of data, the analysis of networks in molecular biology is challenging.

Visualization has proven to be a powerful approach to data analysis. The idea behind data visualization is to use good visual encodings that exploit the capabilities of humans to easily recognize patterns in images. Proper data visualization can facilitate and speed up sense-making, when compared to typical raw data representations, such as large tables containing numbers. Finding intuitive ways of interacting with visualizations is of equal importance when dealing with large amounts of data that cannot be shown simultaneously. Another common approach to data analysis is to use statistics and automated methods that algorithmically extract valuable information. These methods typically scale well to large amounts of data and are good at answering specific questions. However, they are less suited to exploratory analysis, where it is not clear what to look for in advance.

Therefore, visualization and automated methods are often used in conjunction to exploit the strengths and compensate the weaknesses of each approach. To support effective visual data analysis, many factors need to be considered, including the data at hand, analysis goals, and domain-specific requirements. In this thesis, we focus on the visual analysis of graphs, mainly for the application domain of molecular biology. In the following, we will introduce basic concepts in graph theory and explain the problems addressed in this thesis in further detail.

1.1 Preliminaries

In the literature, many different terms have been established to refer to certain concepts in graph theory, but their usage is often inconsistent. In this section, we introduce relevant terms and concepts and explain how they are used within this thesis.

A **graph** describes relationships between objects. These objects are referred to as **nodes** or **vertices**, and the relationships are called **edges** or **links**. The terms graph and **network** are often used interchangeably in the literature. Here, we use the term graph when we refer to the mathematical concept or data structure and network when we talk about concrete instances of graphs, such as biological networks or coauthor networks. In a graph, an edge typically connects a pair of nodes. However, in **hyper-graphs**, a **hyper-edge** can connect an arbitrary number of nodes. An edge is called **directed** when the represented relationship has a designated source and target. Otherwise, an edge is called **undirected**. An edge can be associated with a **weight**, which typically defines the cost of choosing an edge to get from one node to another. In **multivariate graphs**, both nodes and edges may be associated with multiple **attributes** that can be of any data type, including textual, numerical, and categorical (ordinal or nominal) data. In real-world applications, these attributes are often drawn from tabular experimental datasets and mapped to the nodes.

Many graphs, such as biological networks, are divided into **partitions**, which define subsets of nodes and edges. In a **bipartite graph**, the nodes can be partitioned into two disjunct sets, such that no edge directly connects nodes within a set. A **k-partite graph** is the generalization of this concept, where the nodes can be partitioned into k disjunct sets. Linked tabular datasets that are typically stored in relational databases can be considered as a k-partite graph, with table entries as nodes and foreign key relationships as the edges.

A **tree** is a connected and acyclic graph. In many cases, trees depict hierarchical structures, where a root node is defined. The tree is then called **rooted**. In a tree, any two nodes are connected by a unique path. A **path** is an ordered sequence of nodes that are connected by edges. In this thesis, we mainly consider simple paths, where no duplicate nodes are allowed.

1.2 Problem Statement and Contributions

Graph analysis poses a multitude of challenges for visualization. Many of them are associated with scalability. The number of nodes and edges in real-world graphs often exceeds the amount of information that can be presented using conventional graph layouts without overwhelming the user and creating clutter. To alleviate scalability issues, graphs are sometimes divided into smaller partitions. Nevertheless, the analysis of multiple partitions still poses challenges with respect to scalability. Additional scalability problems are introduced by integrating large amounts of attribute data for nodes and edges. Data heterogeneity represents another challenge for visualization, as different data types, characteristics, and semantics require individual adaptation in terms of visual encoding and interaction. All of these challenges are to be faced in molecular biology: Large biological networks are partitioned into pathways modeling cellular processes and need to be analyzed in context of large cohorts of heterogeneous attribute data in order to gain new insights.

In the analysis of graphs and any sort of linked data, an important class of tasks is concerned with paths. Learning about how two suspects are connected in a criminal case or understanding why two genes are co-regulated are examples for important domain tasks that can be abstracted to path analysis tasks. However, due to the complexity of large networks, such path analysis is challenging.

In this thesis, we introduce contributions toward addressing these graph analysis challenges. While assuming underlying graphs of non-trivial size, we particularly want to address scalability issues along three dimensions: the number of graph **attributes**, the number of **partitions**, and the number of **paths**. Dealing with data **heterogeneity** in graph visualization is another important aspect of this thesis. We contribute visualization techniques where paths play a crucial role in addressing these issues: On the one hand, we contribute techniques that use the sensible sub-setting of graphs to paths as a vehicle to achieve scalability with respect to the number of graph attributes and partitions. On the other hand, we contribute techniques that define the analysis of path relationships as the primary goal and thus focus on the scalable visual exploration of multiple path relationships in graphs and linked datasets. Our techniques are primarily presented in context of the molecular biology domain and evaluated in comprehensive case studies. Table 1.1 gives an overview of the techniques and their primary analysis goals.

Technique	**Primary Analysis Goal**
EnRoute (Chapter 4)	Multiple heterogeneous graph attributes
Entourage (Chapter 5)	Multiple graph partitions
Pathfinder (Chapter 6)	Multiple paths
ConTour (Chapter 7)	Multiple paths in heterogeneous linked datasets

Table 1.1: *Overview of visualization techniques and their primary goals.*

Visual analysis of multiple heterogeneous graph attributes. In order to address issues regarding scalability and heterogeneity of graph attributes, we propose the **EnRoute** visualization technique in Chapter 4. It enables users to interactively extract single paths from a node-link representation of a graph. An extracted path is displayed in another view, juxtaposed with large amounts of heterogeneous attributes that map to the nodes of the path. This dual view approach enables users to both explore the graph topology and investigate associated attributes of a selected path in detail. EnRoute is presented in context of pathway graphs and genomic attribute data.

Visual analysis of multiple graph partitions. Effective visual analysis of large graphs requires to limit the amount of displayed elements in order to avoid overwhelming the user and creating clutter. Thus, we propose **Entourage** in Chapter 5, a focus+context technique for the visual analysis of large graphs that alleviates scalability issues by connecting multiple graph partitions. The core idea is to display one partition as node-link diagram at full scale, while only showing contextually relevant paths for other partitions. We indicate relationships between partitions by displaying glyphs on shared nodes that can be expanded to connecting links. To facilitate the exploration of attributes of individual paths in such large partitioned graphs, we combine Entourage with EnRoute. We demonstrate this technique with a biological network that is partitioned into multiple pathways.

Visual analysis of multiple paths. To support the simultaneous investigation of multiple alternative paths and their relationships, we introduce **Pathfinder** in Chapter 6. As the manual selection of possibly hundreds of paths in large graphs is not feasible, Pathfinder is based on queries that can be constrained in various ways. The resulting paths are shown in two complementing representations: a list that displays paths together with attributes, and a node-link diagram that gives an overview of the topology of paths. Query refinements through direct manipulation, ranking of paths by various scores, and visual means for path comparison are employed to quickly identify the most relevant paths for a given task. Pathfinder is demonstrated with a coauthor network and biological pathways.

Visual analysis of multiple paths in heterogeneous linked datasets. Finally, in Chapter 7, we propose **ConTour**, a visualization technique for the exploration of path relationships in k-partite graphs formed from heterogeneous linked datasets as they are found in relational databases. The analysis of relationships between different types of nodes is the primary focus of this technique. ConTour lists all nodes of each type in a column. Relationships can be explored through a combination of interactive highlighting, sorting, filtering, and nesting of columns. In contrast to Pathfinder, ConTour does not display whole paths with intermediate nodes, but rather indicates their existence between two endpoints. In this way, more path relationships can be considered simultaneously. ConTour is presented in context of various datasets that are used in drug discovery.

1.3 Collaboration Statement

This section provides an overview of colleagues and collaborators, who contributed to the work described in this thesis and without whom this thesis would not have been possible.

- **Prof. Dieter Schmalstieg** is the supervisor of this thesis and contributed to all publications in the conception phase and helped with paper writing.
- **Prof. Alexander Lex** is a co-supervisor of this thesis contributed significantly to all publications in terms of conception, evaluation, and paper writing. For the EnRoute, Entourage, and ConTour visualization techniques, he also made contributions to the implementation.
- **Prof. Marc Streit** is also a co-supervisor of this thesis and contributed to the concept and write-up in all publications.
- **Prof. Hanspeter Pfister** was involved in the conception and paper writing of Entourage, ConTour, and Pathfinder.
- **Dr. Denis Kalkofen** helped with the implementation and conception of the EnRoute and Entourage visualization techniques, and supported paper writing of related publications.
- **Dr. Samuel Gratzl** significantly contributed to the implementation of Pathfinder and also made contributions to the implementation of Entourage. He also provided paper writing support for related publications.
- **Dr. Hendrik Strobelt** made contributions to the implementation and write-up of ConTour.
- The **Institute of Pathology at Medical University of Graz**, and a research group at **Novartis** were our collaboration partners from the domain of molecular biology. They provided domain-specific requirements, use cases, background information in molecular biology, data, and feedback for our visualizations. At the Institute of Pathology our main contact was **Dr. Karl Kashofer**, and at Novartis the main contact was **Dr. Anne Mai Wassermann**.

1.4 Related Publications

The content of this thesis is based on the following peer-reviewed publications, which are listed in chronological order. For all of them, the author made significant contributions at all stages of the publication process, including conception, implementation, evaluation, and paper writing.

- **Christian Partl**, Alexander Lex, Marc Streit, Denis Kalkofen, Karl Kashofer, and Dieter Schmalstieg. *enRoute: Dynamic Path Extraction from Biological Pathway Maps for In-Depth Experimental Data Analysis.* In Proceedings of the IEEE Symposium on Biological Data Visualization (BioVis '12), 2012. [131]

  This paper describes the EnRoute visualization technique. It is the source for the ideas presented in Chapter 4.

- **Christian Partl**, Alexander Lex, Marc Streit, Denis Kalkofen, Karl Kashofer, and Dieter Schmalstieg. *enRoute: Dynamic Path Extraction from Biological Pathway Maps for Exploring Heterogeneous Experimental Datasets.* BMC Bioinformatics, 2013. [132]

  This paper is an extended version of the original BioVis paper describing the enRoute visualization technique and is targeted at the Bioinformatics community. Chapter 4 is based on this paper.

- Alexander Lex, **Christian Partl**, Denis Kalkofen, Marc Streit, Samuel Gratzl, Anne Mai Wassermann, Dieter Schmalstieg, and Hanspeter Pfister. *Entourage: Visualizing Relationships between Biological Pathways using Contextual Subsets.* IEEE Transactions on Visualization and Computer Graphics (InfoVis '13), 2013. [111]

  This paper describes the Entourage visualization technique and is the main source of Chapter 5.

- **Christian Partl**, Alexander Lex, Marc Streit, Hendrik Strobelt, Anne Mai Wassermann, Hanspeter Pfister, and Dieter Schmalstieg. *ConTour: Data-Driven Exploration of Multi-Relational Datasets for Drug Discovery.* IEEE Transactions on Visualization and Computer Graphics (VAST '14), 2014. [133]

  This paper describes the ConTour visualization technique and is the main source of Chapter 7.

- **Christian Partl**, Samuel Gratzl, Marc Streit, Anne Mai Wassermann, Hanspeter Pfister, Dieter Schmalstieg, and Alexander Lex. *Pathfinder: Visual Analysis of Paths in Graphs.* Computer Graphics Forum (EuroVis '16), 2016. [130]

  This paper describes the Pathfinder visualization technique and is the main source of Chapter 6.

2 Background in Molecular Biology

Contents

Molecular biology is a sub-domain of biology that deals with the functions and interactions of biomolecules within cells. It is closely related to domains such as genetics, biochemistry, and pharmacology. This chapter provides background information on molecular biology and introduces data and analysis challenges from biological research, which are addressed by the visualization techniques described in this thesis.

2.1 Gene Expression and Genomic Variation

The heredity information of all living species is stored in long double-stranded polymer chains called *deoxyribonucleic acid (DNA)* [6, p. 2ff]. The information is encoded in long sequences of four types of nucleotides that differ from each other in their base, which can be either, *thymine*, *guanine*, *cytosine*, or *adenine*. A strand of DNA can be used as template for self-replication and also for the synthesis of **proteins**. Both of these functions are important for cell division. Proteins are synthesized in the two successive steps of transcription and translation. In the first step, parts of the DNA are transcribed into very similar molecules called *ribonucleic acid (RNA)*. In the second and more complex step of translation, RNA transcripts, which serve as *messenger RNA (mRNA)*, guide the synthesis of proteins. The proteins are formed by stringing together up to 20 different types of amino acids according to the ”blueprint” of mRNA. A **gene** is referred to as the segments of the DNA that code for a protein or other functional molecules. The human DNA contains about 25,000 genes [6, p. 206]. **Gene expression** is the synthesis of coded proteins or other gene products. Proteins have several functions in the cell, such as maintaining structures, or acting as enzymes to catalyze reactions [6, p. 6]. In multicellular organisms, different cell types are created by synthesizing different sets of proteins and RNA molecules [6, p. 411]. The amount of proteins produced is determined by different regulatory mechanisms, such as proteins binding to short non-coding fragments of the DNA, i.e., *regulatory DNA*. Generally, the determination of how much of which protein is produced is referred to as **gene expression regulation**.

Mutations are permanent changes to the DNA that can affect the structure and function of synthesized proteins [6, p. 263ff]. While some mutations account for normal phenotypic variations, others are considered harmful and can lead to, for example, tumor growth. The most frequent genomic variations are *single nucleotide polymorphisms* [163], where individual nucleotides at a given position in the sequence are changed. Structural variations of the genetic sequence include insertions, deletions, and inversions. Deletions and insertions of large scale that affect one kilobase or more are commonly referred to as **copy number variations** [54]. Copy number variations can result in an increased or decreased number of copies of genes in the affected regions of the genome, which in turn may severely affect the levels of gene expression.

2.2 Biological Networks and Pathways

All living organisms can be considered as highly complex networks of biomolecules (genes, gene products, and metabolites) and biochemical reactions. It is the sum of tightly controlled and regulated interactions between these components that determines an organism's form and function. To capture these interactions and biochemical processes, biological networks are constructed. They are derived from experiments and existing knowledge extracted from the literature. Lopez et al. [174] give an overview of methods for the construction of these networks. Various types of biological networks are assembled to focus on different kinds of entities and/or interactions. The following network types play a crucial role in the study of cellular processes [7, 39]:

- **Metabolic networks** describe conversions of *metabolites*, i.e., chemical compounds, in chemical reactions, which are catalyzed by enzymes. In these reactions, sets of metabolites (substrates) are converted into sets of different metabolites (products). The products of one reaction may be the substrates for another reaction and vice versa. When regarding metabolites as nodes and reactions as edges, metabolic networks are directed hyper-graphs. However, these networks are often modelled as bipartite graphs with two types of nodes: metabolites and reactions.

- **Gene regulatory and signaling networks** convey information flow. Gene expression regulation captures regulatory influences of proteins on the expression of other proteins or functional RNA. Signal transduction describes any kinds of processes that transport internal or external stimuli to certain cell compartments and trigger a cell response, such as gene expression regulation. Typically, signaling information is passed in a cascade from one protein to another. Thus, the nodes in these networks represent genes, gene products and possibly other molecules, while edges represent different types of directed information flow.

- **Protein interaction networks** model physical interactions, i.e., bindings, between proteins. While the nodes in such networks represent proteins or sets of proteins, the typically undirected edges represent different types of interactions with very different time spans, ranging from very short and transient processes to permanent formation of complex protein assemblies.

In the study of biological networks, the series of actions among biomolecules that lead to specific biological effects are commonly described as **biological pathways**. In other words, a pathway is a meaningful subset of biomolecules and reactions whose interplay fulfills a function in a cell or organism. Some pathways describe metabolic processes. For example, the *Citrate cycle* pathway describes the final steps of the oxidation of carbohydrates and fatty acids. Other pathways, cover signal transduction, such as the *MAPK signaling* pathway. This pathway represents a well-studied signaling cascade leading from an activated cell surface receptor through a cascade of several proteins to the activation of DNA binding complexes, which regulate transcription of genes involved in the proliferation of cells and thus enable the cell to react to growth stimuli from its environment. Moreover, some pathways also highlight the processes involved in a particular disease, such as the *Glioma* pathway, which describes the molecular mechanisms dysregulated in brain cancers.

Several initiatives, such as KEGG [93], Wikipathways [96], or Reactome[49], create pathway maps and make them available to the scientific community. A comprehensive list of resources for pathway maps and biological networks is provided at `http://pathguide.org`. Most pathway databases provide manually curated pathway maps that present the molecular interactions in visually appealing node-link diagrams following specific drawing conventions. Figure 2.1 shows the *Citrate cycle* pathway and the *MAPK signaling* pathway from the KEGG and Wikipathways databases, respectively. In Figure 2.1(a) circular nodes represent metabolites, whereas rectangular nodes indicate enzyme-catalyzed reactions. In Figure 2.1(b), rectangular protein nodes are annotated using the gene names from which these proteins are transcribed. Different types of interaction are indicated by drawing links and arrows in different styles. To provide orientation, pathway maps sometimes include information about the spatial organization of the cell by drawing, for example, the cell wall (see Figure 2.1(b) left) or the cell nucleus. Many pathway maps also indicate related processes by including references to other pathways.

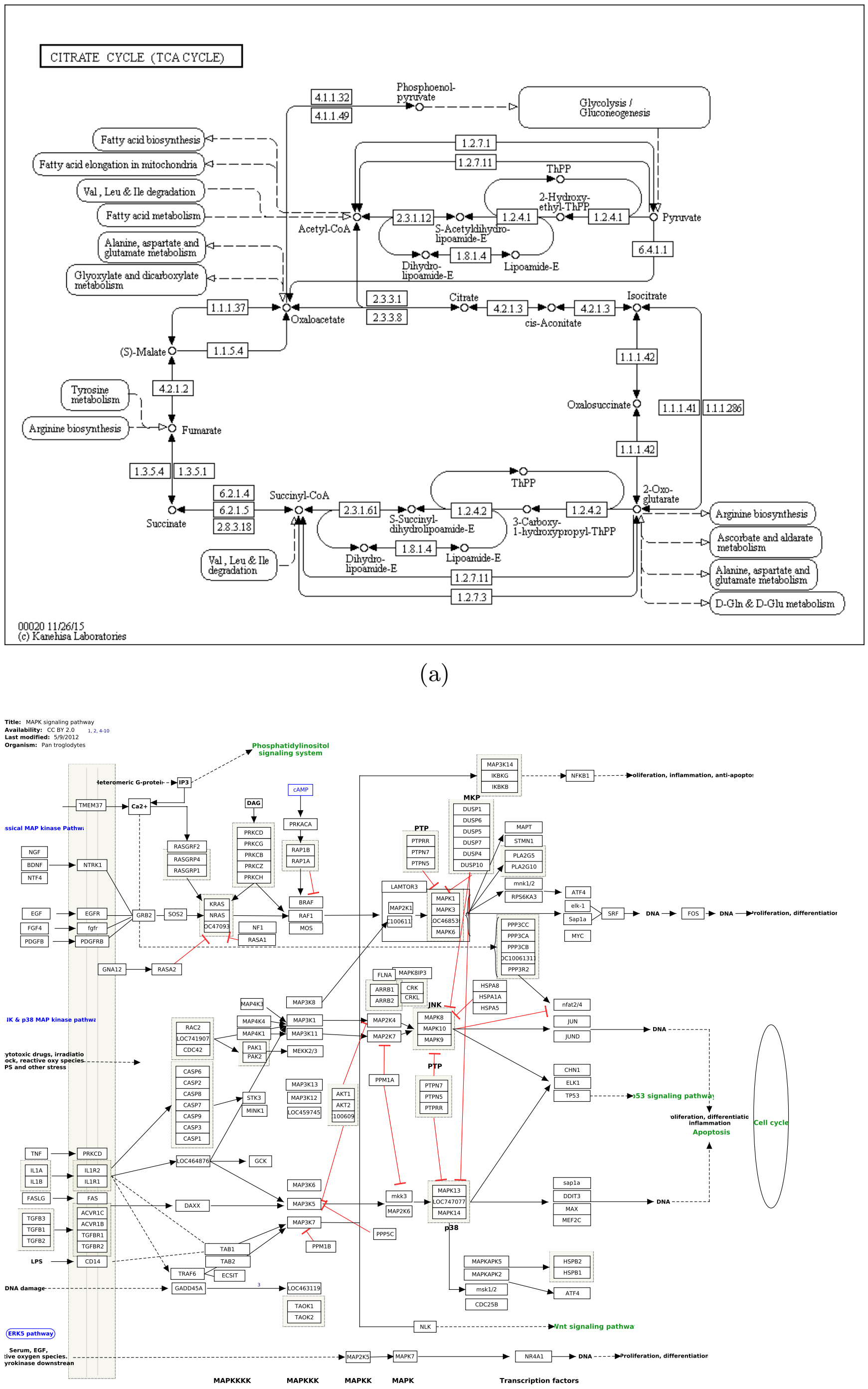

CITRATE CYCLE (TCA CYCLE)
Phosphoenol-pyruvate
Glycolysis / Gluconeogenesis
4.1.1.32
4.1.1.49
Fatty acid biosynthesis
Fatty acid elongation in mitochondria
Val, Leu & Ile degradation
Fatty acid metabolism
Alanine, aspartate and glutamate metabolism
Glyoxylate and dicarboxylate metabolism
1.2.7.1
1.2.7.11
ThPP
2-Hydroxy-ethyl-ThPP
2.3.1.12
1.2.4.1
1.2.4.1
Pyruvate
Acetyl-CoA
S-Acetyldihydro-lipoamide-E
1.8.1.4
Dihydro-lipoamide-E
Lipoamide-E
6.4.1.1
1.1.1.37
Oxaloacetate
2.3.3.1
2.3.3.8
Citrate
4.2.1.3
cis-Aconitate
4.2.1.3
Isocitrate
(S)-Malate
1.1.5.4
1.1.1.42
Tyrosine metabolism
4.2.1.2
Arginine biosynthesis
Fumarate
Oxalosuccinate
1.1.1.41
1.1.1.286
1.1.1.42
1.3.5.4
1.3.5.1
ThPP
6.2.1.4
6.2.1.5
2.8.3.18
Succinyl-CoA
2.3.1.61
1.2.4.2
1.2.4.2
2-Oxo-glutarate
Succinate
S-Succinyl-dihydrolipoamide-E
3-Carboxy-1-hydroxypropyl-ThPP
Arginine biosynthesis
Ascorbate and aldarate metabolism
Val, Leu & Ile degradation
1.8.1.4
Dihydro-lipoamide-E
Lipoamide-E
Alanine, aspartate and glutamate metabolism
1.2.7.11
1.2.7.3
D-Gln & D-Glu metabolism
00020 11/26/15
(c) Kanehisa Laboratories
(a)
Title: MAPK signaling pathway
Availability: CC BY 2.0
Last modified: 5/9/2012
Organism: Pan troglodytes
Phosphatidylinositol signaling system
Classical MAP kinase Pathway
JNK & p38 MAP kinase pathway
Cytotoxic drugs, irradiation, heat shock, reactive oxygen species, LPS and other stress
DNA damage
ERK5 pathway
Serum, EGF, Oxidative oxygen species, tyrosine kinase downstream
MKP
PTP
JNK
p38
DNA
Proliferation, differentiation
Proliferation, inflammation, anti-apoptosis
p53 signaling pathway
Proliferation, differentiation, inflammation
Apoptosis
Cell cycle
Wnt signaling pathway
Proliferation, differentiation
MAPKKKK
MAPKKK
MAPKK
MAPK
Transcription factors
(b)

Figure 2.1: *Hand-crafted pathway maps. (a) The KEGG Citrate cycle pathway and (b) the MAPK signaling pathway from Wikipathways. Available at* `http://www.genome.jp/kegg/` *and* `http://www.wikipathways.org`.

2.3 Analysis Challenges

With the rise of high-throughput technologies, such as next-generation sequencing [147], biological data is generated at a previously unmatched scale and efficiency, thus having the potential to dramatically accelerate research. Using these technologies, large amounts of diverse datasets are generated. Insights gained from analyzing these datasets lead to a better understanding of biological mechanisms and processes and, furthermore, may result in the development of more effective drugs, improved diagnostics, and more targeted treatment of patients. However, several challenges are to be faced in biological and pharmacological research when analyzing such large cohorts of complex and heterogeneous data. In this thesis, we provide several visual analysis solutions to alleviate the analysis process. We developed these solutions in close collaboration with experts from the Institute of Pathology at the Medical University of Graz, and with drug discovery researchers from the pharmaceutical company Novartis. In the following, we elaborate on analysis challenges these experts face in their research.

2.3.1 Analysis of Pathways and Genomic Profiling Data

Pathway maps are an important tool for studying biomolecular processes. As mentioned earlier, a wide variety of pathway maps is available that model interactions of proteins, chemical reactions and their catalyzing enzymes, as well as cellular signaling processes. As pathways represent consensus knowledge for processes in healthy organisms or specific diseases, they are often activated or inactivated in particular conditions, as diseases or other influences change the processes within the cells.

A common approach to study specific influences on cellular processes is to concurrently analyze genomic profiling data. *Genomic profiling data* refers to datasets that measure the activity or structural variation of genes. An example of genomic activity is the previously discussed gene expression (or *mRNA expression*), which indicates how much of a functional gene product, such as a protein, is produced. Two common forms of structural variation data are copy-number variation data, which records large-scale duplications or deletions of genes and thus indicates the number of present copies for each gene, and mutation data, which captures smaller changes within individual genes. Genomic profiling data is often captured for multiple conditions. A condition in this sense describes a group of measurements that are semantically homogeneous. Examples for this homogeneity criterion are samples taken from one species in a multi-species analysis, or samples of patients belonging to a subtype of a form of cancer.

Bringing together pathway maps and genomic profiling data is beneficial for their correct interpretation. The contextual knowledge contained in pathways about how biomolecules interact allows analysts to reason about effects observed in the genomic profiling data. Also, considering genomic profiling data may influence the interpretation of pathways. It can tell analysts that, for example, branches of a graph are inactive for a given condition. Examples of such observations are omnipresent in the literature. For instance, the gene *PTEN* is known to regulate the *phosphoinositide 3-kinase signaling pathway* [28], which is relevant for the regulation of cell-growth . If *PTEN* is mutated, the pathway becomes deactivated, leading to unchecked cell division and tumor growth. Jointly analyzing experimental data and pathways can help in reasoning about and predicting such effects for different conditions. Knowledge about how pathways are modulated by the genetic profile of groups or individual samples can help improving prognosis, treatment, and patient well-being. In Chapter 4, we introduce visual methods to alleviate the analysis of large-scale genomic profiling data in the context of pathways.

2.3.2 Analysis of Multiple Pathways

Biological networks are subdivided into pathways, which consist of biomolecules and reactions whose interplay fulfills a function in a cell or organism. This subdivion is, however, largely artificial, with the goal of reducing the complexity so that it can easily be comprehended by humans. While this approach in general is very successful, it also has its drawbacks. By focusing only on those components that are immediately relevant to a biological process under study, other interaction partners that might only become relevant under specific circumstances, e.g., the treatment of a disease, are left out. For example, cancer is often caused by defects in multiple genes and pathways. In these cases, the identification of genes that are shared between dysregulated pathways is of high relevance, because their products constitute prime targets for modulation by compounds, i.e., changing their activity using drugs. Furthermore, many drugs do not only bind to one target, but exhibit activity against multiple gene products. For example, *Dovitinib (TKI258)* is a drug that targets, among others, the products of the genes *EGFR*, *FGFR1*, and *PDGFR-beta*, which are well-known proto-onco genes (normal genes that, if mutated or highly expressed, can potentially cause cancer). In such cases, the study of all affected genes and pathways in a common reference framework is highly desirable to better understand the drug's effects. In particular, our collaborators at the pharmaceutical company Novartis were interested in addressing the following domain goals in drug discovery:

- **Judge side effects and safety of drugs.** Although drugs are often designed to modulate only one particular biological pathway, their action on the cell and the organism as a whole must be considered in their development to better assess their safety. Being able to analyze cross-talk between pathways can help in judging the potential risks associated with a compound early on in the drug development process [175].
- **Identify potential for repositioning of drugs.** Two alternative routes are usually taken in drug discovery: (1) developing new chemical entities and (2) finding new uses for already existing or previously failed drugs that have shown an adequate clinical safety profile [12]. The second route is usually more efficient, as such drugs can be approved quicker. As more and more knowledge about biological interactions

and refined pathway maps become available, it is quite possible that existing drugs can be repurposed for a broader spectrum of therapeutic indications. Inter-pathway analysis can help to identify potential new therapeutic uses for approved drugs.

- **Understand a drug's mechanism of action and drug sensitivities of cell lines.** The targets of established drugs are typically known. In many cases, such drugs inhibit one or multiple gene products. However, there are fluctuations on how cell lines (cell cultures) with distinct genomic profiles respond to the drugs [16]. Finding out why, for example, some cancer cells are killed by the drugs, while others survive, is one objective.

The data to be analyzed in order to achieve these goals can be classified into three categories: pathway data, genomic profiling data, and pharmacological profiling data. *Pharmacologic profiling data* essentially measures how cells react to compound exposure. A common measure is the *half maximal inhibitory concentration (IC_{50})*, which reports the concentration at which a drug achieves 50% inhibition, e.g., the drug concentration that is required to kill half of the treated tumor cells. The lower the concentration, the more effective the drug is and the more sensitive the cell line under study is to the drug treatment. Pharmacological and genomic profiling data are commonly jointly analyzed to, for example, identify reasons for the differential response that cell lines show to drug treatment. Integrating multiple pathways into such an analysis can make it much more targeted, since it allows analysts to focus on the processes influenced by the considered compounds and to explore related processes.

In Chapter 5, we introduce a novel visualization technique to facilitate the analysis of multiple pathways and their relationships. By combining this technique with the method described in Chapter 4, we also support the integration of genomic and pharmacological profiling data. Cross-pathway analysis is also picked up in Chapter 6, where we provide means to identify and investigate paths of interaction between biomolecules that stretch across multiple pathways.

2.3.3 Analysis of Biological Fingerprints and Related Data

For many years, drug discovery has focused on finding the “magic bullet”, i.e., the identification of a drug that selectively interacts with a disease-causing or pathology-relevant protein target [155]. However, with more and more data describing how drugs interact with biomolecules (bioactivity) and a better understanding of the biological network, evidence accumulates that this strategy employs an overly simplistic view of human disease and drug-target relationships. Indeed, existing bioactivity data suggests that approved drugs interact on average with seven different protein targets [125]. Furthermore, one protein target can be involved in many different biological processes. Therefore, its modulation by a compound can influence multiple, seemingly unrelated phenotypic traits, i.e., have multiple observable effects on the organism, both on a cellular and whole organism level. However, the same phenotype can be induced by compounds that interact with different protein targets, e.g., if the proteins are part of the same signaling pathway. Given this complexity of the biological system, classical structure-activity relationship analyses that study the effect of a compound set against one particular protein need to be complemented by techniques that allow for a more holistic view on the effects that a compound has on a biological network. In the pharmaceutical industry, historical experimental data can be leveraged and combined to generate so-called *biological fingerprints* that report the activity of a compound across dozens of experiments that were designed to monitor different cellular processes. In particular, the compounds studied by our collaborators have been profiled in at least 50 different cell-based screens at the pharmaceutical company Novartis [134]. These screens were tested for compound activities against a panel of diverse targets, pathways, and organisms. In each screen, all compounds were tested at a single concentration, and compound activities were reported in form of Z-scores, i.e., the number of standard deviations that a compound's effect in a screen differed from the mean response of all compounds tested in the screen. For each compound under study, its Z-scores were combined into a vector (the "fingerprint"), where each position was associated with a specific assay. In essence, the fingerprints represent a numerical characterization of different experimentally measured phenotypes. They are thus numerical descriptions of the observable effect of a drug on a cell or organism and provide a more comprehensive view on the manifold biological actions of a compound than simple protein-compound interaction data.

Comparison and clustering of compounds based on their biological fingerprints can lead to the detection of novel compound-target or compound-disease relationships. Therefore, our collaborators used a correlation-based similarity measure [177] to calculate a similarity matrix between all compounds, which was then used as input for hierarchical clustering. The resulting fingerprint clusters provided the basis for their data analysis, which has the following main goals:

- **Identify a drug's mechanism of action.** If a compound with an unknown mechanism-of-action falls into a cluster where all other cluster members are known to modulate the same protein target, it is conceivable that the compound also binds to this target.
- **Identify the biological process a drug modulates.** If compounds that bind to different targets cluster together, one can hypothesize that these targets are involved in the same biological process.
- **Identify new drugs for specific therapeutic indications.** A compound that clusters together with drugs for a particular therapeutic indication could be a novel candidate drug for this therapy, with potentially advantageous properties.

Although the biological fingerprints and clusters are at the core of the analysis, a large variety of data needs to be integrated in order to achieve these goals and paint a holistic picture. First of all, our collaborators considered meta-data of compounds in the form of activities against protein targets. These activities describe whether there is a known interaction for a particular compound-protein pair. Compounds typically interact with multiple proteins, and proteins also interact with multiple compounds. The dataset studied by our collaborators distinguishes between three types of interactions: target activation, inhibition, and binding. That means that compounds can in- or decrease the activity, i.e., the functional effect, of a protein. If, based on the reported bioactivity data, the direction of the interaction cannot be inferred, it is reported as a binding event. In addition, these activities report AC_{50} values, which are similar to the previously discussed IC_{50} values. They refer to the concentration of the drug at which 50% of the maximal response was achieved. For example, for a compound that inhibits an enzyme that cleaves other proteins, the AC_{50} value is the compound concentration at which the observed cleavage is reduced by 50%. AC_{50} values thus characterize the potency of a drug; the lower the value, the more potent the drug. As many of the compounds under study are approved drugs or clinical candidates, they were also classified into several therapeutic groups according to their indication. Finally, pathways can be used to study the biological roles of protein targets in the human organism.

In summary, our collaborators aim to discover relationships between various kinds of pharmacological and biological data. In Chapter 7, we present a visualization technique that facilitates untangling this complex network of compounds, biological fingerprints, clusters, activities, proteins, pathways, and therapeutic groups.

2.4 Biological Datasets

We demonstrate our visual analysis solutions with biological data from various sources. We use pathway data from KEGG [93] and Wikipathways [96], which contain hundreds of manually curated pathway maps for different species.

The used genomic profiling data is from three different sources: The *Broad-Novartis Cancer Cell Line Encyclopedia (CCLE)*[1] [16] is our first data source. It provides genomic profiling data for more than 500 cell lines, i.e., cultures of cells, from 23 different cancer tissues in addition to pharmacological data that records responses of each of these cell lines to a set of 24 approved cancer drugs or drug candidates. The second source is *The Cancer Genome Atlas (TCGA)* project[2] [164]. It collects genomic profiling data for more than 10,000 cancer patients for the molecular characterization of over 30 different cancer types. Finally, our third genomic profiling data source is a gene expression dataset generated by our collaborators at Medical University of Graz [94].

The pharmacological datasets used in Chapter 7 include about 1,100 compounds that have been extracted from the public bioactivity databases *ChEMBL* [58] and *DrugBank* [108]. For the generation of corresponding biological fingerprints, Z-scores from 105 different assays were considered. The hierarchical clustering of the fingerprints resulted

[1] `http://www.broadinstitute.org/ccle/home`

[2] `http://cancergenome.nih.gov`

in a dendrogram that was divided into 100 distinct clusters. All compounds were annotated with about 7,000 activities against protein targets that were extracted from the data repositories ChEMBL and *GVK*[3]. The about 1,100 considered proteins were mapped to their corresponding genes. Finally, the *Prous Integrity database*[4] was used to classify the compounds into about 400 therapeutic groups.

[3] http://www.gostardb.com/

[4] https://integrity.thomson-pharma.com/integrity/xmlxsl/

3 Related Work

Contents

Primarily driven by data analysis challenges in molecular biology, we developed several visual analysis solutions in the context of this thesis. These solutions touch upon various topics in multivariate graph visualization. In this chapter, we discuss existing concepts, approaches, and techniques that are related to our solutions or inspired them. We begin with an introduction of general concepts in graph visualization, including classifications of tasks performed on graphs and common graph representations. We follow up with a discussion of approaches to make graph visualizations scalable and related work in more specific domains that are especially close to our own. These include the representation of graph attributes, techniques for biological pathway visualization, the visual analysis of paths, and the visualization of k-partite graphs. Finally, we introduce the frameworks that were used to implement our visualization prototypes.

3.1 Basic Considerations in Graph Visualization

Data forming graph structures is present in various application domains, such as business, transport, social sciences, and biology. In the visualization community, a lot of effort is dedicated to provide support in the analysis and ultimately the comprehension of this complex data. A myriad amount of graph visualization techniques and approaches were developed addressing different problems. Some of them are general-purpose techniques, whereas others are specific to certain domains. Some are supposed to alleviate certain task, while others make visual graph analysis work with ever-growing datasets. In the following, we describe tasks associated with graph analysis and discuss common ways to represent graphs.

3.1.1 Tasks

The tasks a user needs to perform ideally guide the design choices with respect to visual representation and interaction when developing new visualizations. Throughout this thesis, we base the requirements for our visualizations on tasks. As user tasks are often formulated in a domain-specific way, they are translated into more abstract tasks to make them comparable across domains. In the literature, many frameworks and taxonomies for abstract visualization tasks are proposed [135]. For example, Amar et al. [10] defines ten low-level tasks that may be used as base units to construct more complex tasks. These low-level tasks are *retrieve value*, *filter*, *compute derived value*, *find extremum*, *sort*, *determine range*, *characterize distribution*, *find anomalies*, *cluster* and *correlate*. An alternative taxonomy, which also considers different levels of tasks, is proposed by Valiati et al. [168]. They distinguish between the operational tasks *visualize* and *configure*, the analytical tasks *identify*, *determine*, *locate* and *compare*, and the cognitive task *infer*. Tasks performed on graphs are of particular interest in the context of this thesis. Lee et al. [110] propose four main categories of graph tasks:

- **Topology-based tasks** focus on the structure of the graph. Examples of this category are finding the nodes adjacent to a given one, finding the shortest path between two nodes and finding the nodes that are accessible from a given node within a specified distance.

- **Attribute-based tasks** focus on the attributes of nodes and edges. Examples include finding nodes with specific attribute values and finding nodes connected by a particular edge type.

- **Browsing tasks** refer to tasks like following a specified path, or returning to a priviously visited node.

- **Overview tasks** allow to get quick estimates of the graph, such as determining its approximate size.

Lee et al. also illustrate how tasks from their classes can be decomposed into an extended set of the low-level tasks by Amar et al. Using the taxonomy by Lee et al. as basis, Pretorius et al. [135] suggest an alternative decomposition that is based on the tasks proposed by Valiati et al. and also provide a more thorough description of graph tasks.

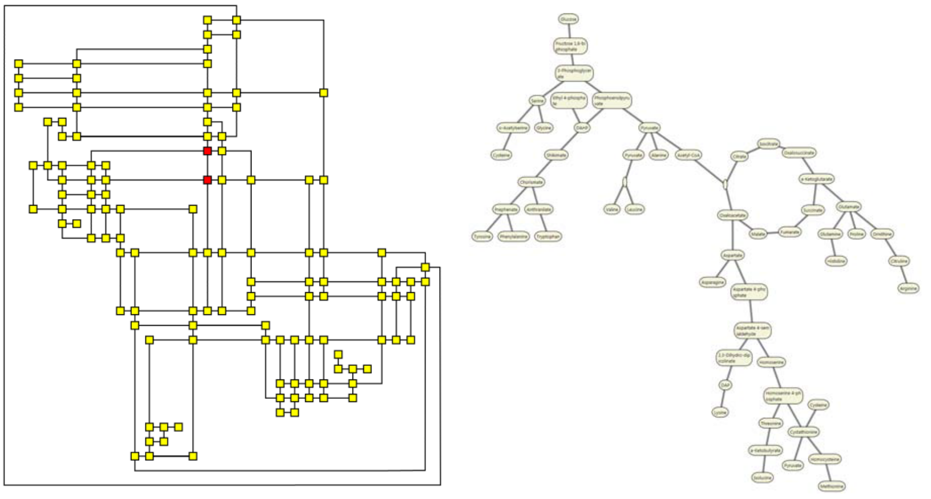
3.1.2 Visual Representation

Many factors can influence the choice of visual graph representation, such as the graph class (e.g., bipartite graph vs. tree), tasks to be performed, user preferences, and application domain. Although various different graph visualizations are proposed in the literature, most of them can be categorized into *node-link*, *matrix* or *implicit* representations, or are hybrids of these representations [142, pp. 9-13].

Node-link diagrams are probably the most common way to visualize graphs. In a node-link diagram, nodes are depicted as points, rectangles or other objects, and edges are represented as lines or arcs that connect the nodes. There are several *aesthetic criteria* for readable and visually pleasing node-link diagrams [37, pp. 3, 7]. These include, for example, the avoidance of edge-crossings, the use of uniform edge-lengths, and the uniform distribution of nodes. However, these criteria are often contradictory, and trying to optimize even a single criterion is NP-hard in many cases [37, p. 8]. Thus, many graph layouts are obtained using heuristics [98].

Force-directed methods represent an important class of graph drawing algorithms. These methods use functions that map the layout to an energy value. Low energy reflects layouts with adjacent nodes being placed at a predefined distance and non-adjacent nodes being farther apart. A layout is obtained by finding a (often local) minimum of the energy function [102]. A prominent example of these algorithms is Peter Eades' spring-embedder [46]. It uses a physical analogy of steel rings representing nodes, which exert repulsive forces on each other, and springs representing edges, which exert attractive forces between adjacent nodes. Kamada and Kawai [92] propose another important method that uses the analogy of springs. Especially their notion that distances between nodes in the layout should reflect their graph-theoretic distances, i.e., the shortest paths, was adopted by many other approaches. Kobourov [102] gives a comprehensive overview of force-directed approaches, which provide several improvements over aforementioned methods in terms of layout quality, scalability and stability.

Several layout methods for node-link diagrams put certain constraints on the placement of nodes or the routing of edges. *Linear layouts* [33] show nodes in a sequence and connecting edges as arcs, which is used, for example, to highlight repetitive patterns in sequential data like text or music [178]. *Layered layouts* arrange nodes in multiple connected layers. They are frequently used to represent directed acyclic graphs due to their suitability to be shown with a uniform orientation, e.g., top-down or from left to right. Layered graph drawing was probably most influenced by Sugiyama et al. [160]. They generate a layered layout in several steps, including the temporal removal of cycles, the assignment of nodes to layers, the minimization of edge crossings between layers and the straightening of edges. Several algorithms built upon this method and provide alternative solutions for the different steps. An in-depth discussion of algorithms and problems concerning layered layouts is provided by Healy and Nikolov [73]. In *grid layouts*, line



(a)

(b)

Figure 3.1: *(a) A grid layout [26]. (b) A force-directed layout with constraints for downward-pointing edges and drawing cycles as circles [44].*

segments of edges are drawn predominantly horizontally or vertically, which is considered aesthetically pleasing [42] and has been shown to have a positive effect on memorability [121]. They are often used in very large scale integration (VLSI) design, due to similar wire routing, and in pathway drawing. An example is shown in Figure 3.1(a). Several force-directed methods, such as the approaches by Dwyer [44] or Wang et al. [176], also include constraints. Examples are fixing the position of certain nodes, which is especially valuable in interactive visualization, choosing orientation according to edge direction, or showing cycles in the graph as circles. The latter two are illustrated in Figure 3.1(b).

In summary, there is a myriad amount of methods to layout node-link diagrams. An exhaustive overview is provided by Tamassia [161].

Matrix representations show the *adjacency matrix* of a graph. The nodes of the graph are represented as rows and columns, and a mark in the corresponding cell indicates the presence of an edge. In directed graphs, rows and columns represent source- and target-nodes, respectively. It has been shown that matrix representations perform well for simple tasks such determining whether two nodes are connected or finding the node with most connections [63, 97]. As matrices per definition do not suffer from clutter caused by edge crossings or node overlaps, they are a viable alternative to node-link diagrams when the number of nodes or density of the graph increases. However, compared to node-link diagrams, they are not well-suited for path-related tasks, especially on smaller graphs [63, 97]. Also, the readability of a matrix is strongly depends on the order of rows and columns [23, pp. 36, 254, 255]. Therefore, various matrix reordering algorithms were developed. Behrisch et al. [20] provide a comprehensive overview of these algorithms.

Implicit representations omit the drawing of edges, but implicitly indicate them through relative positioning, such as inclusion, overlap and adjacency. Implicit representations are mainly used for the visualization of trees. For example, *Treemaps* [87] draw a rectangle for the root node and subdivide its area it into smaller rectangles for each child node. This subdivision is repeated recursively for each level of the hierarchy. *Beamtrees* [172] show nodes as horizontal and vertical beams. Parent-child relationships are indicated by drawing the child node on the top of the parent node, but with different orientation. *Icicle plots* [104] show the nodes of each hierarchy level from top to bottom. Each node is depicted as a rectangle and its child nodes are placed right below it within the width of their parent node. Of course, variants with a different overall orientation, e.g., left to right, also exist. Further examples and an exhaustive discussion of implicit tree representations are provided by Schulz et al. [143].

Due to their intuitive representation of a graph and suitability for path-related tasks, we make extensive use of node-link diagrams throughout this thesis. While we mostly use pathway maps curated by humans in Chapter 4, 5 and 7, we also use layered and force-directed layouts for the path topology view in Chapter 6.

3.2 Exploration of Large Graphs

There are several limiting factors when trying to display and understand large amounts of data. These include the limitations of visual acuity, cognitive limits, computational limits and limits of the display [86]. Attempting to visualize the whole graph exceeds these limits in many real-world applications. Therefore, strategies must be employed to limit the amount of displayed information, while at the same time being able to explore the whole graph. Jankun-Kelly et al. [86] discriminate between strategies for *data transformation and reduction*, *visual mapping* and *view transformation*, which complement each other.

3.2.1 Data Transformation and Reduction

Strategies for data transformation and reduction aim to alleviate scalability issues at data level. Oliveira and Levkowitz [50] distinguish between the following methods:

- **Dimensionality Reduction.** These methods transform data from a high-dimensional space to a space with fewer dimensions, while trying to preserve overall data characteristics. Examples of these methods include *Principal Component Analysis* [82] and *Multidimensional Scaling* [184]. Van den Elzen et al. [169], for example, use dimensionality reduction methods to map the state of a whole network at different points in time to two-dimensional points. This way, similar states of the network a represented by points that are closer together, while dissimilar states are farther apart.

- **Subsetting.** Subsetting methods are used to reduce the data by selecting certain parts of it. This can be done using *queries* or *filters*, or with sampling methods such as *random sampling* [40]. Many graph visualization frameworks such as *Jigsaw* [151], *Ploceus* [117], *Orion* [75] or *GUESS* [3] support subsetting methods.

- **Segmentation.** Segmentation techniques produce multiple subsets of data that are based on data properties. For example, the partitioning of biological networks into pathways can be regarded as segmentation. Automated methods such as *clustering*, which generate groups of similar data, also falls into this category. In the context of graphs, clustering of nodes is typically based structural proximity, similarity of attributes or both [86]. An overview of clustering methods is given, for example, by Fortunato [53] and Berkhin [22].

- **Aggregation.** Aggregation methods produce aggregate representations of underlying data based on their properties or structure. Therefore, dimensionality reduction can also be regarded as special kind of aggregation. In the context of graphs, aggregation typically means to create a structurally more coarse representation of the original graph, where the number of nodes and edges is reduced [86]. In many cases, clustering is used for this purpose. In aggregated representations, nodes that summarize a subset of the original graph are referred to as *super-nodes*. Graph aggregation is employed in a wide range of general-purpose graph visualization frameworks, such as *Tulip* [14] or *CGV* [165].

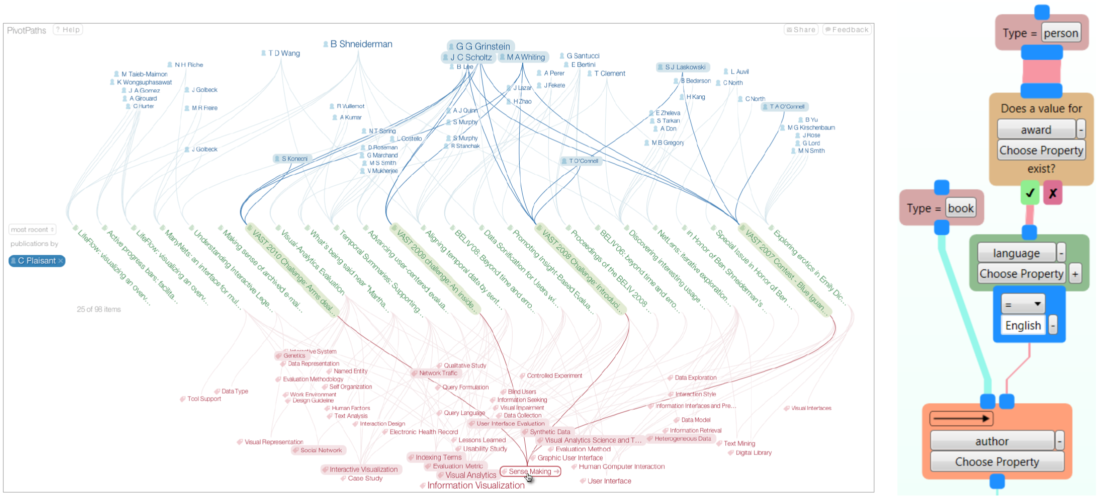
3.2.2 Visual Mapping

A significant factor that can impact scalability besides data transformation and reduction methods is the choice of the visual mapping [86]. Some visual encodings scale better than others, while typically giving up some quality at the same time. For example, node-link diagrams are probably the most natural way to present networks that is easily understood by most people. However, in dense graphs, they tend to produce clutter, which makes them hard to understand. In contrast, matrix representations do not suffer from clutter.

Advantages of different representations can be exploited by showing the data in multiple views [15]. For example, Dwyer et al. [45] implement the well-known paradigm of *overview+detail* for graphs. They show a coarse graph representation in an overview, while displaying a smaller portion of the graph in full detail in another view. Some techniques also take advantage of representing multiple subsets of the data using different encodings. For example, *NodeTrix* [80] tries to exploit the benefits of both node-link and matrix representations. In NodeTrix, the base representation is a node-link diagram, but dense subsets of the graph can be shown as embedded matrix representations that do not suffer from clutter. The basic concept of portals [128], where local regions within a visualization show a different view on the area they cover is adopted for graphs by Hadlak et al. [69]. They show aggregated graph in a node-link diagram as base-representation and allow the user to embed various more detailed visualizations of subgraphs on top on demand. Individual subset visualization is also used for non-graph data. For example, *VisBricks* [112] partitions numerical datasets into subsets and show each subset with the visualization technique most suitable for the contained data and task. We adopt this general idea of flexible subset visualization in the visualization technique introduced Chapter 5. It adapts the amount of detail shown for subsets based on the current focus of the user and available display space.

3.2.3 View Transformation

As not all information can be displayed at once, a user needs to be able to interactively navigate to different parts or show different aspects of the data. View transformations refer to interactive strategies to change the current view on the data [86]. Graph data can be explored interactively, for example, by using basic navigation concepts like *zoom and pan*, or more sophisticated methods, like *edge-based travelling* [166], where the camera is moved from the source to the destination of a selected edge. Many well-known visualization concepts also heavily rely on interactively changing the view to explore the data. For instance, in *multiple coordinated views*, interactions with one view also affect the display of other views. Corresponding elements are highlighted or movements in one view are propagated to another [15]. *Focus+context* techniques, where elements in focus are shown in detail, while other elements provide contextual information, also rely on interactive focus selection to explore the data. For example, Sarkar and Brown [141] use a fisheye lens to magnify the parts of a graph in focus and use mouse-dragging to change the focus.



(a) (b)

Figure 3.2: *(a) PivotPaths [41] shows a query result set of publications at the center. Connected authors and keywords are shown above and below, respectively. (b) A visual query by Haag et al. [68] that retrieves all books with an author who has won an award and whose language is English.*

A common way to express the current interest in a subset of a graph or any data is to use queries or filters. *Facetted search* is a popular and interactive method to query for data, which is implemented in many online shopping platforms such as *Amazon*[1]. In facetted search, the search space is typically divided into a single result set and multiple facets, which are displayed in different areas. For example, products represent the result set, and properties of the product like vendor or price represent facets. The result set is filtered by selecting a facet value, for example, a specific vendor. Changes to the result set are propagated to all facets to only show valid facet values for the current result set.

Flamenco [74] was one of the first systems to implement facetted search in this form. It uses uses a hierarchical organization of facets for the exploration of images. Yee et al. [181] showed that users preferred Flamenco's facetted search approach over keyword-based search. *PivotPaths* [41] is an approach that should encourage casual exploration of facetted datasets. As illustrated in Figure 3.2(a), it shows explicit connections between the result set in the center and facet values at the top and the bottom. PivotPaths uses animation to smoothly blend between filter states upon facet selection. A limitation of this approach is that only two facets can be shown. Heim et al. [77] introduced a more general method, where every facet can take the role of the result set. Both, the current result set and facets are shown as list-nodes in a node-link diagram. Starting from a node for the current result set, related entities can be added as facet nodes on demand. New facet nodes may be added to existing facet nodes as well. Links indicate the semantic relationships between nodes. Like in any facetted search, filters set by selecting entries from facet lists are reflected in the current result set and other facet nodes. An alternative approach to browse facetted data is taken in *PivotSlice* [186]. It displays all publications of a citation network as points in a two-dimensional area. Filters are defined by adding facet values to the left and bottom of that area, which partitions the area. Depending on whether publications comply with the filters, they are assigned to the corresponding partition. Our technique proposed in Chapter 7 is also closely related to facetted search approaches in the way it enables drill down into a complex dataset. However, our approach does not differentiate between facets and result sets. A technique that also does not draw this distinction is *GraphTrail* [43]. When drilling down into a large heterogeneous network, the exploration in GraphTrail starts with an initial chart that shows the nodes of a particular type. Applying filters based on node selection or attributes generates a new chart, which can either show the filtered nodes with the same type as the original chart, or related nodes with a different type. This chart generation can be repeated iteratively to drill down further into the data, which creates a visual history of the exploration process.

Another expressive way to formulate powerful queries for graphs is the usage of dedicated graph query languages, such as *SPARQL*[2] or *Cypher* for the *Neo4j*[3] graph database. However, writing such queries can be tedious and error-prone. Also, in most cases it can-

[1] https://www.amazon.com/

[2] https://www.w3.org/TR/2013/REC-sparql11-query-20130321/

[3] https://neo4j.com/

not be expected from users to know graph query languages. Visual interfaces for building queries alleviate these issues. For example, Haag et al. [68] provide a visual query builder for SPARQL that is based on the *flow/filter model* [183]. Figure 3.2(b) illustrates the method by Haag et al., where filters are applied in a consecutive manner from top to bottom. Different types of nodes represent item types, filters based on attributes, or filters based on relationships between entries. Links between the nodes indicate the flow of items with their width encoding the number of items. In Chapter 6, we also introduce a simple visual query builder that allows to specify constraints for a path search.

3.2.4 Graph Exploration Strategies

Described strategies for data transformation and reduction, for visual mapping and for view transformation are combined in graph visualizations in order to deal with large graphs. Overall, we roughly distinguish between *top-down* and *bottom-up* strategies for graph exploration.

Top-down approaches start by showing an overview using aggregation or sampling methods, and support drilling down into regions of interest. Overall, these approaches go along with Shneiderman's information seeking mantra [148] "Overview first, zoom and filter, then details on demand". For example, the *KEGG Atlas* [93] combines all metabolic pathways in a single overview, which supports zoom and pan navigation. Individual pathways can be shown in detail on demand. Similarly, the *Reactome Pathway Browser* [49] uses a hierarchical organization of pathways that follows the gene ontology (GO) biological process hierarchy[13, 60]. The hierarchies are represented in a circular node-link diagram, where pathways are shown as abstract nodes while edges represent either a hierarchical *is-a* relationship or indicate that pathways are shared between different hierarchies. Like in KEGG Atlas, individual pathways can be shown at full scale on demand. *AVOCADO* [119, 154] employ hierarchical and motif-based aggregation to show large provenance graphs in a layered node-link layout. Sub-graphs can be expanded on demand to show further details.

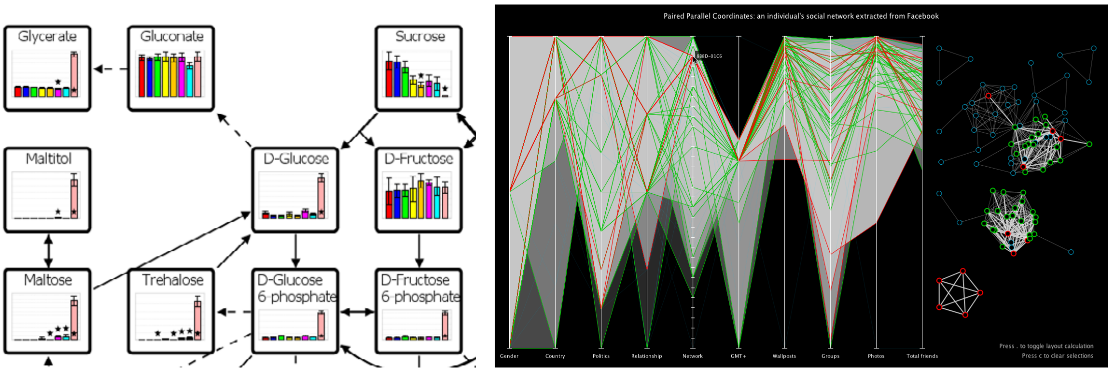
In bottom-up approaches, the exploration starts with a small subset of a graph that can then be expanded. For example, in *Treeplus* [109], graph exploration starts with a seed node. The graph can be then continuously expanded in a flexible tree layout by adding the neighbors of a selected node. Ham and Perer [171] summarize their graph exploration approach with "search, show context, expand on demand". They start with an initial textual search for nodes. The user then picks a node from the result set to show its neighborhood. An adapted version of Furnas' degree of interest function [55], which calculates the overall importance of each node in the graph from its a priori interest and the distance to the current focus node, is used to determine which nodes are displayed in this subgraph. This degree of interest function is reused to expand the subgraph on demand by selecting a new focus node. Abello et al. [2] also use degree of interest functions to select and expand graph subsets. They propose a modular specification for the degree of interest, which allows users flexibly express what is important for the exploration of time-varying networks.

In many cases, there is no clear cut between top-down and bottom-up approaches, as visualizations combine various exploration methods. This is also true for the approaches proposed in this thesis. However, overall, the approach described in Chapter 4 fits into the top-down category, as we display aggregated attribute data on a pathway, which can be shown in further detail on demand for selected subsets. The approach of Chapter 7 also fits this category, as we start by showing all items of a dataset that can be continuously narrowed down to items of interest. Our approaches in Chapter 5 and 6 can be overall regarded as bottom-up. In the former case, we start by selecting an initial pathway and iteratively add related pathways, while in the latter case, we start by querying a large graph for paths and continue the analysis based on this result set.

3.3 Visualization of Graph Attributes

Besides the representation of the graph topology, an important aspect of multivariate graph visualization is the visualization of attributes. Generally, visualization methods for multivariate data can be used for this purpose. Jusufi [90, p. 24] categorizes these methods into *projection-based approaches*, *coordinate axis-based approaches*, *icon-based approaches* and *pixel-based approaches*. Projection-based approaches, such as *scatterplots*, map attribute values into two- or three-dimensional space. *Scatterplot matrices* [29] overcome the scatterplot's limitation of being able to show only up to three attributes. The rows and columns of a scatterplot matrix represent attributes, and 2D scatterplots are displayed in the cells, showing all combinations of attribute pairs. In coordinate axis-based approaches, attribute values are shown on multiple axes. A prominent example of this category are *Parallel Coordinates* [83], where a sequence of parallel axes is drawn, each axis representing one attribute. Each data item is represented by a poly-line that intersects the axes according to its attribute values. Icon-based approaches encode attributes in various glyph properties. For example, *Chernoff Faces* [32] draw one face for each data item and encode its attributes as properties of the face, such as the size of the eyes or curvature of the mouth. Finally, pixel-based approaches show attribute values within very little display space such as single pixels. *Heatmaps*, for example, are frequently used for the display of gene expression data [47] in the domain of molecular biology. A heatmap is essentially a matrix that displays items as rows and attributes as columns or vice versa. Attribute values are color-coded in the cells of the matrix.

A major challenge in multivariate graph visualization is to visualize both, the topology of a graph and its attributes. We distinguish between three major approaches for graph attribute visualizations, which is similar to the classification by Jusufi [90, pp. 25-33]. *Integrated approaches* show both the graph topology and attributes in a single, integrated view. The graph information can also be distributed in *multiple coordinated views*. Finally, the topology can also be shown using an *attribute-based layout*. In the following, we provide examples for each of these classes. Due to their relevance in the context of this thesis, we will focus on examples from biological network visualization and the visualization of node attributes. Showing attributes in the form of mapped experimental data is a critical requirement and open challenge in biological network visualization [7, 140].



(a)

(b)

Figure 3.3: *(a) Cutout from a metabolic network shown with the VANTED system [89]. Attribute data is shown with bar charts embedded in the nodes. (b) Attribute visualization using two coordinated views [145]. Parallel coordinates on the left show attribute values of selected nodes in the topology view on the right.*

3.3.1 Integrated Approaches

The most primitive way of showing node or edge attributes in an integrated way is to encode them in visual properties of nodes and edges, such as size, shape or color. *On-node mapping* has been widely used to associate experimental data with corresponding nodes in biological networks. For example, colored rectangles representing the values of single experiments or an aggregation of multiple experiments are frequently used to augment the nodes of manually curated pathway maps [116, 127, 156]. When the base representation of the graph layout is not fixed, the amount of shown attributes can be increased by embedding small visualizations such as bar charts or line charts within the nodes [89, 179]. Figure 3.3(a) shows an example with embedded bar charts. An overview of further examples for on-node mapping in biological networks is given by Gehlenborg et al. [59].

Overall, integrated approaches associate attributes with nodes and edges in a very intuitive, straight-forward way. The amount of attributes that can be shown effectively is rather limited though.

3.3.2 Multiple Coordinated Views

The most common setup of multiple coordinated views to show multivariate graphs is that at least one view shows the network topology, while other views show attributes. The views are usually linked by means of synchronized highlighting, filtering or other methods. If, for instance, a user selects a node in the topology view, the corresponding attribute values are highlighted in other views, and vice versa. As illustrated in Figure 3.3(b), Shannon et al. [145] use a parallel coordinates plot to show attribute data, which is linked to a topology view of the graph. Parallel coordinates plots are also employed in Cerebral [17] for the visualization of experimental data in context of biological networks. Streit et al. [158] use heat maps in addition to parallel coordinates to show experimental data related to pathways.

A major benefit of using multiple coordinated views is that the amount of attributes that can be displayed is rather large. However, as separate views require interaction to show relationships between a single node and its associated data, the joint analysis of the topology and attributes is hindered, especially compared to integrated approaches.

3.3.3 Attribute-Based Layout

An example of a method that uses attribute-based layouts is *GraphDice* by Bezerianos et al. [24]. This method positions the nodes purely based on their node attributes in a 2D scatterplot, while still drawing the edges between them. In order to show more than two attributes, they adopt the metaphor of interactively rolling a dice horizontally or vertically in a scatterplot matrix and showing the scatterplot at the dice's position. Transitions between different scatterplots are animated using 3D rotations. Van den Elzen and van Wijk [170] also employ the strategy of positioning nodes according to configurable attributes to uncover, for example, geographical patterns by assigning longitude and latitude to the x- and y-axes in geo-referenced data. Instead of positioning the nodes according to individual attributes, Martins et al. [122] want to uncover overall similarities in node attributes. Therefore, they use multidimensional projection methods to determine the nodes' positions.

With attribute-based layouts, only a very limited amount of attribute values can be displayed at the same time. Also, as the layout is completely determined by attributes, the identification of structural properties of the graph can be severely hampered. However, in cases where the focus is on analyzing a limited set of attributes rather than on the graph topology, attribute-based layouts might be a good choice.

3.3.4 Hybrid Approaches

Hybrid approaches combine two or more of the previously discussed approaches. An example that combines multiple coordinated views with integrated approaches are *small multiples* [167, pp. 170-175]. Small multiples show the same configuration of a plot multiple times while changing one variable. Applying this concept to multivariate graphs means to use multiple instances that show the graph topology, each using an on-node mapping that shows different attributes. This principle is employed by Cerebral [17], which shows multiple instances of pathways with experimental data mapped to the node

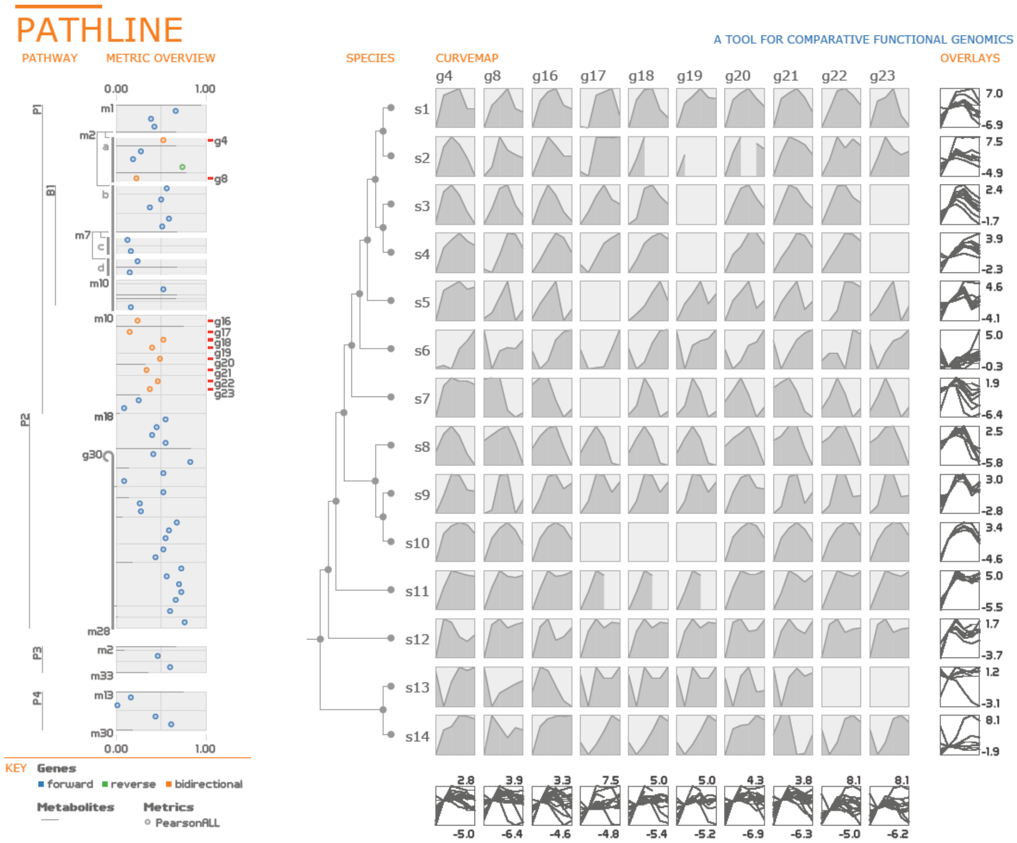


Figure 3.4: *The Pathline visualization technique [126] shows a pathway in a linearized form using special encodings on the left. Nodes are positioned horizontally according to aggregated attribute values. The Curvemap on the right shows time-series data for nodes selected in the pathway as line charts. Columns represent selected nodes, rows different species.*

color. Lex et al. [114] also use small multiples to show differences between experimental data associated with cancer subtypes on top of KEGG pathways. Using small multiples, the number of attributes that can effectively be shown at the same time mainly depends on the display space required to show the graph topology. Also, identifying all attributes of a single node across the multiples might take some time. However, synchronized highlighting of nodes alleviates this problem.

Pathline by Meyer et al. [126] is a tool for pathway analysis that combines multiple coordinated views with attribute-based layouts. As shown in Figure 3.4 on the left, Pathline displays a linearized version of a pathway, where branches and cycles are conveyed using special visual encodings. The horizontal position of the nodes is determined by an aggregated value of associated experimental data. The linearized view of the pathways helps comparing the node positions. Right next to the pathway, the system shows the *Curvemap* view, which displays experimental time series data across different species for selected nodes. As shown in Figure 3.4 on the right, columns represent selected nodes, while rows represent species. As with many multiple-view approaches, the visualization scales well with the number of shown attributes. The linearized pathway view also helps to quickly compare aggregated node values, but can hinder understanding the graph topology. However, the authors argue that this was not of primary interest to the users of Pathline.

3.3.5 Discussion

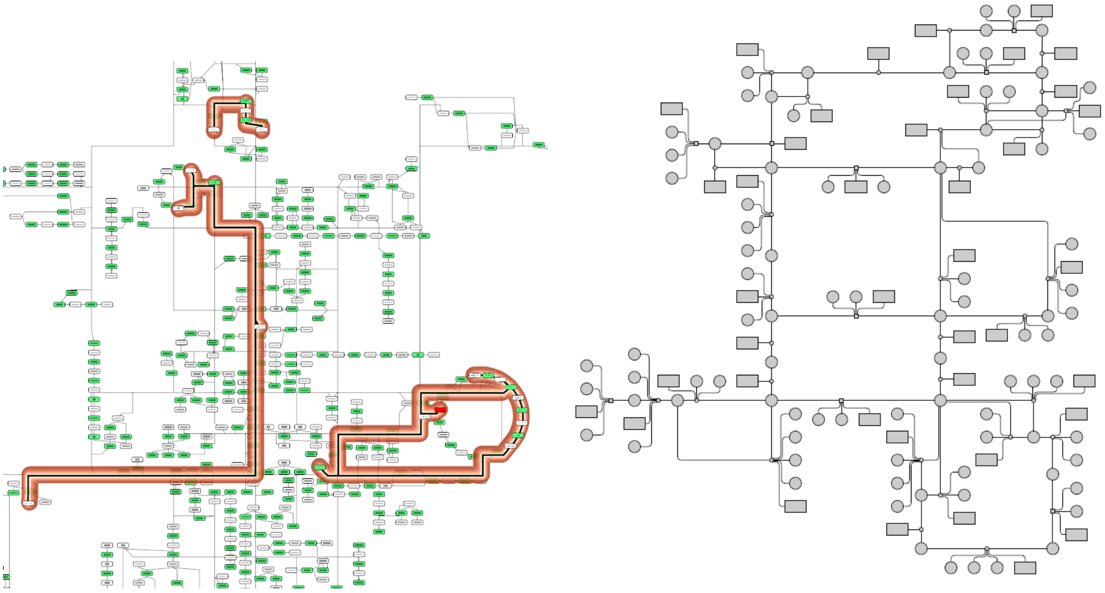
Multiple coordinated views can achieve the best scalability with respect the number of simultaneously shown attributes, since dedicated visualizations for multivariate data may be used. However, the association of the graph topology with attributes relies heavily on interaction and highlighting. With integrated approaches, the association between attributes and topology is straight-forward and typically does not interfere with the identification of structural properties, as attribute-based layouts do. However, the number of attributes that can be displayed is rather limited.

In Chapter 4, we introduce a hybrid method using multiple coordinated views and on-node mapping, which scales well with the number of attributes, while also addressing the issue of associating the graph topology with attributes. We do this in two ways. First, we show aggregated attributes on top of nodes in a graph topology view. Second, we allow users to extract a focus portion of the topology, i.e., a path, and, inspired by the overall idea of Pathline [126], display it linearized next to a detailed visualization of attributes. This helps to directly associate attributes with the path, while also having contextual information of the rest of the graph in the topology view. The general idea of displaying attribute data next to linear paths is also picked up in Chapter 6.

3.4 Visualization of Biological Pathways

As described in Section 2.2, many initiatives such as KEGG or Wikipathways use handcrafted node-link diagrams that follow certain drawing conventions for the representation of individual pathway maps. However, automatic layouts are also applied to pathways. For example, *Cytoscape* [150], a platform for biological network analysis, provides several layouts, including force-directed, layered and grid layouts. As illustrated in Figure 3.5(a), some layout algorithms also incorporate drawing conventions, such as representing cycles as circles or using orthogonal edges [106]. The explicit goal of the recently introduced method *HOLA* by Kieffer et al. [100] is to produce layouts that have a quality comparable to layouts created by humans. By comparing several manually curated layouts in a user study, they identified relevant aesthetic criteria that they incorporated in a multi-step algorithm. The resulting layout of a pathway graph is shown in Figure 3.5(b).

One challenge, but, at the same time, a critical requirement in pathway analysis, is the visualization of relationships between multiple pathways [7, 140]. We identified two major classes of visualizations that show relationships between pathways. The first merges



(a) (b)

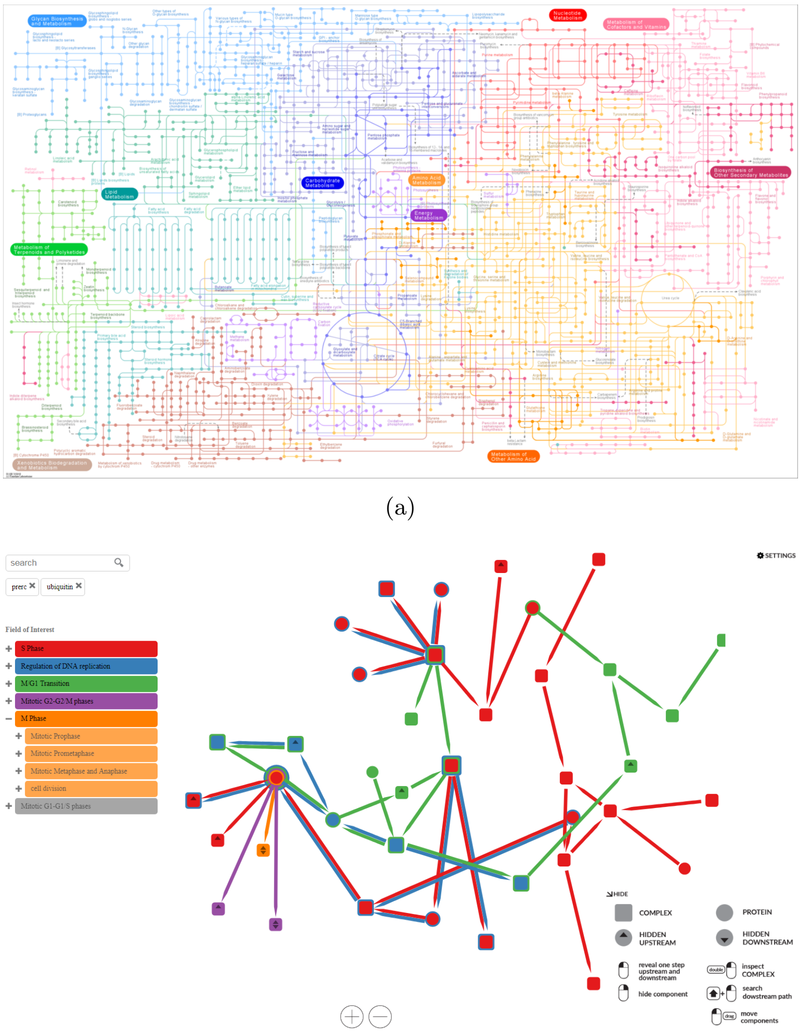
Figure 3.5: *(a) Automatic layout of multiple pathways by Lambert et al. [106] with one pathway highlighted. (b) Automatic orthogonal layout of a pathway with HOLA [100].*

the pathways into a single network. The second shows multiple pathways and visualizes relationships between them. In the following, we discuss examples from each of these classes.

3.4.1 Single Network Approaches

As an example of a single network approach, KEGG [93] provides the *KEGG Atlas*. As shown in Figure 3.6(a), it gives a high-level overview of the metabolic network that lets analysts drill down into individual pathways. Other tools, like *iPath* [180] or *Pathway Projector* [103] use the same underlying data, but improve the interaction with the atlas. While the original KEGG Atlas and iPath both use hyperlinks to replace the network overview with pathway maps on demand, Pathway Projector embeds node information directly on the all-encompassing map, thereby relying wholly on zooming and panning for navigation. Pathway Projector differs from the other two also in respect to how it represents gene/protein nodes. The KEGG Atlas and iPath both represent genes and embedded pathways only as edges between the metabolites, while the Pathway Projector actually shows nodes for genes, enzymes, and metabolites, which allows for on-node mapping of experimental data.

All of these techniques show an incredibly large amount of data on a single screen. For example, the KEGG Atlas for *E. coli*, a comparatively simple organism, summarizes 1365 genes/proteins, 1813 enzymes, and 1572 metabolites. In order to handle all this data, these



(a)

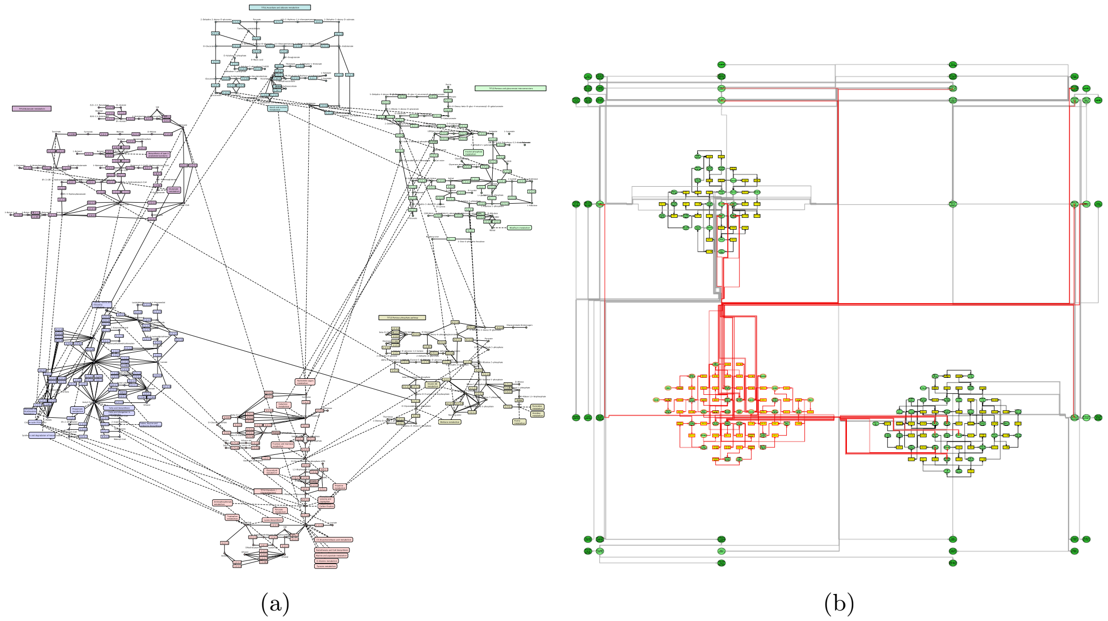
(b)

Figure 3.6: *(a) The metabolic network of the KEGG Atlas [93] summarizes information of several pathways. (b) Extended Line Sets [100] allow users to interactively add and remove nodes to the network. Pathway membership of nodes and edges is indicated by color.*

techniques rely on selecting a focus, either by zooming and panning, or by changing into a different view altogether. This makes it very hard to identify interrelations to parts of the network outside of the currently visible area. As these examples rely on a static layout, using layout-adapting methods to alleviate the problem, such as the *layout lens* [165], which pulls connected nodes into focus, is not possible.

The automatic layout algorithm by Lambert et al. [106] may also include multiple pathway maps in a single network. They added constraints to keep nodes that belong to a pathway as close together as possible. As nodes may be shared by multiple pathways, this can be challenging, which is why they also make use of node duplication. As shown in Figure 3.5(a), individual pathways may be highlighted within the overall network.

The Extended Line Sets approach [129] allows users to interactively add and remove elements to the overall network. Starting by adding initial nodes with a keyword search, the immediate neighbors of nodes can be added and removed interactively. As shown in Figure 3.6(b), pathway membership of nodes and edges is indicated by color coding. Although the set of pathway information that is of particular interest can be defined very precisely using this method, the addition and removal of information on a node-by-node basis might be cumbersome, as, for example, there is no way to directly add all nodes of a particular pathway. Also, the produced layout also does not adhere to typical biological drawing conventions.



(a)

(b)

Figure 3.7: *(a) Klukas and Schreiber [101] preserve the original KEGG layout for individual pathway maps, arrange them in a circle and draw links for shared nodes. (b) The grid layout by Rohrschneider et al. [138] shows three pathways in detail and other pathways as collapsed super-nodes. Relationships between pathways are indicated using links.*

3.4.2 Multiple Pathway Maps Approaches

The approach of showing multiple pathways and visualizing relationships among them is taken in earlier versions of the *Caleydo* system [158], for example, in the *Jukebox* [156] and the *Bucket* [113] techniques. Both of them arrange multiple pathways in a 2.5D layout, while one of the pathways serves as focus. Visual links are used to connect related nodes on demand. Additionally, other related pathways are displayed as thumbnails and can be added to the 2.5D layout on demand. The number of relationships that can be shown at a time is limited to one for both approaches.

Similar to the Caleydo techniques, Jusufi et al. extended the *VANTED* system to show one full-size pathway in focus, while arranging multiple pathways shown as thumbnails around it [91]. They use navigation glyphs that show whether individual nodes of the pathway in focus are connected to other pathways. A glyph has one petal for each possible link to other pathway groups. The direction of the petal corresponds approximately to the location of the associated pathway thumbnail and the length of the petal encodes whether there is a link or not. These petals can be used to navigate to other pathways. The system, however, does not show relationships on a node level in the pathways that are not in focus.

Several approaches embed multiple pathways in a single layout and show links connecting shared nodes. Additionally, they employ the concept of super-nodes, where pathways can be collapsed into single nodes and expanded on demand. An example is the approach by Klukas and Schreiber [101]. It uses either a force-directed or a circular graph layout of abstracted pathway nodes and their relationships. Each abstract pathway node can interactively be expanded to show all the nodes of the corresponding pathway in detail. As shown in Figure 3.7(a), these nodes are arranged using the original layout of KEGG pathway maps. They show all cross-pathway connections of individual nodes for multiple pathways at the same time. While this approach is reasonable for a limited set of pathways, the addition of more pathways continuously introduces clutter and reduces node size when trying to view all pathways.

A similar super-node approach is taken by Gerasch et al. [61], who also rely on KEGG layouts for the representation individual pathways. However, they make slight modifications to these layouts as they remove non-species-specific elements, which are not important in species-specific analyses, in order to reduce the overall amount of data shown.

Rohrschneider et al. [138] show multiple pathways at the same time, but use a grid-based automatic layout for representing an overall metabolic network, as illustrated in Figure 3.7(b). Their navigation approach is of particular interest, because they use the table-lens metaphor to switch between pathway super-nodes and detailed renderings of the pathway, which increases/reduces the space allocated for columns and rows of a particular cell. One particular issue in this layout is that links between nodes of different pathways tend to be quite long and have several bends, which makes tracing them rather difficult.

3.4.3 Discussion

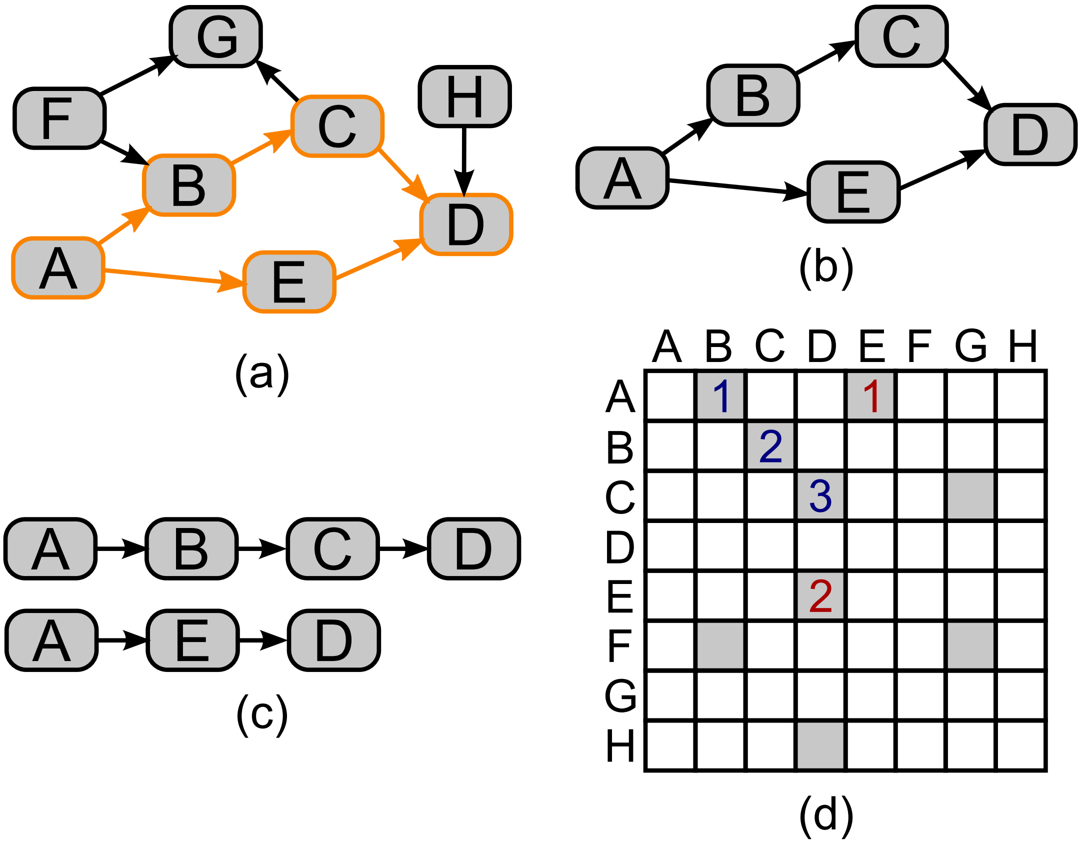
A major concern in most of the discussed approaches is that the amount of data that needs to be displayed to investigate relationships between multiple pathways increases very quickly. This can lead to a cluttered and overwhelming visualization. It also requires the user to use a zoomed out view in order to see all relationships, where details such as labels of individual nodes are no longer legible. The introduction of super-nodes alleviates the problem as pathways that are not currently in focus of the analysis can simply be collapsed. However, expanding whole pathways, which might consist of more than 50 nodes in larger pathways, already increases the amount of data that needs to be shown quite a lot. We argue that the additional information of subsets of pathways is enough in many cases, which increases the amount of pathways that can be simultaneously analyzed. In Chapter 5, we introduce a method for visualization of relationships between multiple pathways that relies on presenting only subsets of pathways that are currently not in the focus of the analysis, but contextually relevant.

3.5 Visualization of Paths

When visualizing paths in a graph, the first step is to specify the paths of interest. In principle, this can be done manually by the user by selecting relevant nodes and edges. However, especially in large networks, this approach is not feasible. Therefore, relevant paths are typically determined by path search algorithms. In many applications, for example, when searching for a route between two locations, the most important paths are the shortest paths between two nodes. The simplest shortest path algorithm is *breadth-first search*. It determines the shortest path between two nodes with respect to the number of hops. When the shortest path with respect to costs, i.e., the sum of weights of taken edges, should be found, *Dijkstra's* algorithm [38] is a popular choice for non-negative edges. The *A** algorithm [70] may perform better, when the costs of the shortest path can be estimated for each node. The *Bellman-Ford* algorithm [21] may consider negative edge weights, and the *Floyd-Warshall* algorithm [52] determines the costs between all pairs of nodes in a graph. In cases where the investigation of multiple alternative paths between two nodes is important, k-shortest path algorithms, for example, by Yen [182] or Eppstein [48], may be used.

Path search and path visualization are very common in route-planning or geographic information system applications. These tools provide users with multiple alternative paths to choose from. The routes differ in terms of time to reach the destination (fastest route), distance between source and destination (shortest route), but can also vary between alternate modes of transport (e.g., biking, driving, or public transport). While in commercial systems, such as *Google Maps*[4], overlays on top of the road-network without distortions dominate, there is work on visual simplification for these paths to make them resemble hand-drawn maps [5, 36]. Of course, visualizing paths for the purpose of directing people on a map is very different from visualizing paths in general graphs. Here, we focus on the visualization of paths in abstract graphs that have no physical mapping. We identified four major approaches for path visualization: *highlighting paths in node-link diagrams*, *displaying paths as node-link diagrams*, *showing path lists* and *displaying paths in matrix representations*. An overview of the approaches is given in Figure 3.8. In the following, we discuss these approaches and concrete examples in further detail.

[4] `https://maps.google.com/`



	A	B	C	D	E	F	G	H
A		1			1			
B			2					
C				3			▨	
D								
E				2				
F		▨					▨	
G								
H				▨				

Figure 3.8: *Four ways of visualizing paths in a graph, connecting the nodes A and D: (a) highlighting in a node-link diagram, (b) drawing only the subset of the graph connecting node A and D, (c) drawing a path list, (d) enumerating edges in a matrix.*

3.5.1 Path Highlights in Node-Link Diagrams

Highlighting paths in node-link diagrams, as shown in Figure 3.8 (a), is probably the most common path visualization method. Many tools support highlighting the shortest path between two nodes. For example, computing frameworks such as *MATLAB*[5], *R*[6] and *Mathematica*[7] support plotting static node-link diagrams with highlighted paths. Graph visualization frameworks and their plugins typically allow to interactively query for the shortest path by selecting source- and target-nodes within the node-link diagram or by using dedicated query interfaces. Examples are the two popular graph visualization tools *Cytoscape* [150] and *Gephi* [19]. *TimeArcTrees* [67] also highlight multiple shortest paths in a small multiples arrangement of node-link diagrams, each representing a different point in time.

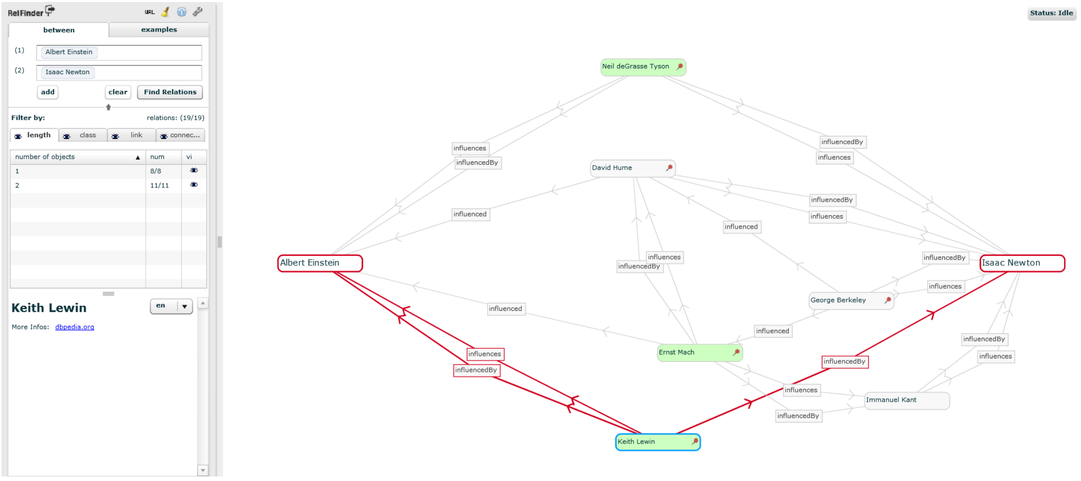
When multiple paths are highlighted in the same way, it might be difficult to discrimi-

[5] `https://www.mathworks.com/products/matlab.html`

[6] `https://www.r-project.org/`

[7] `https://www.wolfram.com/mathematica/`

nate individual paths due to shared nodes and edges. Interaction methods for highlighting one path at a time can mitigate the issue. Alternatively, multiple paths can be shown at the same time by using different highlights, such as by varying colors. However, the number of paths that can be shown in this way is limited by the number of discriminable highlights, which is about seven or slightly more when using color [72]. An advantage of the highlighting approach is that the topology of the graph can be perceived well. However, this approach only works well for smaller networks that can be displayed on the screen without introducing clutter, especially when details such as node labels should be legible.



(1) Albert Einstein

(2) Isaac Newton

Filter by: relations: (19/19)

number of objects ▲	num	vi
1	8/8	👁
2	11/11	👁

Keith Lewin

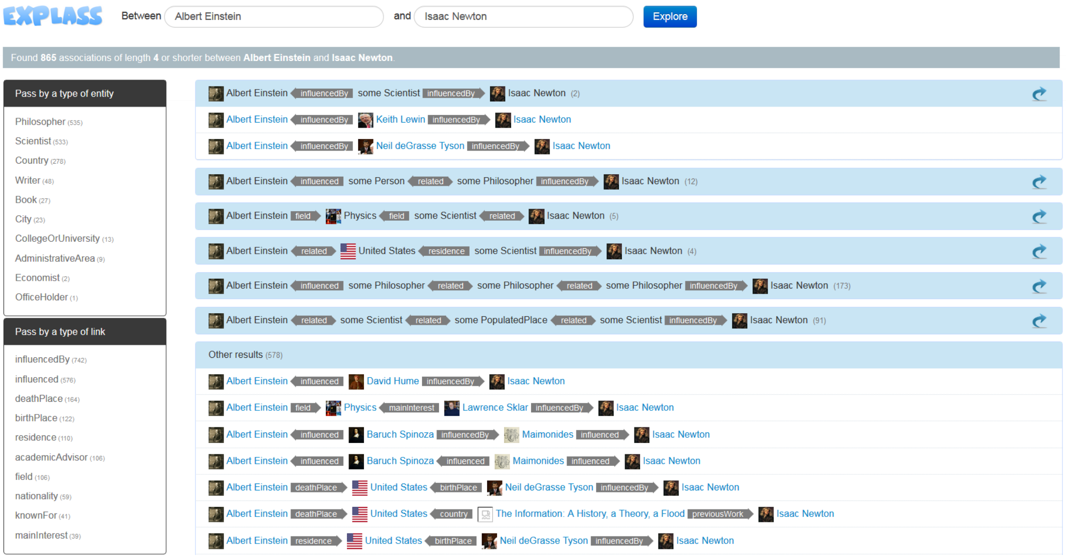
More Infos: dbpedia.org

Figure 3.9: *RelFinder [78] shows paths that connect Albert Einstein and Isaac Newton. The query interface is on the left and the resulting paths are shown as node-link diagram on the right. Paths that contain Keith Lewin are highlighted.*

3.5.2 Paths as Node-Link Diagram

Displaying paths as node-link diagrams, as shown in Figure 3.8 (b), is similar to the previous approach, with the critical difference that *only* nodes and edges of the paths are rendered. Typically, a dedicated query interface is used for path retrieval. *RelFinder* [78] is an example of such a technique for semantic web networks. It supports queries for paths between nodes and allows users to dynamically refine their queries or filter the result set based on different properties, such as path length or connection type. As shown in Figure 3.9, the start and end nodes of the paths use a fixed position on the left and right of the display, respectively. Other nodes are positioned using a force-directed algorithm. Edge types are indicated using labels in the node-link diagram, and detailed information about individual nodes can be accessed on demand in a different view. Selecting individual nodes highlights the paths that contain them. Tekusova and Kohlhammer [162] show paths connecting two nodes (companies in that case), but, in contrast to RelFinder, they also include important nodes that are not part of the paths.

As a consequence of only showing paths, approaches in this category scale better with respect to the number of paths than highlighting methods. However, topological analysis is limited to the paths as well. Moreover, in order to disambiguate individual paths, methods such as interactive highlighting need to be employed.



Found **865** associations of length **4** or shorter between **Albert Einstein** and **Isaac Newton**.

Albert Einstein ◀ influencedBy some Scientist influencedBy ▶ Isaac Newton (2)
Albert Einstein ◀ influencedBy Keith Lewin influencedBy ▶ Isaac Newton
Albert Einstein ◀ influencedBy Neil deGrasse Tyson influencedBy ▶ Isaac Newton

Albert Einstein ◀ influenced some Person ◀ related ▶ some Philosopher influencedBy ▶ Isaac Newton (12)

Albert Einstein field ▶ Physics ◀ field some Scientist ◀ related ▶ Isaac Newton (5)

Albert Einstein ◀ related ▶ United States ◀ residence some Scientist influencedBy ▶ Isaac Newton (4)

Albert Einstein ◀ influenced some Philosopher ◀ related ▶ some Philosopher ◀ related ▶ some Philosopher influencedBy ▶ Isaac Newton (173)

Albert Einstein ◀ related ▶ some Scientist ◀ related ▶ some PopulatedPlace ◀ related ▶ some Scientist influencedBy ▶ Isaac Newton (91)

Other results (578)

Albert Einstein ◀ influenced David Hume influencedBy ▶ Isaac Newton
Albert Einstein field ▶ Physics ◀ mainInterest Lawrence Sklar influencedBy ▶ Isaac Newton
Albert Einstein ◀ influenced Baruch Spinoza influencedBy ▶ Maimonides influenced ▶ Isaac Newton
Albert Einstein ◀ influenced Baruch Spinoza ◀ influenced Maimonides influenced ▶ Isaac Newton
Albert Einstein deathPlace ▶ United States ◀ birthPlace Neil deGrasse Tyson influencedBy ▶ Isaac Newton
Albert Einstein deathPlace ▶ United States ◀ country The Information: A History, a Theory, a Flood previousWork ▶ Isaac Newton
Albert Einstein residence ▶ United States ◀ birthPlace Neil deGrasse Tyson influencedBy ▶ Isaac Newton

Figure 3.10: *Explass [31] shows paths between Albert Einstein and Isaac Newton as a grouped list. Different groups are defined by certain patterns of entity and relationship types in paths.*

3.5.3 Path Lists

Path lists, as illustrated in Figure 3.8 (c), are also mostly based on queries. Each path is visualized in a separate row as part of a list, which causes nodes and edges that occur in multiple paths to be duplicated. An important aspect in paths lists is path ranking, as the most important paths should be displayed on top. Aleman et al. [8] ranks path as a weighted combination of properties, such as path length, trust, popularity and a user's interest in entity types. *SemRank* [11] computes a semantic score for each path, which specifies whether the path describes a rather expected or unexpected relationship. Users may specify their preference in expected or unexpected results for the ranking.

To give a better overview in large result-sets of paths, some approaches group similar paths together. For example, *Explass* [31] groups paths based on patterns that use similar entity or relationship types. Figure 3.10 shows an example where the first group summarizes two-hop paths between Albert Einstein and Isaac Newton with some scientist being the second node. Explass also supports filtering paths based on entity types, relationship types and path lenght. *RelClus* [185] uses a similar approach, but groups paths hierarchically.

Even though the amount of displayed information in path lists is larger than when displaying paths in a node-link diagram due to node and edge duplication, we argue that lists scale better with respect to the number of paths, as more paths only affect the list's length and not the legibility of paths. Ranking also helps to easily discover the most important paths. Another advantage over the display in node-link layouts is that individual paths can be discriminated easily without interaction. However, topological features are very hard to track.

3.5.4 Paths in Matrix Representations

As discussed in Section 3.1.2, matrices without extensions are considered ill-suited for path-related tasks and, consequently, are rarely used to visualize paths. There is, however, an extension by Henry and Fekete, *Matlink* [79], which supplements a matrix with explicit links drawn as arcs between the rows and columns, respectively, which is reminiscent to sequential node-link layouts. MatLink also automatically calculates the shortest paths between two selected nodes and highlights corresponding links. The authors compare path finding tasks in MatLink with a node-link diagram and a matrix that uses highlighting for shortest paths similar to Figure 3.8 (d). The authors found that users performed best on finding shortest path tasks with MatLink. Shen and Ma introduce an augmentation for matrices that visualizes multiple paths as links on top of a matrix [146], with the goal of combining the benefits of matrix layouts with good path-finding performance. However, their approach works well for only a limited number of paths.

Graffinity [99] is a visualization technique that focuses on giving an overview of paths between multiple different start and end nodes. To do so, Graffinity uses a *connectivity matrix*, where columns represent start nodes and rows represent end nodes. The color of cells indicates the number of connecting paths. Additionally, an *intermediate node table* gives an overview of in how many paths an intermediate node takes a certain position. In this table, rows show intermediate nodes, while columns show the position relative to the start node. Again, the colored cells indicate the number of paths.

In summary, none of the discussed methods use matrix representations directly to show paths. Either they augment a matrix representation with a node-link representation to show the paths, or the matrices actually do not show the paths themselves, but give a path overview.

3.5.5 Discussion

As matrix-based visualizations introduce node-link diagrams to enable path analysis, we argue that these approaches are not the right choice, when path analysis is in the focus. For the other approaches, important aspects are scalability and perception of the graph topology. While the latter is best in path highlighting methods, they can show the fewest paths. With path lists, many paths can be shown, but topological information is lost. Displaying paths as node-link diagrams is somewhere in the middle with regard to both aspects.

In Chapter 4, we resort to path highlighting, as the graph to be analyzed fits the screen and only few alternative paths between selected nodes need to be shown. However, in Chapter 6, we introduce a visualization technique to particularly address the challenge of scalable path analysis. It makes use of a path list display and a node-link diagram to show the results of path queries. Synchronized highlighting is employed to identify corresponding paths in the respective views. A key aspect to make the node-link diagram scalable is to show only the currently top-ranked and therefore most important paths in the path list. As the importance of paths may change during an analysis, we employ a flexible, extensible and interactive ranking of paths, which is missing in many path list approaches. As a holistic path analysis also includes the examination of path attributes, we incorporate the display of rich attribute data for paths, which is either completely neglected or limited to on-demand display for individual nodes in discussed approaches.

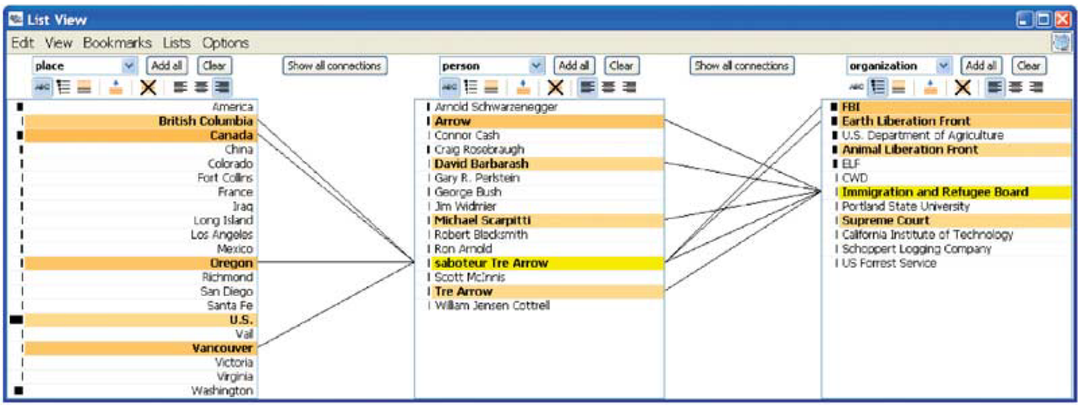


Figure 3.11: *The Jigsaw list view [151] shows places, persons, and organizations in separate lists. When an item is selected, related items are indicated with highlights and connecting links.*

3.6 Visualization of k-partite Graphs

Linked datasets that contain different types of data items are common in many application domains. Such datasets can be regarded as k-partite graphs, where each item represents a node and each partition represents the set of items of a particular type.

When visualizing k-partite graphs, different colors or shapes can be used to indicate the partition of a node in a node-link diagram. However, to make the association to partition even more obvious, and to put an emphasis on the fact that relationships are between node partitions, visualization techniques for k-partite graphs typically divide the screen space into separate areas, each showing the nodes of a single partition. For example, as illustrated in Figure 3.11, *Jigsaw's list view* [64, 151] shows different types of items in separate lists. Selecting an item highlights related items in the other lists and also draws links between adjacent lists. Schulz et al. [144] describe a similar table-based approach to visualizing bipartite biological networks. In contrast to Jigsaw's list view, each partition of the network is drawn in a table, which can be sorted based on various attributes. The two partitions are connected with links. Both tools visualize attributes within the cells. Ghani et al. [62] conducted a design study on multimodal social network analysis and developed *parallel node-link bands*, which are also similar to Jigsaw's list view. The evaluation of Ghani et al. shows that the parallel division of items into multiple columns was easy to understand and worked well for the domain experts in social network analysis. All the discussed methods use visible links to associate items of the individual lists. This is useful, when all list items fit on the screen, but less helpful, when the targets of links are not in the viewport. An approach that does not use lists to represent the partitions of a graph was proposed by Lieberman et al. [115]. They employ the *semantic substrates* concept [149] of visualizing connections between different semantic partitions of a network. Each partition is represented by items from different biomedical databases, such as a gene database, the PubMed document database, and the OMIM disease database. The items of a partition are displayed in a scatterplot and explicit links are drawn between them. An advantage of this approach over the list-based approaches is, that all items are displayed within the viewport. However, items with similar attributes might be displayed very close or even on top of each other, which can make disambiguation difficult.

In Chapter 7 we introduce a method for the exploration of linked data used in drug discovery, which also represents a k-partite graph. We use a list-based visual representation to show the graph, due to its clean layout and easy to understand visual metaphor. However, in contrast to discussed list-based techniques, we refrain from drawing explicit links in order to save screen space and reduce the visual complexity. Instead, we rely on interactive highlighting and reordering of items, and nesting of lists to uncover relationships.

3.7 Visualization Frameworks

When it comes down to visualizing data at hand, there are multiple ways to do so. For many people, using office software such as Microsoft Excel is the most straightforward way to go. However, the charting capabilities of these tools are limited to basic visualizations like static bar and pie charts. For a much more advanced and interactive visual data analysis, general purpose visualization software like *Tableau*[8], *TIBCO Spotfire*[9], *PowerBI*[10], or *Qlik*[11] may be consulted. For the analysis of graphs, there are more specialized tools like *Cytoscape* [150], *Gephi* [19], or *Graphviz* [57]. Although these tools are very powerful, they often still lack certain capabilities to effectively or efficiently deal with the special requirements of real-world data analysis problems. Therefore, there is a demand for the development of customized solutions for such cases.

[8] https://www.tableau.com/

[9] http://spotfire.tibco.com/

[10] https://powerbi.microsoft.com/

[11] https://www.qlik.com/us/

Most analysis problems described in this thesis originate from the domain of molecular biology and required the implementation of customized solutions. To avoid re-implementation of basic concepts such as data loading or synchronized highlighting and also to make use of synergistic effects when combining multiple solutions, all of the visualization prototypes proposed in this thesis are integrated in one of our research frameworks: *Caleydo* and *Phovea*.

3.7.1 Caleydo

Caleydo [158] is an open source visualization framework that was developed over several years. It is mainly aimed at data analysis in molecular biology. Caleydo integrates several visualizations in a multiple view system. These visualizations include *EnRoute* (see Chapter 4), *Entourage* (see Chapter 5), *ConTour* (see Chapter 7), and other visualizations not discussed within this thesis, such as *StratomeX* [114, 157], a visualization for the identification of cancer subtypes based on grouped patient data, or *LineUp* [66], a visualization for the interactive ranking of data items based on multiple criteria. Besides these more advanced visualizations, Caleydo also provides rather basic visualizations, which are often embedded within advanced techniques. Examples are a texture-based pathway viewer for the display of pathway images from KEGG and Wikipathways, or heatmaps [47] and parallel coordinates [83] for viewing tabular data like gene expression data.

Caleydo is implemented in the Java programming language for easy cross-platform development and deployment. It uses the *Rich Client Platform (RCP)*[12], a set of plugins to build rich client applications. The *Standard Widget Toolkit (SWT)*[13] is part of RCP and provides several user interface components and widgets with native look and feel. To build complex visualizations that cannot be assembled from the SWT widgets, Caleydo uses *JOGL*[14], a lightweight wrapper library that grants Java programs to access *OpenGL*[15] for rendering. The framework provides several facilities to alleviate rapid development of visualizations. They include, amongst others, support for loading tabular and pathway data, handling and propagating selections and other events across multiple views, basic statistical calculations, mapping between different biological identifiers, and color-schemes based on *Colorbrewer* [27]. The plugin-based architecture of the framework ensures extensibility. For example, a new visualization may be integrated by adding a view plugin.

3.7.2 Phovea

Phovea, formerly known as Caleydo Web [65], can be considered as a web-based successor of Caleydo. It is an open source framework for building client-server web applica-

[12] https://wiki.eclipse.org/Rich_Client_Platform

[13] http://www.eclipse.org/swt/

[14] https://jogamp.org/jogl/www/

[15] https://www.opengl.org/

tions, whose development started recently. Visualizations like StratomeX and LineUp were ported to this framework. The Pathfinder visualization technique presented in Chapter 6 was also developed using Phovea. One main benefit of Phovea is that the visualizations may be accessed from any computer without requiring a local installation as they run in the web-browser.

Phovea is based on a plugin architecture on the client and server. Therefore, visualizations developed for Phovea are essentially web applications that consist of client and server plugins. Client plugins are implemented with HTML, CSS, JavaScript, and TypeScript. Currently, most visualizations developed for Phovea use *D3*[16] for vector graphic-based display. However, other rendering technologies like *WebGL*[17] are also feasible. The server side is implemented with Python and uses the *Flask*[18] framework. Client and server communicate via REST and Websocket interfaces. To ease the development of web-based visualizations, Phovea provides support in several areas like data handling, selection management, provenance tracking, and client-server communication.

[16] `http://d3js.org`

[17] `https://www.khronos.org/webgl/`

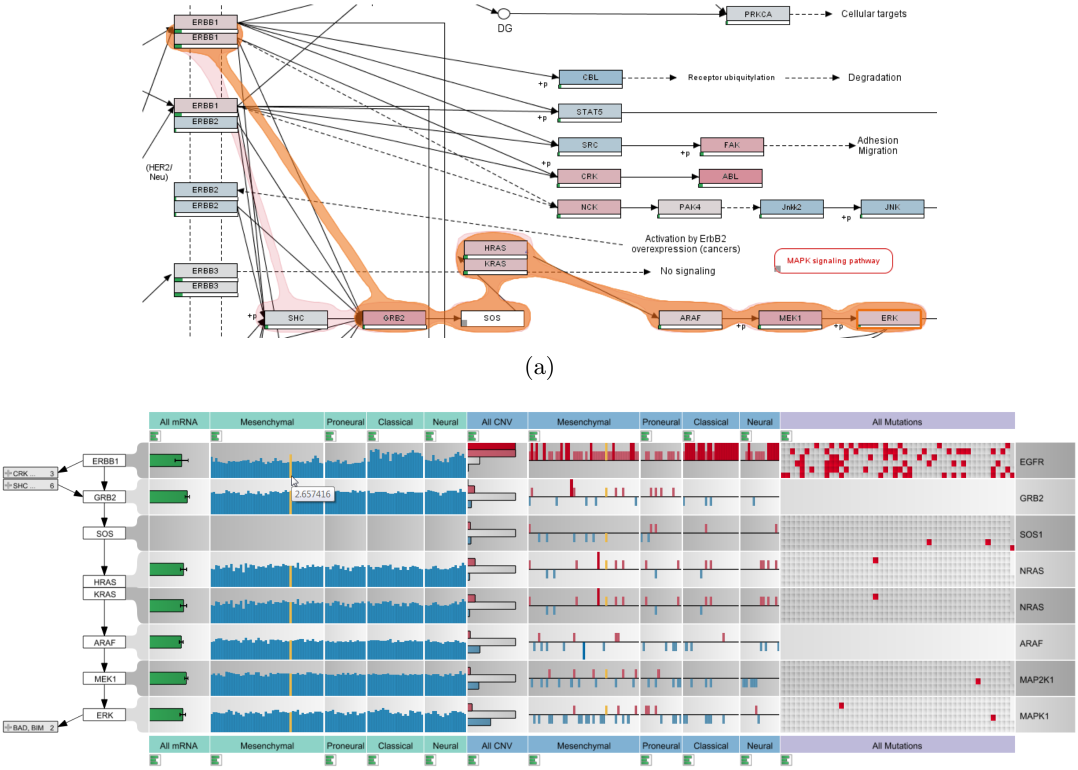
[18] `http://flask.pocoo.org/`

4 Visualizing Attributes in Graphs

Contents

Many real-world applications require the combined analysis of graphs and heterogeneous multi-dimensional data. In molecular biology, large amounts of experimental data are collected for different biomolecules. As discussed in Section 2.3.1, pathway graphs model interactions between these biomolecules and are consulted as contextual information to understand patterns and correlations within the data. Thus, from a more general point of view, large amounts of node attributes need to be analyzed in context of the graph topology. However, in most existing visualization techniques for displaying attributes in graphs, only a very limited amount of attributes can be shown, or it is hard to intuitively associate the graph topology with attributes.

In this chapter, we introduce **EnRoute**, an interactive visualization technique to show large amounts of heterogeneous attribute data, while preserving topological information. This is achieved using a dual-view approach and interactive selection of paths: Figure 4.1(a) shows the graph as a node-link diagram augmented with abstractions of attributes mapped onto the nodes. The selected path highlighted in orange is extracted and displayed top-down, juxtaposed with detailed representations of attribute data for the nodes, as shown in Figure 4.1(b). While this concept is domain-agnostic, EnRoute was specifically developed to address issues in the analysis of pathways in context of genomic profiling data. In the following, we first analyze the requirements for the domain of molecular biology, then introduce the EnRoute visualization technique and demonstrate EnRoute in two case studies.



(a)

(b)

Figure 4.1: *The dual-view setup of the EnRoute visualization technique. (a) The ErbB signaling pathway from the Wikipathways database is augmented to show abstract experimental data and a selected path (orange). (b) The selected path is extracted and displayed top-down along with associated experimental data from a TCGA glioblastoma multiforme dataset.*

4.1 Requirement Analysis

In collaboration with our partners from the Medical University of Graz, we have elicited the challenges experts face when analyzing individual pathways and associated experimental data. From this analysis we derived five requirements that must be met by a visualization system to successfully support the joint analysis of pathways and experimental data.

A I: The Scale Requirement – A common challenge in any type of visualization is scalability. When analyzing individual pathways, the scale of the graph is hardly a problem. The problem of scale is primarily concerned with the large number of experiments and experimental conditions. Scaling to dozens of experimental conditions or groups and hundreds of experiments is a requirement for integrated pathway analysis.

A II: The Heterogeneity Requirement – Modern biological studies often include a wide array of complementary, but heterogeneous experimental datasets. While, for example, mRNA expression data are direct measures of the gene activity, copy number or mutation data are used to reason about the causes of deviating expression values. These heterogeneous datasets need to be presented using different visualization techniques, as they differ in terms of data type. For example, mRNA expression data is numerical, copy number data is a hybrid categorical/numerical dataset, which is often binned into ordinal (ordered categorical) data, and mutation status data is nominal (unordered categorical). In order to analyze these different kinds of data in context of pathways, the visualization system needs to handle all of them simultaneously and also represent each of them using suitable visual encodings.

A III: The Multi-Mapping Requirement – Many nodes in pathway maps are gene products summarizing various entities such as RNA, enzymes, proteins, etc., which have complex relationships. One gene can be the template for multiple proteins with slightly distinct domain compositions called isoforms. In some cases, pathways also summarize a whole gene family into a single node, where different genes produce functionally similar proteins. This is what we call a multi-mapping: One node in a pathway actually represents multiple entities, and, therefore, multiple entries of an experimental dataset can be associated with this node. As understanding this complexity is essential for judging effects of experimental data on a pathway, it is critical to convey multi-mappings adequately.

A IV: The Layout Constraint Requirement – The layout of pathways is either produced manually by experts, or automatically. Manual pathway layouts follow biological drawing conventions, by, for example, drawing cycles in circles or using pseudo-orthogonal edges. Also, these carefully hand-crafted layouts contain rich meta-data and annotations that indicate, for example, the cell compartments in which specific processes occur. As discussed in Section 3.4, automatic layouts either try to respect these conventions, or optimize global layout properties using, for example, algorithms for force-directed layouts. We observed that domain experts prefer manually created layouts or, at least, layouts that adhere to biological drawing conventions. A reason for this might be that biologists are often intimately familiar with the layout of particular pathways and are reluctant to see it changed, as this requires additional effort on their side. While showing experimental data is easier in automatically generated layouts, as the layout can be adapted to suit the representation, a good visualization technique for joint analysis of experimental data and pathways also needs to work with the large baseline of existing, manually produced pathways.

A V: The Topology-Attribute Coexistence Requirement – Analyzing experimental data in the context of cellular processes can be described as tasks on a graph. We distinguish between two main types of tasks conducted on pathways: tasks that are based on the topology of the underlying graph, and tasks that are based on the node or edge attributes of the graph [110]. As discussed in Section 3.1.1, **topology-based tasks** are concerned with the structure of the graph, e.g., which nodes can be reached from a given node, or what the articulation points of a graph are. An example for a topology-based task in pathway analysis is to find all processes that are influenced by a receptor at the cell surface. **Attribute-based tasks** are concerned with analyzing node or edge attributes. Edge attributes in pathways commonly describe the type of a relationship between two nodes, such as transcriptional activation or inhibition, while mapped experimental data represents the majority of node attributes. An example for an attribute-based task for pathway analysis is to find all genes in a pathway that are mutated in a large number of the mapping samples. Visualization techniques for graphs are usually optimized for either topology-based task or for attribute-based tasks, but are rarely suitable for both at the same time. When analyzing pathways and experimental data, however, both types of tasks need to be addressed simultaneously. The two central questions an analyst is trying to answer when analyzing both pathways and experimental data are (a) how the experimental data for particular experimental conditions or groups of samples influences the topology of the graph and (b) how effects observed in the experimental data can be explained using the topology of the pathways. Consequently, an effective visualization technique has to enable both: an in-depth analysis of the topology and the pathway attributes.

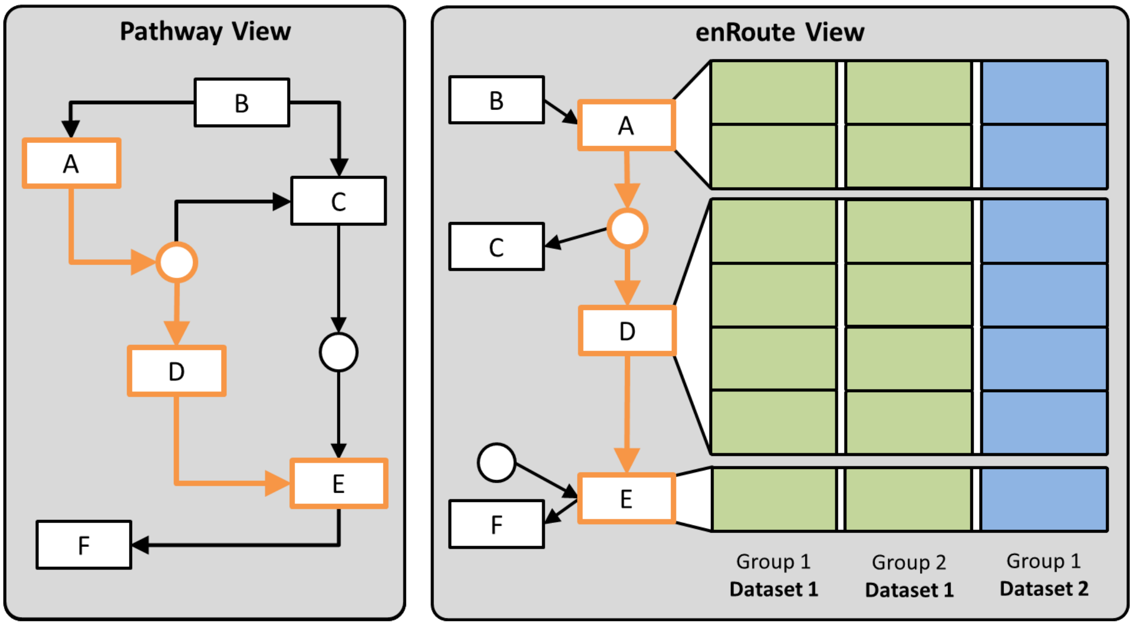


Figure 4.2: *The EnRoute visualization technique with its two basic building blocks: the pathway view and the EnRoute view. The pathway view shows the pathway map in its original layout. In the example shown, a path from node A to E is selected, which is extracted and shown on the right in the EnRoute view. Due to the linear layout of the extracted path, the associated experimental data can be visualized next to it. The data can originate from different datasets and can be grouped.*

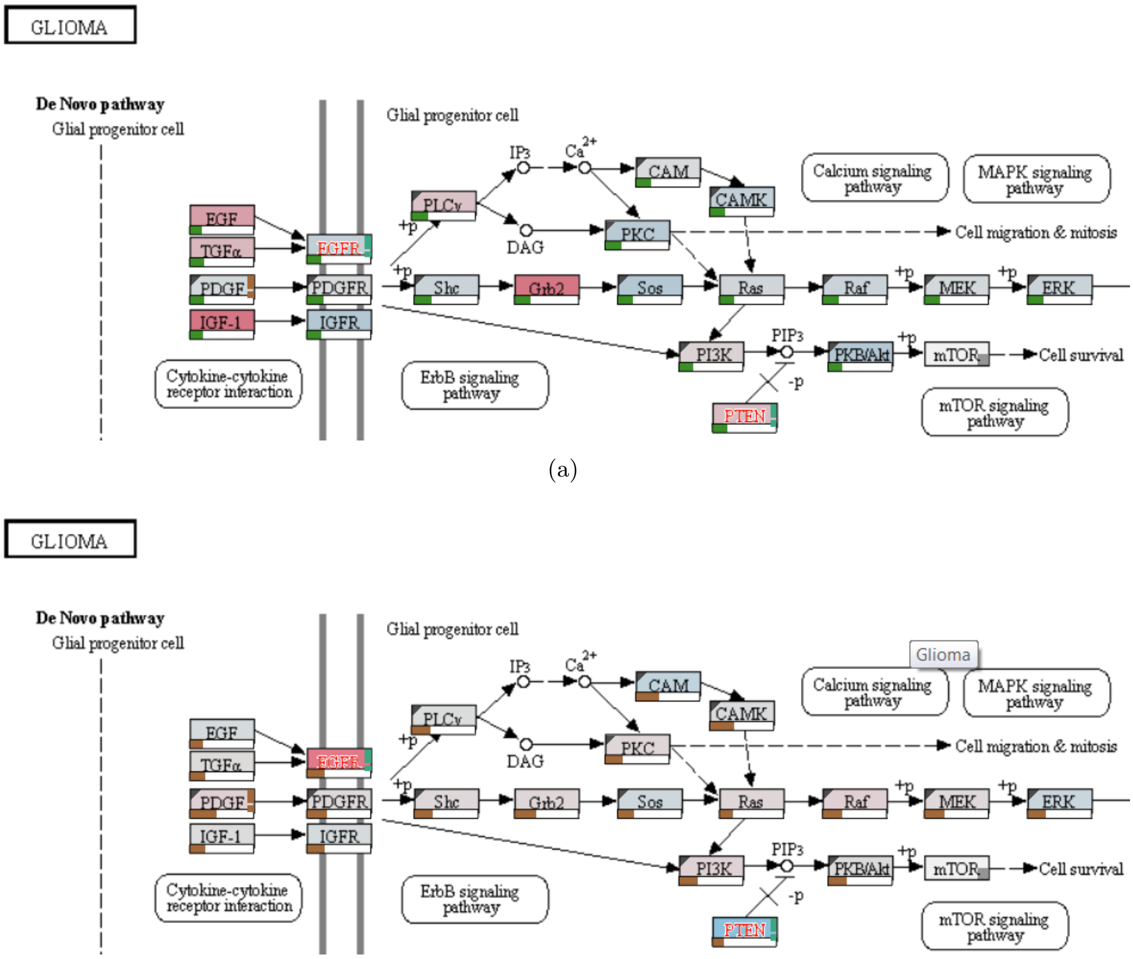
4.2 The EnRoute Visualization Technique

The overall goal of the EnRoute visualization technique is to jointly visualize experimental data and pathways in a way that addresses all five requirements discussed. We identify the topology-attribute coexistence requirement $(A \quad V)$ as the most critical requirement to address, as current techniques usually either support only topology-based or attribute-based tasks. Only small-multiples and direct on-node mapping are able to address requirement *A V*, however, both do not scale to many experiments (*A I*). Our solution to this problem makes use of an observation we made in discussions with our collaborators: They usually reason about and analyze experimental data associated with a single path at any given time in detail, while the rest of the network merely informs them about the context of this path. However, the path of interest might change through the course of the analysis. This temporal separation of high-level topology-based tasks and low-level attribute-based tasks allowed us to create a solution that meets all five requirements. In our approach, an analyst starts by investigating the pathway in its standard layout as taken from one of the major databases in the *pathway view* (*A IV*). She then selects a concrete path for which she wants to investigate experimental data. The chosen path is then shown in the *EnRoute view* in a linear form. Next to the nodes, we now have space available to concurrently show all mapped experimental data in a tabular format (*A I* and *A II*). Due to the linear arrangement of the nodes from top to bottom, it is possible to encode multi-mappings (*A III*) by giving them more vertical space. This dual-view concept is depicted in Figure 4.2. The *pathway view* provides the complete topological information as well as an overview of the experimental data, while the *enRoute view* contains the linear path and the experimental data visualization. The elaborate interplay of all these systems can solve the critical *Topology-Attribute Coexistence Requirement (A V)*: The topological information for the whole pathway is preserved in the pathway view, while the topological information for a path and the experimental data is shown in the EnRoute view. Overall, our approach of displaying linearized topology information next to attribute information was inspired by Pathline [126], which uses special encodings to show whole pathways in a linearized fashion. However, it suffers from its unconventional layout that can hinder understanding the graph topology. Additionally, it currently requires manual creation of the linearized pathways. Also, the association of nodes and attributes is less obvious. In the following sections, we discuss the components of our approach and their interplay in more detail.

4.2.1 Pathway View

In our approach, the pathway view serves two main purposes: First, it is the primary view for conducting topology based tasks, as the whole pathway map is shown. Second, it is used to interactively select the path to be shown in the accompanying EnRoute view along with associated experimental data. To facilitate identifying interesting paths, the pathway view shows abstract summaries of the mapped experimental datasets. In this section, we provide details about the design of the pathway view and its features.

Visualizing Experimental Data on Pathways. As discussed previously, directly mapping experimental data on pathway nodes using color-coding does not scale to more than a few experimental values, due to the small size of the nodes in the pathway maps. De-



(a)

(b)

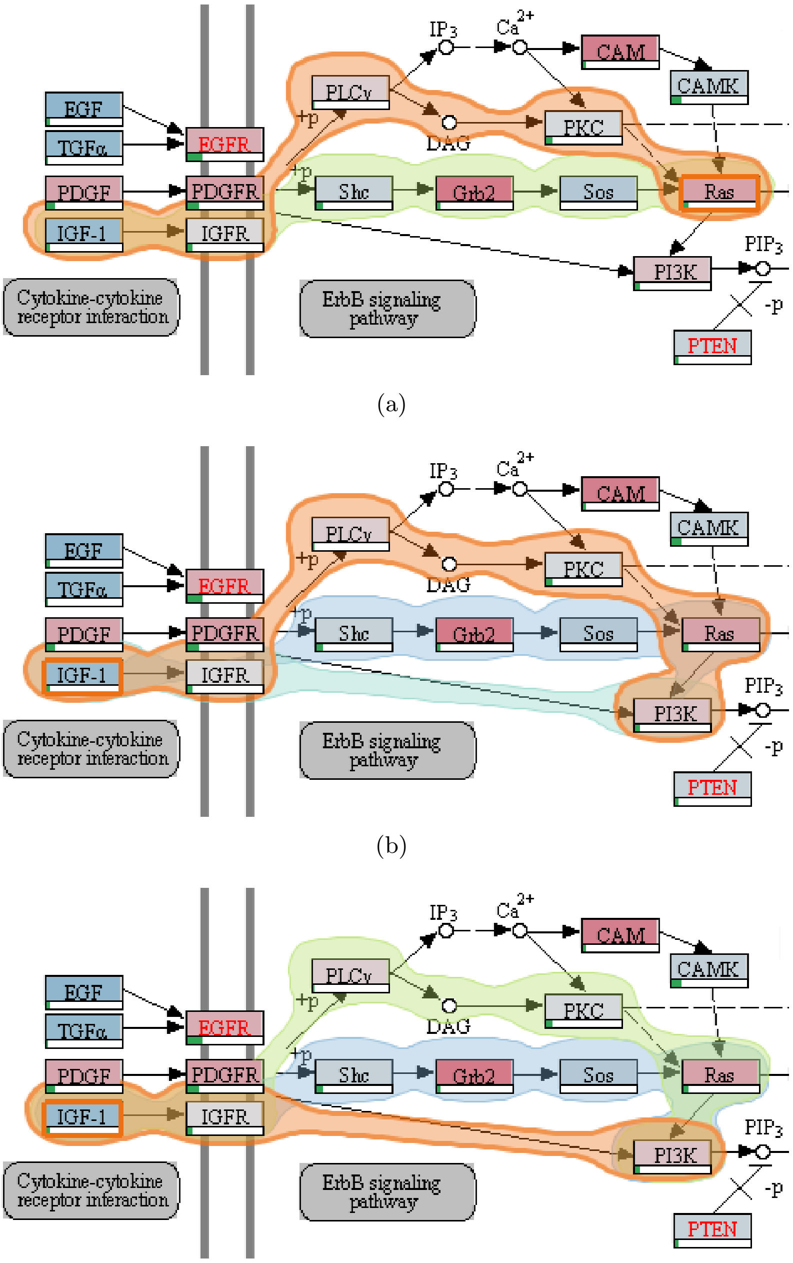
Figure 4.3: *Averages of mapped samples for different data types of the TCGA glioblastoma dataset are overlaid as color codes on nodes of the KEGG glioma pathway. Bars at the bottom of the nodes encode the standard deviation across the mapped samples. (a) mRNA data, using a blue-gray-red color map, where blue corresponds to under-, gray to regular, and red corresponds to overexpression. (b) Copy number data, also on a blue-gray-red color map, where blue corresponds to deletions, gray to a regular copy number, and red to increased copies of the gene.*

spite this limitation, direct on-node mapping is valuable in two scenarios: First, mapping summarized data allows analysts to gain an overview of the main trends in the pathway, which can be helpful when deciding which paths to choose for detailed investigation. Second, analysts want to investigate a condition, i.e., a group of samples, or even a single sample in its high-level topological context. This allows analysts to consider experimental data associated with nodes that are not in the currently extracted path. For this purpose, the pathway view can be configured to show only the mapping of selected samples.

To get a rough indicator of mapped experimental data, we calculate the average of all experimental sample values and multi-mappings, if applicable, and color-code the nodes accordingly. If multiple datasets are available, the analyst can choose which of them should be mapped in the *data configuration view* described in Section 4.2.3. Figure 4.3(a) shows the *Glioma* pathway with on-node mappings of mRNA data, while Figure 4.3(b) shows the same pathway overlaid with copy number data. For encoding numerical and ordinal data that diverges from a neutral center value, we avoid the red-black-green color map that is commonly used in the biological domain, but is problematic for many color blind people, and use a blue-gray-red color map instead. For example, for copy number data, gray is used for the *normal* status, while red indicates amplifications and blue indicates deletions. A two-color gray-red color map is used for non-diverging numerical data and nominal data with two categories, such as mutation status data. To indicate cases where experimental data is missing, we show a small rectangle in the lower left corner of the node, as can be seen, for example, in the *mTOR* node in the lower right part of Figure 4.3(b). Since the aggregation of all samples and possible multi-mappings into an average value hides all variation, we additionally provide the standard deviation encoded as a bar below each node. This indication of variance is very valuable for the overview task. High variation, which corresponds to longer bars, is an indicator for potentially interesting experimental data that is worth to be investigated in detail using the EnRoute view. In order to point out interesting nodes with respect to all present experimental datasets, rather than only for the currently mapped dataset, we also added exclamation marks to indicate that the standard deviation in a dataset is higher than a given threshold. The color of the exclamation mark encodes the dataset where the high standard deviation was observed. For example, the brown exclamation mark mapped onto the *PDGF* node in Figure 4.3(a) indicates high variance in the copy number dataset. If multiple datasets exhibit a high variance, we encode only the largest.

Selecting and Visualizing the Path As the EnRoute visualization technique builds upon the idea of providing experimental data along a path in the pathway, the user-driven selection of the path and its visualization in the pathway view is a critical step in the overall process. While selected paths are commonly indicated by highlighting edges in node-link diagrams, we opted for a slightly modified version of the *Bubble Sets* technique [34], as shown, for example, in Figure 4.1(a). Compared to simple edge highlighting, the contour-based Bubble Sets are more salient and can therefore be perceived faster. Furthermore, due to their curve-shaped outline, Bubble Sets can be easily discriminated from the mainly orthogonal structures in the pathway maps [81]. Also, the precise routing information of edges is often not available for pathway maps, and one would have to resort to drawing not-exactly matching edges. As the original Bubble Sets technique is meant for visualizing a set of items, we modified it for our purposes of highlighting a path. Thus, instead of allowing arbitrary branching between the nodes within a Bubble Set, we connect only those nodes that are connected by an edge in the pathway.

For path selection, analysts can choose between two approaches: the *iterative approach*



(a)

(b)

(c)

Figure 4.4: *Multiple differently colored Bubble Sets, each visualizing an alternative path between two user-selected nodes. (a) The analyst has selected IFG-1 as a start and Ras as end node. (b) The path is extended to also include the PI3K gene. (c) This results in a newly added alternative path, which is finally selected by the analyst.*

and the *start-stop approach*. In the iterative approach, the analyst can directly select a series of nodes that should be part of the path of interest. After selecting an initial node, the analyst can interactively extend the path in both directions by holding the *control* key, while clicking on connected nodes. In the start-stop approach, the analyst picks a start and an end node. We then calculate all paths between them using a variant of the Bellman-Ford algorithm [21]. All path alternatives are indicated with Bubble Sets, with the currently selected path highlighted in orange and using thicker outlines. The shortest path is selected by default, but analysts may switch to alternative paths by either using the mouse wheel or by directly clicking on the Bubble Sets of other paths. The path selection process is illustrated in Figure 4.4: In Figure 4.4(a) the system shows two possible paths between two genes that were selected using the start-stop approach. The analyst extends the currently selected path in Figure 4.4(b) and switches to an alternative path in Figure 4.4(c).

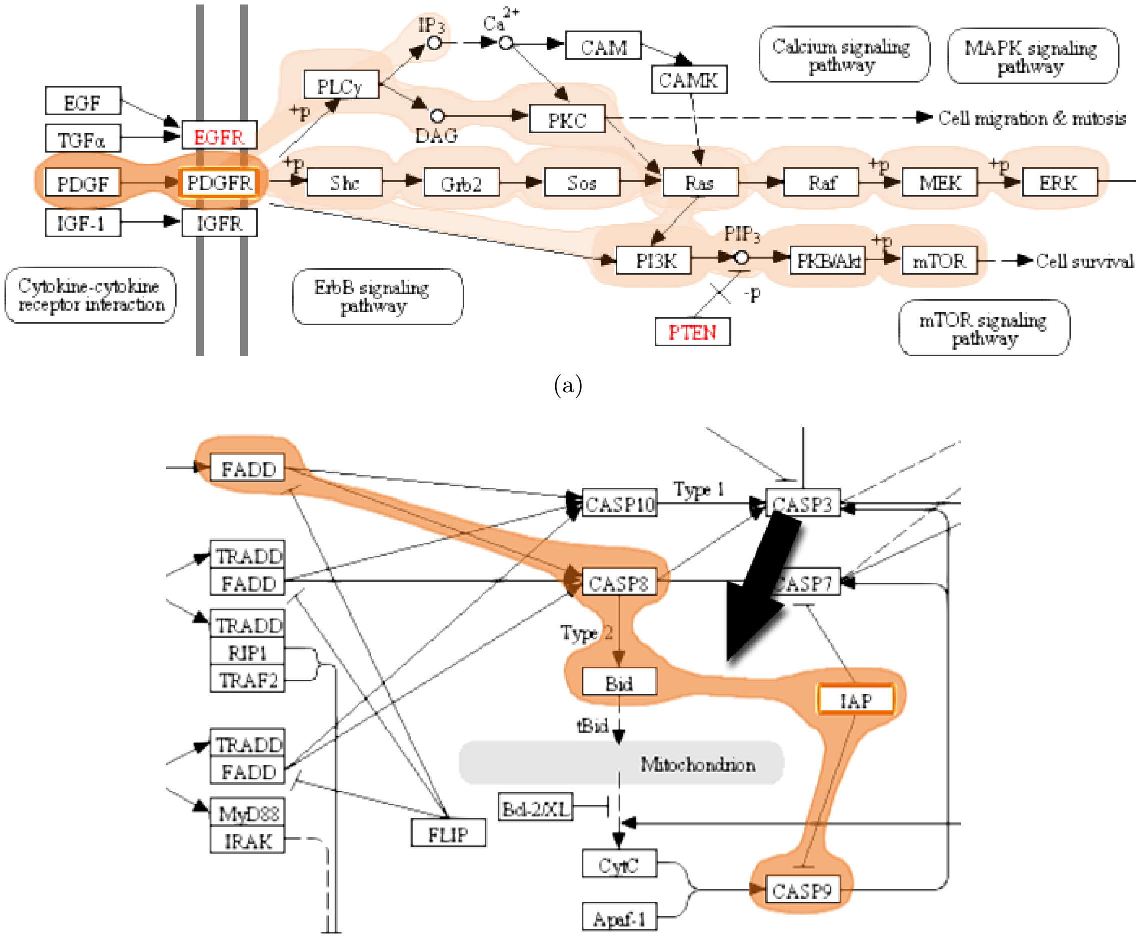
Both of our supported selection approaches have benefits and drawbacks. While the iterative approach allows analysts to determine paths that cover various kinds of topological structures, like, for instance, cycles, the start-stop approach makes it possible to investigate multiple alternative paths between nodes without the need to find and select the route by hand. Additionally, the start-stop approach is more efficient for selecting longer paths.

However, pathway maps are often very complex, and, sometimes, it is not obvious which other nodes can be reached from a given node. Therefore, we provide an interactive *preview mode* for selecting paths on user request. Starting at the end of the current selection, we highlight possible extensions. For example, in Figure 4.5(a), all possible path extensions for the end of the current selection at *PDGFR* are highlighted.

In some cases, the information of pathway maps is not complete or simply outdated. As a consequence, they may not correctly reflect the actual underlying biological process, especially not for all experimental conditions. Additionally, pathway databases can also contain errors that users are aware of. In order to cope with such incomplete or outdated pathway descriptions, we enable analysts to extend paths to arbitrary nodes within the pathway map. In this way, the analyst can consider missing edges during path selection. For example, in in Figure 4.5(b), a path is selected where the second to last edge does not exist.

4.2.2 EnRoute View

Once a path has been selected in the pathway view, it can be analyzed in detail in context of experimental data in the EnRoute view. The path is displayed in a linear, top-down layout, which is ideally suited to show rows of experimental data (*data rows*) right next to the associated nodes. As a node can have multiple mapped data rows, we adapt the spacing between nodes of the path so that all rows can be shown with a uniform height. Such multi-mappings or the occurrence of complex nodes (nodes that consist of multiple subnodes) in the path make it very hard, if not impossible, to determine which data row belongs to which node using their position alone. Therefore, we connect each node with corresponding data rows using ribbons, as shown in Figure 4.2. To make the association



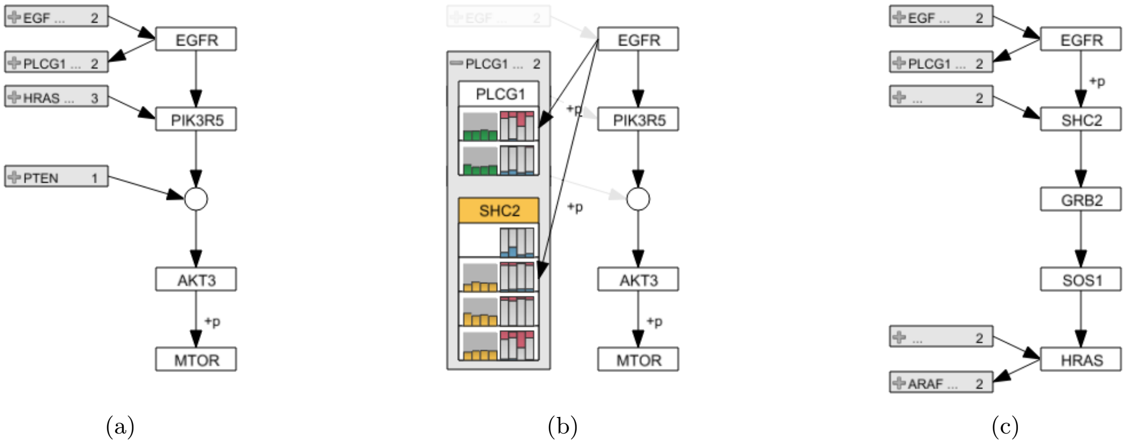
(a)

(b)

Figure 4.5: *Support for path selection. (a) All possible extensions of a path are shown using the preview mode. Here, all paths continuing after PDGFR are highlighted. (b) Considering edges that do not exist in the original pathway. Notice that no edge is shown between Bid and IAP in the original pathway map, but is introduced during path selection.*

between data rows and nodes even more obvious, we alternate the shade of gray in the data rows' backgrounds for each node. Figure 4.11(b) illustrates an example where these alternating shades of gray allow us to disambiguate the mappings of multiple subnodes of a complex node to corresponding data rows.

Following the divide-and-conquer visualization strategy [112], we group experimental data in the EnRoute view based on a homogeneity criterion. For example, experiments can be grouped by the species they belong to (homogeneity with respect to semantics), or a grouping can be obtained by clustering (homogeneity with respect to statistics). As illustrated in Figure 4.2, the groups are depicted as columns resulting in an overall tabular layout. We address the *heterogeneity requirement (A II)* by allowing the individual groups to originate from different datasets. However, all experiments within a group must originate from a single dataset.



(a) (b) (c)

Figure 4.6: *Path representation and branch switching in the EnRoute view. (a) The extracted path from the node EGFR to MTOR is shown top-down along with branches on the left. (b) Expanding the abstract node for leaving branches of EGFR reveals the individual branch nodes PLCG1 and SHC2, which show previews of associated experimental data. (c) By selecting SHC2, the associated branch replaces all path nodes below EGFR. All nodes of the branch are added up to the point where the branch is no longer unambiguous. In this case, HRAS represents the end point, as it has two leaving branches.*

Visualizing the Path In addition to showing the extracted path top-down in the EnRoute view, we also display branches that join or leave the path in order to preserve some of the topological information present in the pathway maps. We indicate a branch by showing its first node relative to the node where the branching occurs in the extracted path. In order to maintain a compact path representation, multiple branches that join or leave a single node of the path are abstracted into expandable nodes, one for all joining and one for all leaving branches, as shown in Figure 4.6(a). These abstract branch nodes indicate the number of branches they represent and also show labels for them, if sufficient space is available. Abstract branch nodes can be expanded on demand to reveal the individual branch nodes. When a node is expanded, all other branches are grayed out, and the expanded nodes are rendered on top of them. Expanded nodes show a preview for their associated experimental data, as illustrated in Figure 4.6(b). This facilitates the identification of potentially interesting branches. An analyst can interactively switch to a branch by selecting the corresponding branch node, as shown in Figure 4.6(c). A selected branch replaces all nodes in the extracted path above or below the node where the branching occurs, depending on whether it is a joining or leaving branch. All nodes of the branch are added to the path, until either a new branch or a dead end is reached. As the EnRoute visualization technique synchronizes all corresponding elements among its components, any changes to the path caused by branch switching are propagated back to the pathway view, thus keeping the highlights of the selected path up-to-date. Also, the synchronization of node highlights facilitates the association of branches shown in the EnRoute view with corresponding branches in the pathway maps.

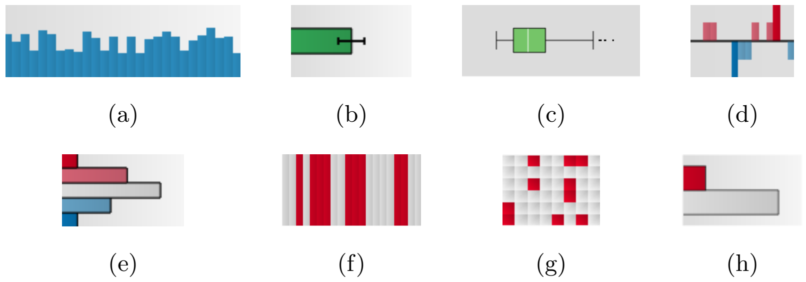


Figure 4.7: *Eight visual encodings for different types of experimental data. (a) One vertical bar is shown for each numerical data value. (b) A summary representation of a group of numerical data values that consists of one horizontal bar with whiskers. (c) A box plot as summary representation of a group of numerical data values. (d) Copy number values are redundantly encoded using color and length. Red bars pointing upwards indicate an increased number of copies, whereas reduced copy numbers are shown as blue bars pointing downwards. (e) A group of copy number values are abstracted into a histogram. (f) Mutation status data encoded as colored bars. Red bars indicate samples with a mutated gene. (g) More compact matrix visualization of mutation status data. (h) Histogram summarizes mutation status data by displaying the number of samples with and without mutations.*

Visualizing Experimental Data Being able to display large amounts of heterogeneous experimental data is an integral part of the EnRoute visualization technique (see requirements *A I* and *A II*). EnRoute supports the visualization of quantitative, ordinal, and binary categorical data. As previously mentioned, we organize experimental data in rows and columns. Each row shows data that maps to a certain node in the path, and columns group the data by a homogeneity criterion. Different groups may also have overlapping experiments. The captions of the individual groups are displayed at the top and at the bottom of the corresponding columns. Their background color indicates the dataset they belong to. For example, in Figure 4.1(b), the background of groups showing mRNA expression data is turquoise, whereas the background of copy number data groups is blue, and the background for mutation data is light violet.

In molecular biology, heatmaps are the standard way to visualize quantitative and ordinal data. However, it is well known that hue or value are inferior to other encodings with respect to communicating changes in the data. For both quantitative and ordinal data, encodings in position are a better choice and for quantitative data, length encodings are also superior [120]. Meyer et al. [126] showed that a mirroring effect in expression data was much more apparent when it was visualized using line plots compared to when using heat maps. Heatmaps or any other pixel-based visualization techniques are superior in terms of space efficiency and therefore scalability. EnRoute, however, only requires the visualization to be scalable with respect to experiments, since the number of genes is typically small, as it is limited by the number of nodes in the path. Therefore, we prefer bar charts over heat maps for the representation of quantitative data as well as for ordinal data.

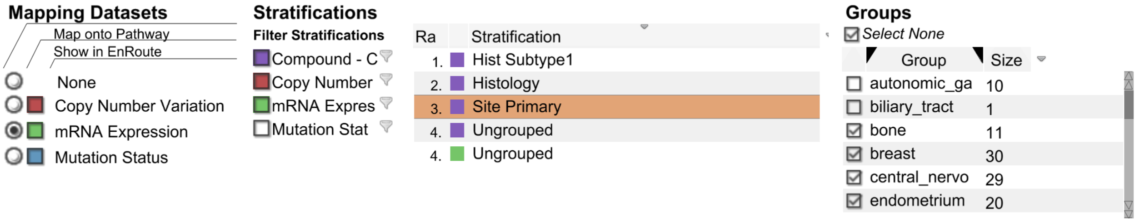
In the bar charts used for quantitative data, each bar represents one value of a single experiment, as shown in Figure 4.7(a). In order to make the borders of adjacent bars apparent without having to waste space for drawing outlines, we color the bars using a gradient from left to right. As shown in Figure 4.1(b), tooltips are used to show the numerical values of the underlying data. In some cases, it might be desirable to see an abstract and more compact visualization of a group of quantitative data. For this purpose, we use one horizontally aligned bar that represents the mean value of a group together with whiskers encoding the standard deviation, as shown in Figure 4.7(b). In a later iteration of EnRoute, we replaced this representation with box plots [123] as shown in Figure 4.7(c), which provide more information about the underlying value distribution: The box ranges from the first ($Q1$) to the third quartile ($Q3$) of the distribution, representing the interquartile range (IQR). The white line within the box indicates the median value. The left whisker shows the smallest value within $Q1-1.5\cdot IQR$ and the right whisker the largest value within $Q3+1.5\cdot IQR$. More extreme outliers below and above are shown as individual dots. The exact values of median, quartiles etc. are accessible via tooltips. In contrast to the detailed representations, where the width adapts to the number of experiments in the group and available display space, the width of abstract group representations is fixed. This constant width and the horizontal alignment of the abstract bars and box plots, respectively, allows analysts to compare values of the same group across rows along the path more easily. However, for tasks that require comparisons across multiple groups, the detailed representation with vertical bars are preferable.

As copy number data commonly occurs either in ordinal or quantitative form, we use an optimized encoding that can deal with both of them. Ordinal copy number data is often categorized into *high* and *low* increase of gene copies, a *normal* copy number, *deletion on one allele*, and *deletion on both alleles*. As shown in Figure 4.7(d), our encoding of this data redundantly uses the length, color, and orientation of bars. For highly increased copy numbers, we show long, dark red bars pointing upwards from a base line. For low increases, we use shorter, light red bars. Similarly, deletions are represented by dark and light blue bars pointing downwards. No bar is shown for normal copy numbers. The same encoding can be used for quantitative copy number data. The higher the increase in copies, the longer and darker the red bar is. The same concept applies to deletions. Just like for general quantitative data, we also employ an abstract representation for groups of copy number values. As shown in Figure 4.7(e), we use a horizontal histogram, which makes use of the same color coding as the detailed copy number representation.

For binary categorical data, such as data on whether a gene is mutated or not, we provide two alternative representations: In the first representation shown in Figure 4.7(f), we use equally sized colored bars, where each bar corresponds to one sample value. For the mutation status example, we color samples that are mutated in red, while non-mutated samples are shown in the background color. The same color-coding is employed in our second representation, which displays each sample value as a cell in a matrix visualization, as shown in Figure 4.7(g). This representation is significantly more space-efficient, which is important for mutation data, since mutated genes are scarce in many datasets. However, the matrix representation deviates from the convention of placing samples side-by-side, which makes tracing of individual samples along the path more difficult. Similarly to the abstract summary representation of copy number data, we use a histogram to summarize mutation data, as illustrated in Figure 4.7(h).

The previously mentioned data previews that are shown on demand for branch nodes use an encoding similar to the abstract data representations, as can be seen in Figure 4.6(b). For each group of mRNA data, one bar indicating the group's mean value is drawn. For copy number and mutation data, we show one stacked histogram per group.

The EnRoute visualization technique makes use of synchronized highlighting of corresponding elements across all its components, but also within all components. The latter case is especially useful in the experimental data display. By highlighting a set of experiments in one group, we allow analysts to identify these experiments in other groups, even for different data types. For example, in Figure 4.10(b), all cell lines with an increased copy number are highlighted, which allows analysts to relate the increase in copy number with mRNA expression. As evident in this figure, scattered selections make it difficult to quantify the number of selected experiments. To alleviate this problem, we add tooltips to the groups' captions showing the total number of experiments and the number of currently selected experiments of each group.



Mapping Datasets

Map onto Pathway
Show in EnRoute

○ None
○ 🟥 Copy Number Variation
● 🟩 mRNA Expression
○ 🟦 Mutation Status

Stratifications

Filter Stratifications

🟪 Compound - C
🟥 Copy Number
🟩 mRNA Expres
□ Mutation Stat

Ra		Stratification
1.	🟪	Hist Subtype1
2.	🟪	Histology
3.	🟪	Site Primary
4.	🟪	Ungrouped
4.	🟩	Ungrouped

Groups

☑ *Select None*

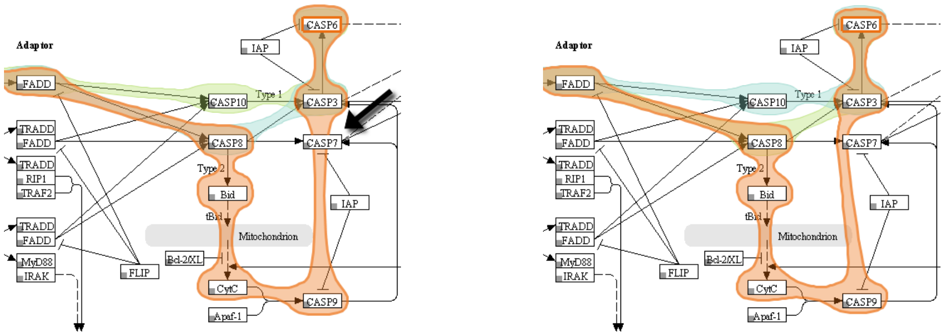
	Group	Size
□	autonomic_ga	10
□	biliary_tract	1
☑	bone	11
☑	breast	30
☑	central_nervo	29
☑	endometrium	20

Figure 4.8: *The data configuration view. It is used to specify the datasets and stratifications of samples shown in the pathway and EnRoute view. Available datasets are displayed on the left, stratifications at the center. The individual groups of a selected stratification are listed on the right.*

4.2.3 Data Configuration View

Up to this point, we have assumed that decisions on which datasets and which groupings of the datasets to show are already made. However, given a large set of datasets and a variety of alternative groupings to choose from, this presumably easy task is, in fact, not trivial. To support analysts in the task of selecting datasets and groupings and assigning them to the pathway view and the EnRoute view, we provide a dedicated data configuration view.

As illustrated in Figure 4.8, the view consists of three major parts for the selection of datasets, groupings, and individual groups. On the far left, all datasets eligible for display with the EnRoute visualization technique are listed. In the shown example, these are copy number variation, mRNA expression, and mutation status datasets from CCLE. Analysts may choose which dataset to map onto the pathway nodes using the radio buttons, and which datasets to show in the EnRoute view using colored check boxes. All available stratifications of samples are displayed in the list at the center of the configuration view. A stratification determines into which groups the samples are partitioned. By selecting a stratification, its grouping of samples is applied to all datasets displayed in the EnRoute view. In Figure 4.8, the stratification "Site Primary" is selected, which specifies groups according to the origin of the samples, such as "bone" or "breast". All groups of a selected stratification are listed on the far right, and each group can be added or removed from the EnRoute view by checking or unchecking them.



(a) (b)

Figure 4.9: *Re-routing of Bubble Sets for path highlights. (a) As the pathway description does not provide information about the routes used to connect the nodes, a collision of the path leading from CASP9 to CASP3 with the unrelated node CASP7 occurs. (b) To avoid this, we redirect the path around the node.*

4.3 Implementation and Scalability

EnRoute is implemented as a view plugin for the Caleydo framework. Our current implementation builds upon the well-established KEGG and Wikipathways databases, which provide information about pathways as image data in combination with an XML-based descriptions of the graph. The description includes the topological information as well as the position and the size of nodes. We use this information to render the various augmentations described on top of the pathways.

In order to highlight selected paths, we make use of a free implementation of the Bubble Sets technique[1] that we integrated into the Caleydo framework. Since pathway databases

[1] http://github.com/JosuaKrause/Bubble-Sets

such as KEGG do not provide information about how edges are routed in the images, directly connecting nodes in a path using straight edges can lead to overlaps between the Bubbe Sets overlay and nodes that are not part of the path. In Figure 4.9(a), for example, the edge highlight between node *CASP9* and *CASP3* collides with the representation of node *CASP7*, which is not a member of the selected path. In such cases, we route edges around nodes to avoid overlaps. Figure 4.9(b) shows the Bubble Set with a refined route from *CASP9* to *CASP3*.

The Bubble Sets algorithm renders the overlays semi-transparent on the top of an existing base representation. However, if the base representation uses color-coding, overlaying colored Bubble Sets can lead to a wrong interpretation of the encoded information, as the colors of the Bubble Sets interfere with the color of the augmented content. Additionally, blending decreases the legibility of the original content. To avoid this problem, we cut the area of nodes out of the Bubble Sets.

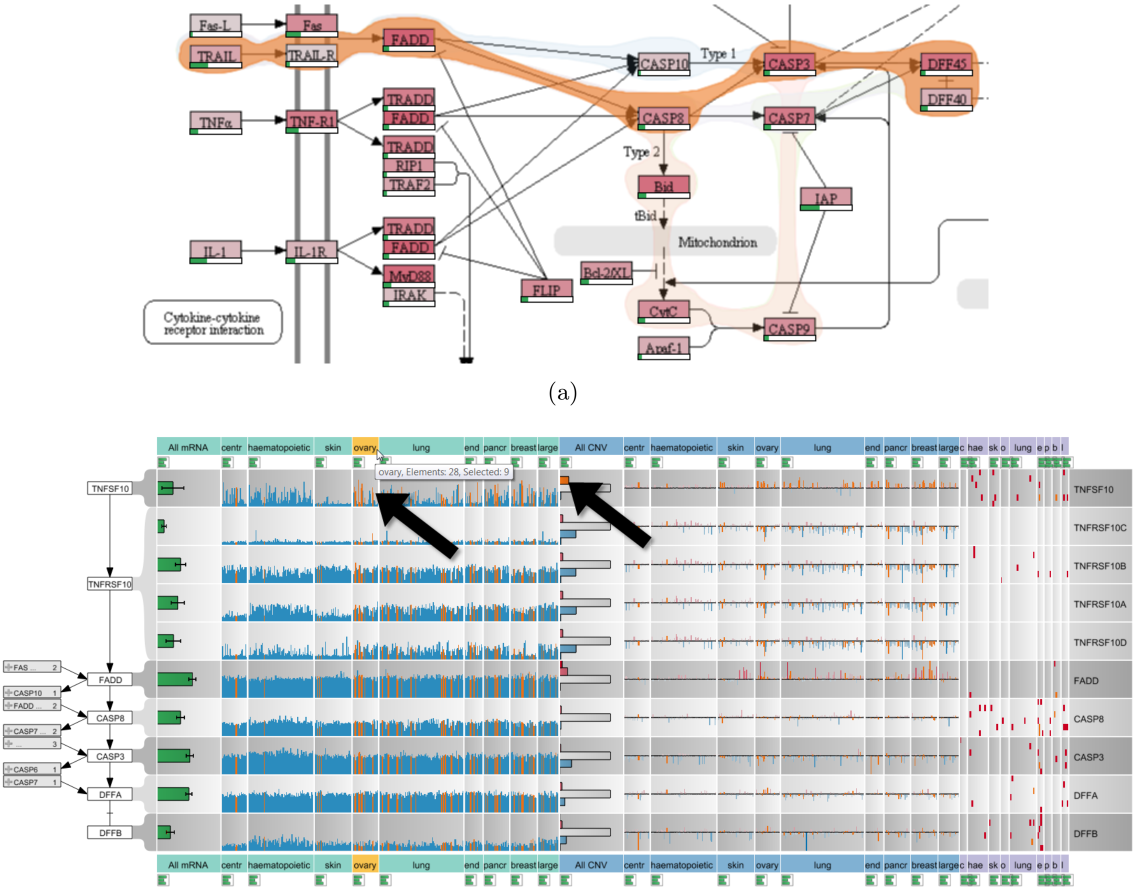
EnRoute scales to hundreds of experiments covering extensive TCGA and CCLE datasets. For example, Figure 4.10 shows public mRNA and copy number datasets that contain data values for about 350 samples and ∼20,000 genes. In theory, a width of one pixel is enough to encode one sample. Thus, the maximum number of displayable experiments is very much dependent on display resolution. However, larger amounts of experiments can be coped with by showing summary representations or selecting only those groups of data for display that are in the current focus of the analysis. If the length of the path exceeds the available screen space, we use scroll-bars to navigate to the off-screen parts. We found this to be reasonable due to the linear nature of the exploration process along the path.

4.4 Case Studies

We developed the EnRoute visualization technique in close collaboration with an expert from the Medical University of Graz. To evaluate the utility of the EnRoute visualization technique, we conducted two case studies using different datasets together with this expert. The first dataset is taken from the Broad-Novartis Cancer Cell Line Encyclopedia, which contains the genetic and pharmacologic characterization of a large panel of human cancer cell lines. The second is a gene expression dataset from a model of hepatocellular carcinoma collected at the Medical University of Graz.

4.4.1 Apoptosis Regulation in Cancer Cell Lines

The first case study investigates the difference in regulation of the apoptosis pathway in different human tumor cell lines from various organs. Apoptosis is the programmed death of a cell due to internal damage or as a consequence of external stimuli and involves a signaling cascade, which is not mediated by phosphorylation, but by targeted degradation of proteins through enzymes called caspases. To initiate apoptosis in a cell, a ligand called



(a)

(b)

Figure 4.10: *Analysis of CCLE experimental data in context of apoptosis. (a) Different paths between the nodes TRAIL and DEF40 are highlighted in the Apoptosis pathway map. The orange Bubble Set shows the chosen path. (b) The selected path is shown in context of associated mRNA expression, copy number, and mutation status data from the CCLE dataset with about 350 samples each. As indicated by the arrows, highlighting all samples with an increased copy number of TNFSF10 reveals the correlation with increased expression levels in the ovary cell lines. Notice that, while the labels shown in the EnRoute view are not identical to the labels in the pathway view, the labels are homologous and deviate, since pathway databases use one of many aliases for genes or proteins.*

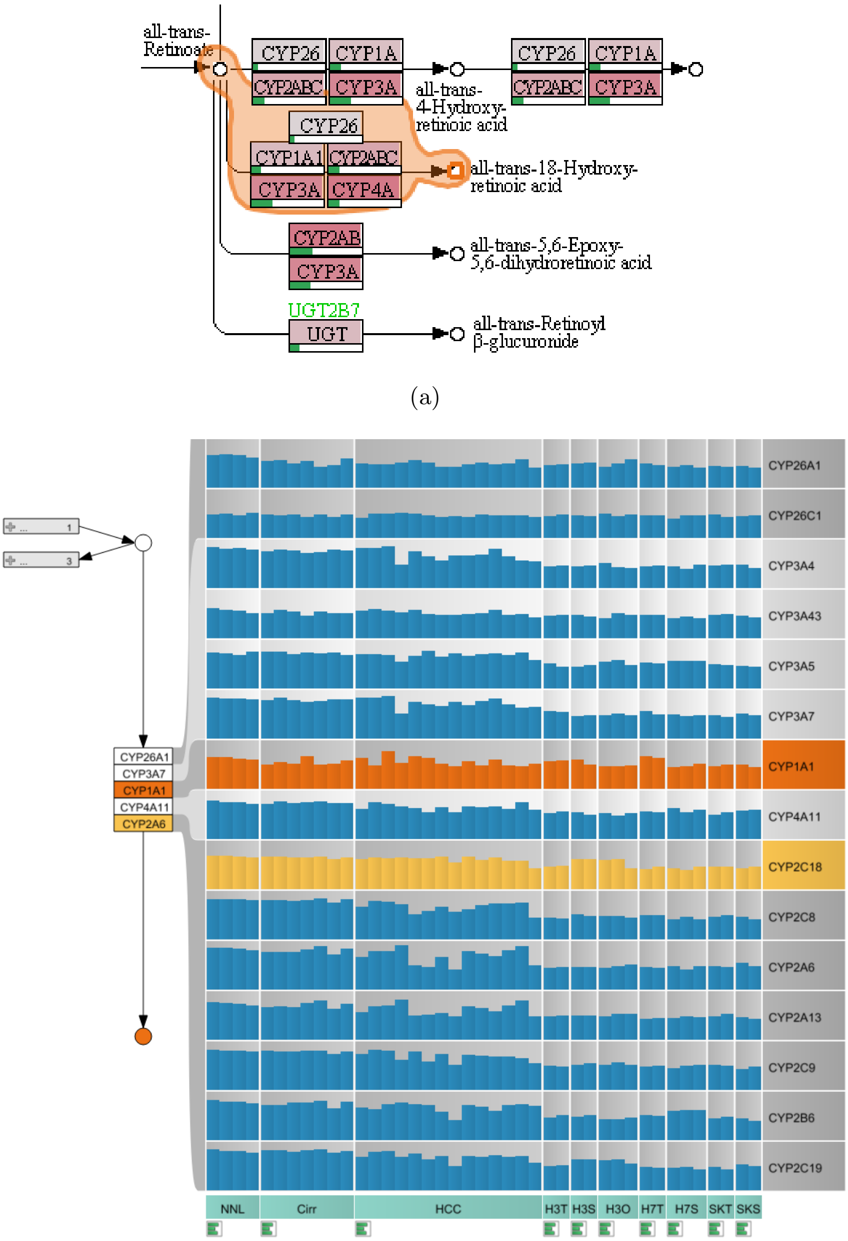
TRAIL or *TNFa* binds to receptors on the outer cell surface. This leads to activation of receptor associated death domain containing proteins, which activate the first of a cascade of caspases, which, in turn, eventually leads to the cleavage of proteins in the cell subsequently triggering apoptosis. During apoptosis the cell shrinks in volume, exhibits nuclear fragmentation, chromosomal DNA fragmentation, and release of Cytochrome c from the mitochondria, which eventually leads to cell death.

As first step of the analysis, the *Apoptosis* pathway of the KEGG database is loaded into the pathway view. The researcher selects *TRAIL* as the starting point and *DEF40* as the endpoint of the signaling cascade to be investigated. As shown in Figure 4.10(a), the system automatically highlights all possible paths from *TRAIL* to *DEF40* and allows to switch between these alternative paths. The selected path is extracted and displayed in a linear layout in the EnRoute view, which shows detailed CCLE mRNA expression, copy number variation, and mutation status data for the genes that map to the nodes of the path, as can be seen in Figure 4.10(b). In this example, the grouping of the experimental data reflects the source organ of origin of the cell lines. The analyst can now easily identify differences in the copy number variation, gene expression, and mutation status for these groups and relate them immediately to the stages of the linearized path, which would not be possible using other techniques. As indicated by the upward pointing bars for the copy number data, there is a clear amplification of *TNFSF10* (an alias for *TRAIL*) for cell lines originating from ovary, lung, or breast tissue. By selecting all samples with an increased copy number of *TNFSF10* in the histogram, the system also highlights these cell lines in the mRNA expression plots, as indicated by the arrows in Figure 4.10(b). The analyst can now confirm that ovary cell lines with a higher copy number of this gene also show a higher expression of *TNFSF10* in the mRNA expression plots. Based on this data, one could interpret that these cell lines should be prone to apoptosis, as they have high amounts of the cytokine triggering the cell death. However, given the nature of these cell lines, which are derived from human tumors and are by definition resistant to apoptosis, this conclusion would be misleading. An explanation for this seemingly strange behavior can be found by looking at the copy numbers of the genes *TNFRSF10A* to *TNFRSF10D*, which are receptors for *TNFSF10* cytokine, further downstream in the path. Here, it quickly becomes evident that these genes exhibit a frequent loss of copies, which is easy to see when looking at the many blue bars pointing downwards for these genes. This explains how these cells can still form a tumor, as the loss of the receptor makes the cells immune to the stimulus by the *TNFSF10* cytokine, thus preventing apoptosis initiation.

In summary, EnRoute enabled the analyst to study copy number variation, gene expression, and mutation status of a large number of samples in parallel in a clearly visualized and linearized sub-path of the complex network initiating apoptosis. The researcher stated that the linearization of the signaling cascade and the clear association of the other data entities to the individual steps of this cascade are very intuitive and greatly facilitated the interpretation of the data and the deduction of a biological interpretation from the dataset.

4.4.2 HCC Xenograft Models

The second use case demonstrates the ability of the EnRoute path extraction feature to aid in a common problem when interpreting pathway maps. In biological reaction systems, enzymatic reactions are often carried out not by a single gene or protein, but by a family of proteins encoded by several genes. Additionally, enzymatic reactions are often not specific to a single protein family, but can be substituted by other enzymes, which, however,



(a)

(b)

Figure 4.11: *Resolving multi-mappings in the KEGG pathway map Retinol Metabolism. (a) A path that contains a complex node with many multi-mappings is selected. (b) This path is shown in the EnRoute view, together with gene expression data for normal liver (NNL), cirrhosis (Cirr), hepatocellular carcinoma (HCC), and three cell line models, Hep3B, HUH7, and SK-Hep in the conditions Hep3B tissue culture (H3T), Hep3B subcutaneous graft (H3S), Hep3B orthotopic graft (H3O), HUH7 tissue culture (H7T), HUH7 subcutaneous graft (H7S), SK-Hep tissue culture (SKT), and SK-Hep subcutaneous graft (SKS). CYP2C18 (yellow) is highly expressed in the Hep3B model and CYP1A1 (orange) is upregulated in HUH7.*

often work with different efficiency. This biological diversity leads to functional nodes in pathways that contain a multitude of genes with historically designated gene names often not indicative of the real function of these genes. It is extremely difficult for the researcher to keep in mind which genes are behind a single node of a pathway and thus understand the mapped biological data. Cross-referencing of expression levels to genes and functional nodes can thus only be achieved by resolving the multi-mapping in a node to gene names, which can then be mapped to the individual gene expression levels. In EnRoute, the researcher can select nodes upstream and downstream of the enzymatic reaction, which is then resolved in a linearized representation of all genes involved. An example is the conversion of *all-trans-Retinoate* to *all-trans-18-Hydroxy retinoic acid*, which is a reaction contained in the *Retinol (Vitamin A) metabolism* pathway shown in Figure 4.11(a). The single node that represents this reaction contains 16 different proteins or genes that are involved. As depicted in Figure 4.11(b), EnRoute resolves this node into a convenient map of genes, thus making it possible to map the gene expression of each individual gene in all experimental conditions of this experiment. The dataset mapped onto the Retinol pathway was generated by gene expression profiling of normal and cirrhotic human liver, hepatocellular carcinoma, and grafts of three human tumor cell lines (Hep3B, HUH7, and SK-Hep) into immunodeficient mice. Expression was measured in liver samples from patients and in samples from tumor cells grown in culture (TC), as subcutaneous grafts (SC), or as orthotopic grafts (Ortho) in the liver of experimental animals. The task was to find out how well the cell line models correlate to the human disease, with special focus on drug metabolism and oxidative stress. The response of tumors to anticancer treatments is closely linked to the activity of *cytochrome p450* enzymes, which metabolize drugs and mediate oxidative stress. When studying the aforementioned conversion of retinoate to retinoic acid by cytochrome p450 enzymes, it can now be seen that the node *CYP2A6* actually contains members of the whole *CYP2* family of genes and that the expression of these genes is uniform in normal liver and cirrhosis, but very variable in HCC, dividing these cases into a low expressing and a high expressing group. It is immediately visible that the xenograft models are only representative of the low *CYP2* expressing group of HCCs. Decomposition of the complex nodes allows the identification and investigation of the individual expression patterns of genes contained in the complex node. It becomes obvious that the expression of *CYP2C18* (highlighted in yellow), contained in the node labeled *CYP2A6*, is higher in the Hep3B model than in the other cell lines.

Additionally, it can be detected that *CYP1A1* is highly expressed in HUH7 tissue culture cells. All this information was not visible to the researcher using conventional on-node mapping approaches and was successfully visualized using EnRoute. Node decomposition is an integral feature of the EnRoute path extraction and thus makes the association of many mapped nodes and their corresponding experimental data readily available to the researcher.

4.5 Summary

In this chapter, we introduced EnRoute, a visualization technique for the analysis of large amounts of heterogeneous experimental data in the context of pathways. While we tackle several domain-specific issues with EnRoute, we also address the generic problem of the joint analysis of graph topology and heterogeneous node attributes in a scalable way. The key concept behind this is sub-setting. In particular, we allow users to select a path in the graph for which detailed attributes shall be shown. We exploit the sequential nature of paths and display it top-down, which enables us to conveniently visualize large amounts of attributes in a tabular display right next to the path. The practical applicability of this approach is confirmed in case studies from Molecular Biology, where analysts were able to detect correlations in mRNA and copy number data along a path and to resolve data associations in nodes with excessive multi-mappings. In summary, we conclude that flexible sub-setting to paths represents an effective and scalable way to investigate attributes in a graph.

Up to now, we considered the analysis of only single pathway maps in EnRoute. Although EnRoute scales well with the number of node attributes, it requires the size of the graph to be rather small in order to enable users to select paths in the node-link representation. Thus, to make the EnRoute concept work with larger graphs, intelligent sub-setting is required. In the next Chapter, we introduce a focus+context technique that enables the analysis of a larger biological network by simultaneously considering multiple pathway maps. Combining of this technique with EnRoute allows for a scalable cross-pathway analysis of attributes.

5 Connecting Graph Partitions

Contents

The visualization of large graphs is challenging. A key aspect for making the visual analysis of such graphs possible is finding ways to reduce the data that needs to be displayed to a manageable amount. The partitioning of biological networks into pathways can be regarded as meaningful way of data reduction, as, in many cases, only information about a single biological process, which is captured by a pathway, is required. However, as discussed in Section 2.3.2, the consideration of only single pathways is not enough in drug discovery, as drugs can potentially affect multiple biological processes. Therefore, a more holistic analysis of the biological network that involves multiple pathways is required.

In this chapter, we introduce **Entourage**, a bottom-up focus+context technique for the visual analysis of large partitioned graphs, in particular, biological networks. Entourage exploits the existing partitioning of biological networks into pathways and combines it with intelligent sub-setting: As shown in Figure 5.1, it displays one pathway that is in focus of the analysis at full scale, while only showing contextually relevant subsets in the form of paths for other pathways. The technique allows users quickly change what is displayed in focus and context. To indicate relationships between pathways, stubs are attached to shared nodes, which can be expanded to full links connecting the nodes on demand. Entourage was developed to support inter-pathway analysis in drug discovery, which also requires the investigation of experimental data in context of pathways. Therefore, we combined Entourage with the EnRoute visualization technique introduced in the previous chapter. This combination overcomes the limitation of EnRoute to single pathways and allows for inter-pathway attribute visualization.

In the remainder of this chapter, we analyze the requirements of the visual analysis of multiple pathways for drug discovery and follow up by describing the Entourage visualization technique and also its combination with EnRoute. Finally, we demonstrate the use of Entourage and the combination with EnRoute in case studies from drug discovery.

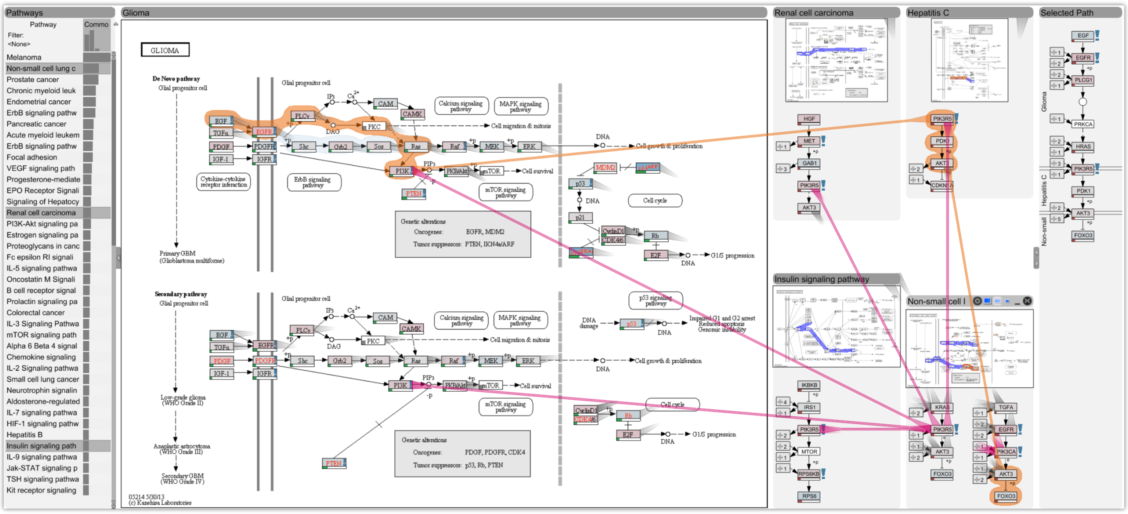


Figure 5.1: *Entourage showing the Glioma pathway in detail and contextual information of multiple related pathways.*

5.1 Requirement Analysis

All of the domain goals described in Section 2.3.2 have pathway analysis as pivotal part of them. Together with our collaborators, we elicited the analysis tasks to be carried out and derived requirements for the Entourage visualization technique accordingly. We classify the detailed requirements into two categories, the *Pathway Interconnectivity* requirements and the *Pathway-Experimental Data Linking* requirements.

The *Pathway Interconnectivity* requirements deal with finding pathways related to each other and analyzing the relationships between pathways. The requirements are:

B I: Show related pathways. While an initial pathway is typically known for the stated goals, it is important to easily find related pathways, as cross-talk and other interdependencies are more likely between highly related pathways. We consider two pathways as related when they either share one or multiple genes (nodes), have an edge crossing from one pathway to the other, or if one is contained or referenced in the other.

B II: Show high-level relationships of pathways. When related pathways are found, it is also important to see how they are related. For example, it is interesting to see whether the same sub-process is contained in both pathways, or whether one pathway is contained or referenced within the other pathway.

B III: Identify a gene's role in multiple pathways. Identifying the role of a gene in multiple pathways is important to determine the different cellular processes that a gene is involved in. This represents valuable knowledge, for example, when assessing the suitability of a gene as a drug target.

B IV: Find path intersections. A change in gene activity, e.g., caused through a mutation or modulation by a drug, can influence the activity of subsequent genes in a path. Consequently, it is important to not only look for the role of the originally altered gene, but also to explore the role of genes that might be influenced by it. This can be done by exploring the relationships of the nodes downstream of the original gene, i.e., by finding pathways that intersect the path of a changed gene.

The *Pathway-Experimental Data Linking* requirements are equally important to achieve the goals stated above. Only experimental data can give insight into the effects a change, either naturally occurring or introduced by drug treatment, has on the whole cell or organism. In the previous chapter, we introduced the EnRoute visualization technique, which addresses several requirements involved in linking pathways and genomic experimental data. These requirements also hold for the analysis in drug discovery. However, with the addition of non-genomic experimental data, the following requirement is not yet covered by EnRoute:

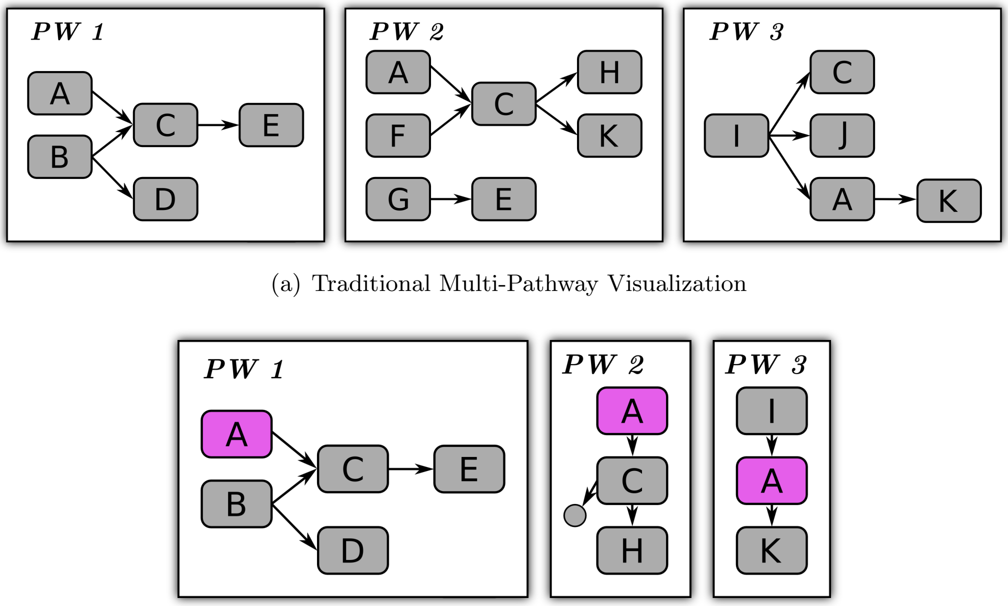
B V: Show relationships between cell line responses to drug treatment and genomic data. Our collaborators would like to find out why certain cell lines react differently to the same compound treatment. These differences typically have genetic causes. Joint analysis of all the aforementioned data types can reveal the reasons for such differential behavior. This knowledge could in turn be used for targeted therapy, i.e., the identification of patients that are most likely to respond to a specific drug treatment.

5.2 The Entourage Visualization Technique

The main goal of Entourage is to enable the *Pathway Interconnectivity* tasks, which requires a joint analysis of multiple pathways. However, current pathway visualization techniques (see Chapter 3) lack the flexibility required for exploring interdependencies across pathway boundaries. The main problem one must address is scale. Current approaches either cannot show individual nodes sufficiently large or cannot show relationships between multiple pathways. Here, we introduce the concept of contextual subsets to remedy this issue.

Figure 5.2 illustrates the difference between a traditional multiple pathway analysis and the contextual subsets method. The traditional approach depicted in Figure 5.2(a) shows all nodes for all pathways. The pathways in this example share several nodes. We refer to such shared nodes as *portals*, as they allow us to jump from one pathway to another. Figure 5.2(b) illustrates the same set of pathways using the contextual subsets technique. Instead of showing all pathways in detail, we distinguish between *focus pathways*, shown at full scale (PW 1 in Figure 5.2(b)), and *context pathways*, which are smaller and show only a contextually relevant subset of their graph (PW 2 and 3 in Figure 5.2(b)). What is contextually relevant is driven by a user-selected *focus node* (A, purple in Figure 5.2(b)). The context pathways only show limited subsets of their network that also contain node A. In the example shown in Figure 5.2(b), the most important path is shown for each occurrence of the focus node, while other branches are only indicated, as is evident in Pathway 2.

This simple, yet effective principle makes use of the observation that analysts want to see all the details of one pathway map (their focus pathway), but do not need to see all the intricate details of other potentially involved processes (the context pathways) to judge interdependencies to their focus pathway. Entourage utilizes the observation that the focus of attention shifts serially to optimize the visible content to what is currently relevant to the analyst. The challenge we have to address is the continuous change of attention, the adaption of the analysis focus in the process of an exploration. Entourage employs a series of visual encodings and interaction techniques to make these changes as convenient and transparent as possible.



(a) Traditional Multi-Pathway Visualization

(b) Pathways as Contextual Subsets

Figure 5.2: *Comparison of a traditional multi-pathway approach and contextual subsets. (a) All pathways are shown at the same scale competing for display space. (b) The contextual subset technique showing one* ***focus pathway*** *(PW 1) and two* ***context pathways*** *(PW 2, PW 3). The context pathways only show paths that contain the* ***focus node*** *A.*

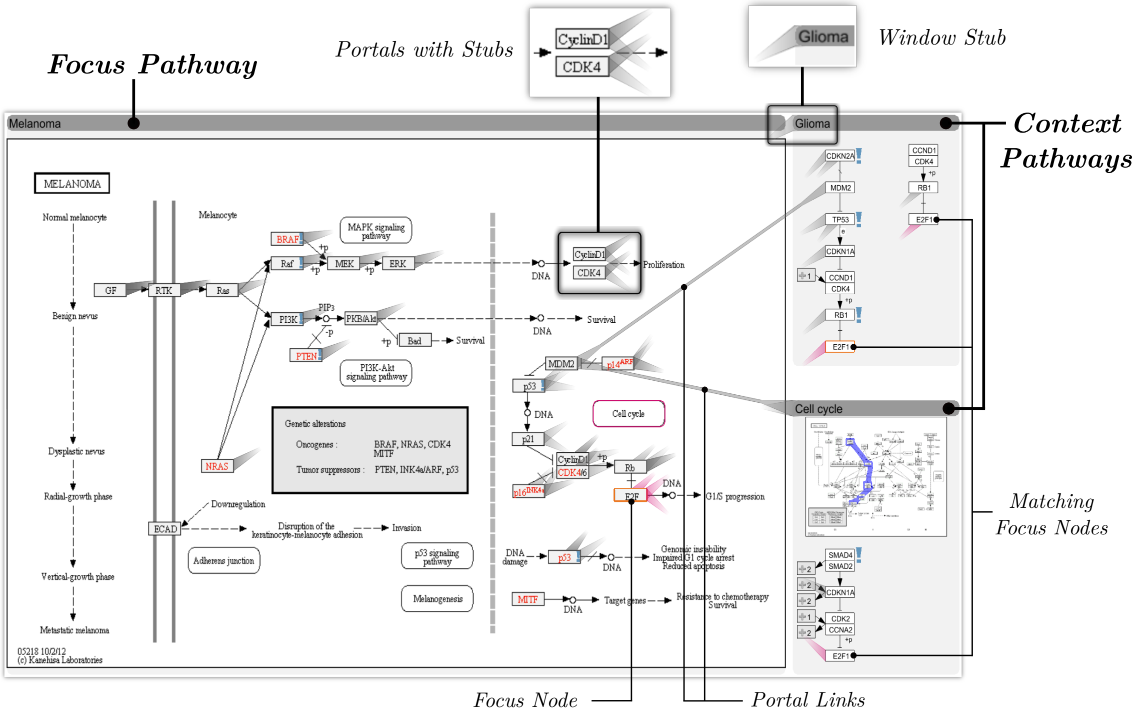


Figure 5.3: *The major components of Entourage. The* ***focus pathway*** *shows all details while the* ***context pathways*** *only show what is relevant in the context of the* ***focus node****. The insets at the top show how we indicate connections between pathways.*

5.2.1 Overview

Figure 5.3 shows Entourage's main components. The focus pathway takes up the majority of the space, while the context pathways are shown at the side. In this example, *E2F* was selected as the focus node, and the context pathways show their paths related to this node. Details on how context paths are selected are explained in Section 5.2.2, as are our methods to find relevant pathways.

Changes in focus are driven by user selections. However, choosing a meaningful focus is not always easy. Sometimes analysts will need to understand high-level relationships of pathways before they can set a sensible focus. Visualization is ideally suited to convey such high-level relationships. Relationships between pathways are largely driven by portals, as they connect two pathways. Showing portals and where they link to is therefore the most important aspect of showing high-level relationships between pathways. Figure 5.3 shows our approach for visualizing portals. We use a combination of stubs, which are shown for all portals at the same time, and visual links, i.e., visible edges, which are shown on request. These visual encodings efficiently convey high-level relationships between pathways and enable an analyst to set good focus points. Our visual encodings for showing relationships are explained in Section 5.2.3.

Finally, we need to address how to efficiently manage display space, as multiple focus and context elements compete for the limited screen real estate. We use an intelligent arrangement of pathways as well as multiple levels of detail for context pathways to optimize the display space, which are described in Section 5.2.4.

5.2.2 Determining Context Paths and Pathways

As discussed before, contextual subsets are based on showing contextual information for a user-chosen focus, i.e., a focus node of a pathway. This can help to quickly identify the role of the focus node in multiple pathways (B III). Which context information is eventually displayed depends on two factors: which paths in a pathway contain a focus node and which pathways are considered in the first place. Although potentially any subset of a pathway could be shown, we opted for paths due to their linear structure, which is favorable for layouting, and due to the high relevance of cascades of signals or reactions in pathways.

Determining Context Paths Context paths are selected by searching the graph for occurrences of the focus node or for immediately related nodes. Related nodes are, for example, nodes belonging to the same gene family. As it is common in nature that several distinct genes can fulfill the same role, albeit often with varying efficiency, pathway maps use both, a single label for the whole family or individual label for each of the family members. We consider these multi-mappings in our choices of relevant paths. This is the reason why occasionally differently labeled nodes are connected in Entourage.

Paths can either be unambiguous, as is the case in Pathway 3 of Figure 5.2(b), or contain branches, as in Pathway 2. If a path contains branches, we automatically determine the branch that is likely to be most interesting by calculating the most variable branch in terms of the underlying experimental data. We do so by calculating the standard deviation across all experiments for each of the mapped datasets and for every possible branch, and ultimately choose the branch that exhibits the highest deviation. The visual representation of context paths is essentially a more compact version of the path representation in EnRoute (see Section 4.2.2): We preserve as much of the topology in the vicinity of a path as possible. Incoming and outgoing branches are collapsed into abstract nodes to save space, but can be extended to full-size nodes and switched in to replace the main branch on demand. We decided against more complex attempts of linearizing larger portions of the network and including branches and cycles [126], to make the paths easy to understand for the analyst. Furthermore, we limit the length of automatically determined paths to what fits conveniently in the available space constraints, but give analysts the ability to extend the paths manually.

Determining Pathways Entourage shows pathway maps and context paths, respectively, only for pathways that were manually selected and added to the workspace. In order to quickly find relevant pathways, Entourage can suggest relevant pathways that are related to a given pathway or contain a selected focus node (B I). Figure 5.1 shows a list of pathways on the left side. This list shows all pathways that contain the currently selected focus node, or a node of the same gene family. The pathways in the list are ranked by their similarity to the current focus pathway. We calculate a similarity score for each pathway by computing the number of nodes shared with the focus pathway and normalize it by its size. The score is shown as a bar next to the pathway name. To quickly determine which pathways have already been added to the workspace, we mark loaded pathways using a dark gray background. In some situations analysts are interested in pathways that are generally similar to a selected pathway, without choosing a focus node. We use a similar algorithm to calculate scores of pathways in this scenario.

An alternative to the automatic, similarity-based list is an alphabetic list, which can be searched using keywords or regular expressions. This is especially helpful to find an entry point of an analysis. Finally, since pathway maps often embed related pathways, we enable the adding of such pathways to the workspace by clicking on embedded pathway nodes. The *Melanoma* focus pathway shown in Figure 5.3, for example, contains six embedded pathways indicating that these pathways play an important role in the context of the focus pathway. One of them (*Cell cycle*) is also a current context pathway, which is indicated by its purple border.

5.2.3 Visualizing Connections

To identify high-level relationships (B II) and find path intersections (B IV), we need to visually communicate which portal nodes connect two pathways. This requires visual encodings to (a) convey that a node (either in a focus or in a context pathway) is a portal and to (b) tell the analyst to which other portals it can be connected to.

Since related pathways often contain a substantial quantity of portal nodes, obvious approaches, such as color-coding or drawing visible edges, may easily fail. Even though objective (a) could be addressed by using a color-based highlighting of portal nodes, objective (b) would potentially require assigning many different colors to a single node. Visual links (i.e., visible edges), on the other hand, can connect a node to many others, but can result in significant clutter, given the many nodes and the dense layout of pathway maps, even if they were intelligently routed [153]. Therefore, we have chosen to primarily use stubs to encode relationships between nodes. Stubs were shown to be effective for indicating a connection without cluttering the display [35]. Figure 5.3 illustrates our stubs implementation. The two insets at the top show them in detail. For each pair of related portal nodes, we render a pair of stubs pointing at each other. The direction of a stub thus indicates the location of its target. We attach the stubs to the side of the node closest to the target, and quickly let them fade while they are converging to a point. We also show portals only with respect to the "active" pathway, i.e., stubs only point to and from the pathway on which the mouse pointer rests. This reduces the set of portals, minimizing clutter and ambiguities, while showing all relevant connections.

As context pathways only show a subset of nodes, potential portals might not be displayed. Nevertheless, we also want to communicate the presence of hidden portals. To achieve this, we show that a pathway has a relationship to the active pathway through one or multiple hidden portals by placing a stub on its window's title bar, as shown in Figure 5.3 (labeled *window stub*).

As is evident from the figures in this chapter, stubs are excellent at indicating connections between many portal nodes without introducing a high amount of visual clutter. However, they can be ambiguous at times, especially when the angle between two stubs attached to the same node is small. To resolve potential ambiguities, we show the exact connections out of a portal node by using visual links when the mouse hovers over the portal (labeled *portal links* in Figure 5.3).

Notice that while relationships between portals are generally indicated by gray stubs, the recurring focus nodes in the different pathways are emphasized by using purple stubs or links. This combination of gray and purple stubs and on-demand visual links results in a clean visualization showing cross-connections between pathways in a minimally obtrusive way. These visual encodings also work well for comparing two focus pathways. What remains to be discussed is how we can make good use of the limited screen space.

5.2.4 View Management

Using contextual subsets significantly reduces the number of elements that need to be displayed, yet preserves the relevant context. Nevertheless it is prudent to make good use of the available screen space. Here we describe how we optimize the arrangement, size, and amount of data shown in the various pathways under analysis.

When optimizing a layout for pathway analysis, one is confronted with a range of partially conflicting goals. The first and most obvious goal is to *maximize the amount of relevant content shown.* This often conflicts with the goal to *ensure legibility of all elements.* Following the contextual subsets concept, we always use at least one focus pathway, for which we comply with the legibility goal, thus limiting the remaining space for contextual information. To deal with the varying amounts of space, we promote and demote pathways to various levels of detail and optimize the pathway layout.

Levels of Detail Our approach to efficiently layout pathways requires us to change their size. We achieve this by introducing three levels of detail for context pathways, *high*, *medium*, and *low*, which are illustrated in Figure 5.4. The thumbnail used in the highest level is typically large enough to convey a sense of the overall topology of the pathway. In order to aid orientation, we highlight the route of the context path(s) in the thumbnail, as shown in the inset of Figure 5.4. However, we consider this topological information less relevant than the actual context, which is why we omit the thumbnail, if space is limited. In situations where there is not enough space to show any context paths, we resort to showing only the pathway titles. While this is not ideal, it is better than removing the

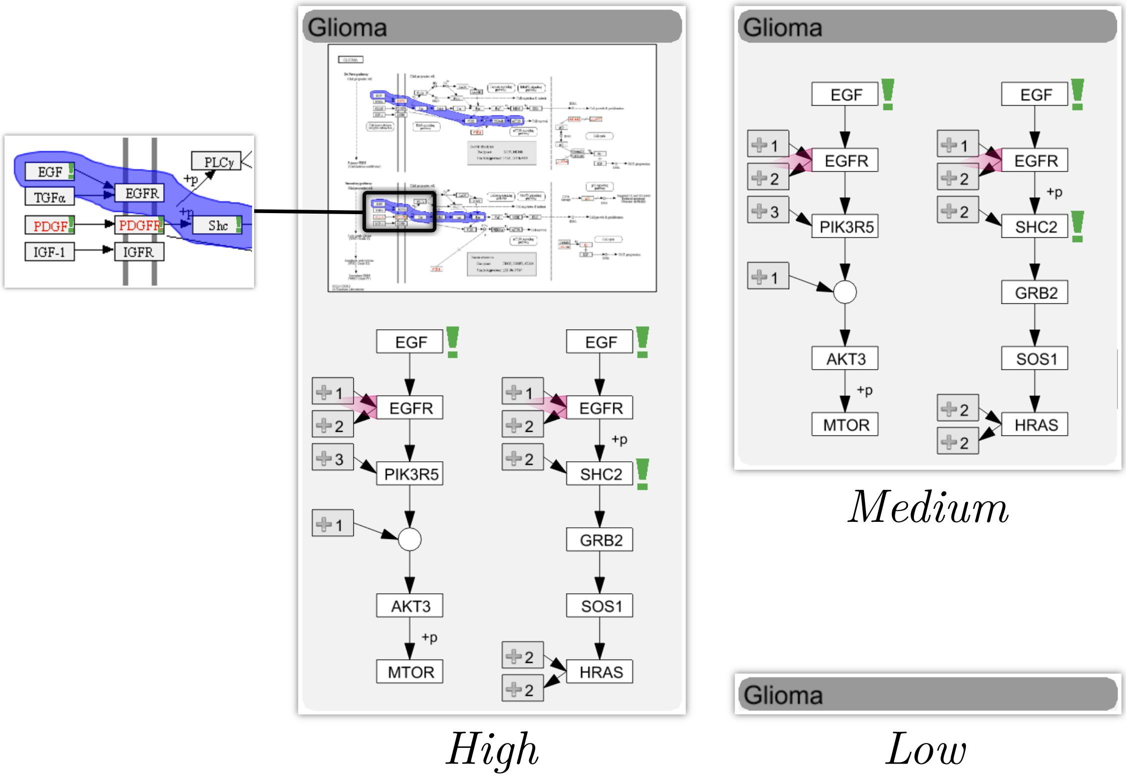


Figure 5.4: *The three different levels of detail of a context pathway. The highest level shows context paths plus a thumbnail of the overall pathway. Notice that the thumbnail also highlights the context paths. The medium level only shows the context paths, and the lowest level reduces the pathway to its title.*

pathway, since it can be conveniently brought back into focus or another level of detail, and it still indicates whether there is context information to be shown. Promotion or demotion of pathways between these levels of detail and the focus can be triggered manually, but is also done automatically. Automatic actions can be disabled for individual pathways. This also makes a high-level comparison of two focus-pathways possible.

Layout Optimization We decided to use a rigid column-based layout to arrange pathways as opposed to a free layout, since matrix-like layouts are more space efficient when it comes to layouting rectangular shaped objects like pathway maps. Also, a column-based layout is well suited to reflect the history of the analysis process by sorting the pathways by age. Entourage can accommodate as many columns as are reasonable for a given screen resolution, but always enforces at least one context column as well as a minimum width for each column.

Our initial implementation followed the goals outlined above, always aiming to maximize the visible context information while ensuring legibility. Early feedback, however, triggered the realization that another factor is essential: *layout stability*. Even though we were using animated transitions, which have been shown to be advantageous for tracing changes [76], we observed that our collaborators were irritated by larger changes in the pathway layout. As a consequence, we added the goal of *minimizing layout changes.* We also found that changing the size of a particular pathway is much less irritating than changing its position, either within or between columns. Consequently, our layout algorithm now prohibits position changes, unless the focus pathway is exchanged, but permits resizing and switching between levels of detail of context pathways.

Within a context column, we maximize the vertical space between individual pathway windows. While this might not be as aesthetically pleasing as stacking them on top of each other, this strategy serves a purpose: It helps to avoid ambiguities of stubs pointing to the pathways by increasing the angle between stubs.

To fulfill our goal of maximizing the amount of displayed content we promote and demote pathways intelligently. Automatic demotion of pathways is triggered when the horizontal or vertical display space is insufficient for displaying all elements at a reasonable size, while automatic promotion is triggered as space becomes available. An important decision in this regard is which pathways to demote or promote. This primarily depends on the causes of the space change. For example, if the vertical space is exceeded by the pathways in a context column, only pathways within that column have to be considered for demotion. In contrast, if there is too little horizontal space, the demotion of any pathway can potentially free up space.

To ultimately decide which of the pathways to demote or promote, we consider three properties with different priorities. The highest priority is given to pathways that contain a user-selected path (see Section 5.3). The second-highest priority is given to pathways that currently contain context paths. Finally, pathway "age" is considered as the lowest priority, where "young" pathways, i.e., those that were recently in focus, are given priority. We calculate a ranking of the candidate pathways based on these attributes and eventually demote the pathway with the lowest priority.

Overall, our layout considerations guarantee a stable and predictable management of many pathways. Together with showing the vicinity of a selected focus node in multiple context pathways, the possibility to bring any pathway into focus on demand helps to quickly determine the role of a node in multiple pathways (B III).

The techniques and encodings discussed in this section allow analysts to take a detailed look at one pathway, while always keeping an eye open for cross-connections to other pathways. By showing only the information relevant to the current analysis, the important parts of the data can be shown at full scale. We thus provide an analyst with the necessary tools to address the *Pathway Interconnectivity* requirements.

5.3 Combining Entourage with EnRoute

So far, we have focused on how to visualize relationships between pathways considering only the pathways and the underlying network. To address the *Pathway-Experimental*

Data Linking requirements, we combine Entourage with the EnRoute visualization technique, which was introduced in the previous chapter.

For easy identification of subsets of pathways that warrant detailed investigation, we employ the minimalistic on-node mapping introduced for EnRoute: On demand, we color the nodes according the average value of samples in a selected dataset and add a bar to indicate the variance in these samples. We also add exclamation marks to indicate high variance in samples of other datasets. The color of these exclamation marks encodes the associated dataset. As shown in Figure 5.5, we employ this on-node mapping in pathway maps and context paths.

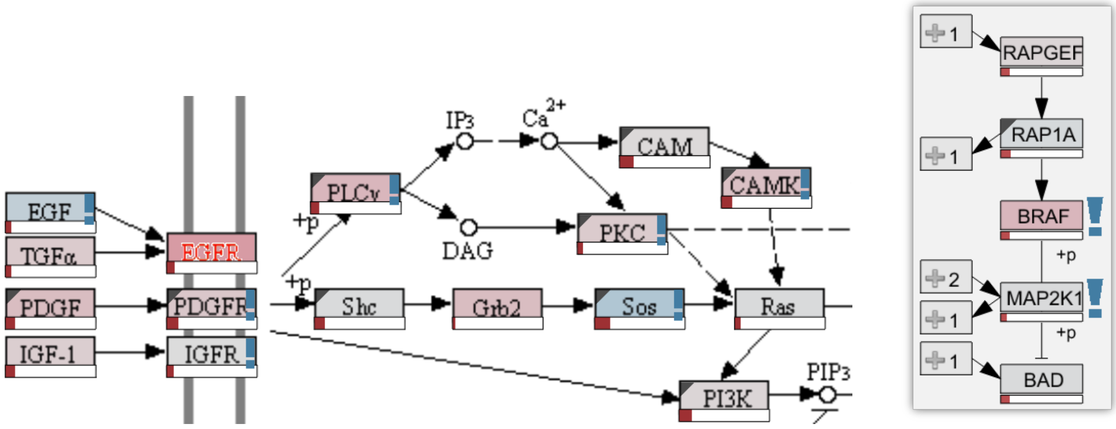
To inspect experimental data in detail, we employ and also extend the path extraction strategy of EnRoute. Paths are still selected by selecting start and end nodes and iteratively extending a path. However, while doing so, portal nodes can be used to cross pathway borders, as shown in Figure 5.1. This allows for path-based inter-pathway analysis of attributes. To highlight a path within a pathway, we make use of Bubble Sets, but chose not to extend them across pathways. Instead, we use the visual links we also use for portals, as the connecting portal nodes are in fact the same node.

As illustrated in Figure 5.1, the extracted path is displayed in a separate EnRoute view on the far right and also uses labels and horizontal lines to indicate pathway border crossings. By default, only the path is shown to grant space for the analysis of pathway relationships. However, this view can be expanded to show associated experimental data, as demonstrated in Figure 5.10. Depending on the current focus of the analysis, the amount of display space granted to the EnRoute view can be varied, ranging from none to full-screen.

Finally, to address the requirement to show relationships between cell line responses to drug treatment and genomic data (B V), we extended EnRoute to show contextual data that is not associated with genes. Such data is shown above the gene-associated data and uses the same ordering of samples. Figure 5.6 shows the compound sensitivity of ovary CCLE cancer cell lines to the drug *AEW541* on top of the expression (on the left) and copy number values (on the right) associated with the *RAF* gene family (*BRAF*, *ARAF*, *RAF1*). Here, low bars indicate high sensitivity, i.e., low IC_{50} values. Note that the samples are sorted with respect to their sensitivity to the compound. This is a simple yet effective way to facilitate the identification of relationships between genomic and pharmacologic data. Moreover, since for the small set of compounds, the targeted processes and genes are known, it is easy to identify paths where interesting relationships between genomic and pharmacological data occur.

5.4 Implementation and Scalability

Entourage is implemented as a view plugin for the Caleydo framework and currently works with pathway maps from KEGG and Wikipathways databases. Although we use the layouts provided by these databases in our current implementation, our technique



(a) (b)

Figure 5.5: *On-node data mapping in (a) pathway maps and (b) context paths. The node color in this example encodes the average copy number of mapped samples, while the red bars indicate the standard deviation. The exclamation marks indicate that the mapped experimental data varies considerably. The color of the exclamation marks and the standard deviation bars encodes the dataset in which the variation occurs. In this example, blue marks signal variation in mutation data.*

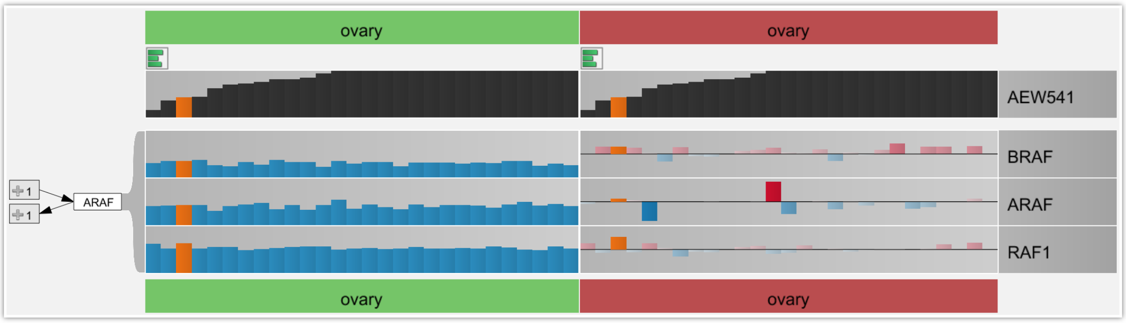


Figure 5.6: *Juxtaposition of pharmacological (on top) and genomic data. The pharmacological data captures the sensitivity of cell lines to drugs. The genomic data shown is mRNA expression (green, on the left) and copy number variation data (red, on the right). Orange bars are highlighted. Note that the samples are sorted by the magnitude of their responses to the drug AEW541.*

is not limited to those and can equally be applied to automatically generated pathway layouts.

Depending on the size of the current focus pathway, Entourage can display up to about ten pathways simultaneously, where one pathway is the focus pathway, while the other pathways are at least in “medium” level of detail on a full-HD display (see Figure 5.7). On larger, higher resolution screens, this number increases. The space for pathways can be increased by hiding currently unused views, such as the pathway list and the EnRoute view shown in Figure 5.1. Feedback from our collaborators indicates that this number of simultaneously explorable pathways is sufficient in all but the rarest cases and superior to other systems. We believe that our visual encodings are also suitable to point at interesting relationships outside an analyst's primary field of view, making it suitable for the increasingly large displays that are becoming commonplace. On conventional displays, we typically limit the number of focus pathways to one in order to guarantee readability. This number, however, can be temporarily increased if detail about the structural relationship of pathways should be shown.

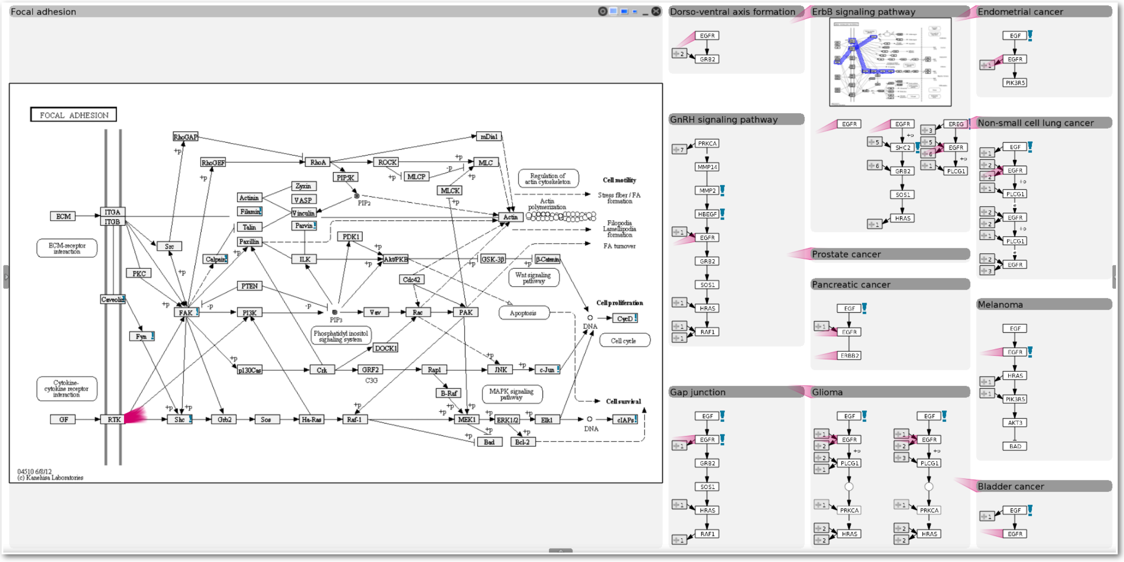


Figure 5.7: *Entourage with one focus pathway and ten context pathways with a detail level of at least "medium". The pathway list and the EnRoute view are hidden to grant more space for the display of pathways.*

5.5 Case Studies

Entourage was developed in a user-centered design process including weekly meetings between the visualization developers and multiple domain experts. As a result of these meetings, we have established previously discussed domain goals and requirements. We deployed various iterations of Entourage and our primary contact, a chemical biologist, used Entourage over a period of four weeks. During this period, we were in constant contact with her and refined various aspects of the system. The case studies presented here report on her observations. Prior to the deployment of Entourage, the team was using conventional pathway tools and had to resolve any questions concerning pathway relationships manually. For visualization of experimental data they mainly relied on tools like TIBCO Spotfire[1], whereas interactions between biomolecules were analyzed with network

[1] http://spotfire.tibco.com/

visualization tools like Cytoscape [150] or the KEGG web interface. Hence, the combination of Entourage and EnRoute allowed them to integrate two analysis steps that were previously carried out separately into one single task.

In the following, we describe case studies that demonstrate Entourage for judging drug side-effects, identifying potential for drug repositioning, and understanding drugs' mechanisms of action. The case studies either describe a novel observation or clearly demonstrate how a known effect can be rationalized with Entourage.

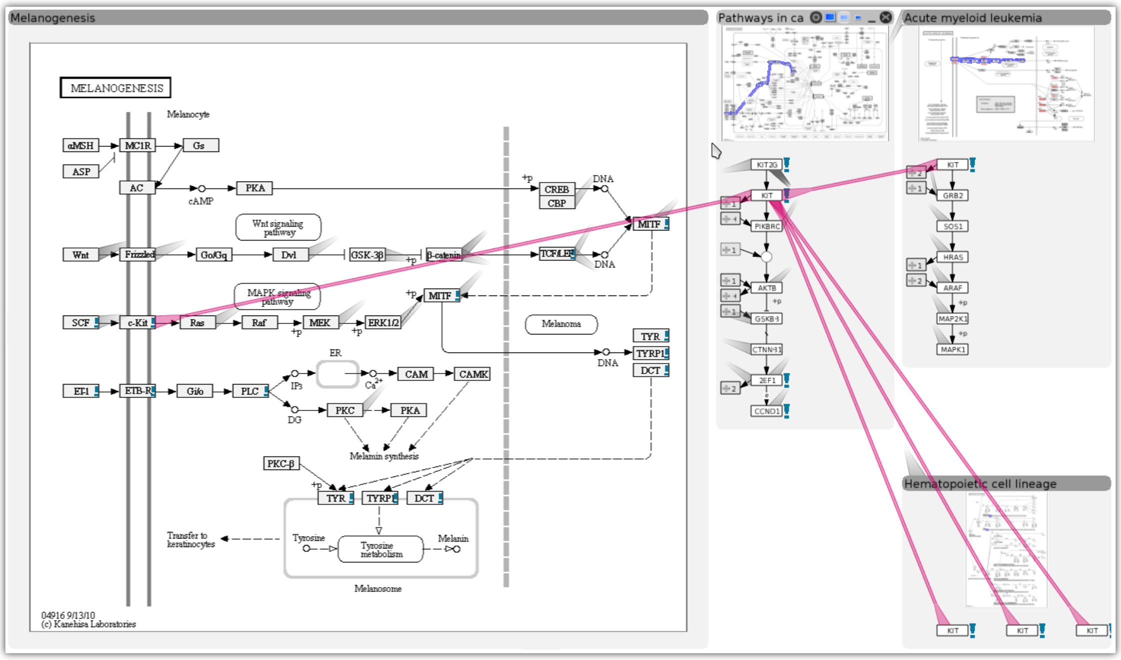


Figure 5.8: *Imatinib is a drug that is used in cancer treatment and inhibits c-Kit. However, Imatinib has side effects such as loss of skin color and bleeding problems. Setting c-Kit as the focus node reveals that it is involved in only four pathways, but two of them, the Melanogenesis and the Hematopoietic cell lineage pathway can be related to the side-effects observed.*

5.5.1 Rationalizing Imatinib-Associated Side-Effects

Kinases are enzymes that transfer phosphate groups to substrates, thereby communicating signals in the cell. *Imatinib* is a drug acting as a kinase inhibitor that was developed to specifically target the *BCR-Abl* kinase, a fusion protein that results from a chromosome translocation. This protein is exclusively expressed in specific cancer cells, in particular in *Philadelphia chromosome-positive chronic myelogenous leukemia (CML)* [4]. However, it was later found that *Imatinib* is less specific than originally assumed and also targets other protein kinases, such as *c-Kit*, *PDGFRalpha*, and *PDGFRbeta*. Hence, *c-kit* expressing gastrointestinal tumors are now the primary indication for this drug.

Side effects that have been observed in *Imatinib*-treated patients are *hypopigmentation*,

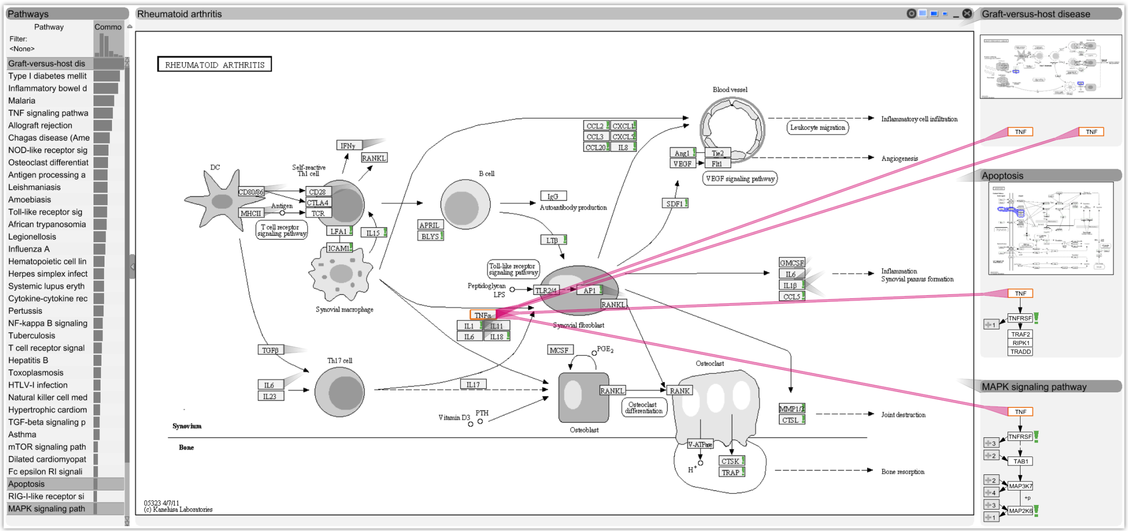


Figure 5.9: *TNF-alpha (the focus node) was originally explored as a target for the Graft-Versus-Host Disease (GVHD, top-right pathway). However, when tested in clinical trials, TNF-alpha inhibiting compounds were not effective against GVHD, but could later be repositioned for the treatment of Rheumatoid Arthritis (focus pathway). Entourage shows Rheumatoid Arthritis as closely related to the GVHD pathway (see pathway list on the left). Entourage also reveals seemingly contradictory roles of TNF-alpha. It is involved in cell death (Apoptosis) and also in cancer (i.e., uncontrolled cell growth) through the MAPK signaling pathway.*

i.e., loss of skin color, and *myelosuppression*, i.e., reduced blood cell and platelet formation, which can lead to frequent infections and severe bleeding problems [71]. Our collaborator wanted to find out whether these side-effects of the drug can be rationalized using inter-pathway analysis. She started by opening the *Pathways in cancer* map, since she knew that it would contain all targets of *Imatinib*. From there, she queried for all pathways in which the targets of *Imatinib* are involved. She found that only four other pathways contain *c-Kit*, but that two of them seem related to the observed side-effects. The first one is the *Hematopoietic cell lineage* pathway, which highlights the role of *c-Kit* in early *hematopoiesis*, i.e., blood formation, whereas the second one is the *Melanogenesis* pathway, which shows that *c-Kit* is involved in the development of *melanocytes*, which are the cells that are responsible for skin pigmentation. Figure 5.8 highlights the role of *c-Kit*, the focus node in the figure, in these pathways. Our collaborator stated that, by making use of the visualization of pathway interconnectivity, she was able to find possible reasons for the adverse effects caused by inhibiting *c-Kit*, which would not have been possible if the pathway map showing the role of *c-Kit* in cancer had been studied in isolation.

5.5.2 Rationalizing Successful Drug Repositioning

Graft-versus-host disease (GVHD) is frequently observed after tissue or organ transplantation and is caused by immune cells that originate from the donor and were transplanted with the tissue. These immune cells perceive tissue of the recipient as foreign and attack it, thereby causing damage. The *Graft-versus-host disease* pathway identifies *TNF-alpha*, a gene involved in inflammation, as an important player in the disease. Accordingly, molecules counter-acting (inhibiting) the effect of *TNF-alpha* have been evaluated for preventing GVHD in transplantation patients, with no success. However, the roles of *TNF-alpha* in the organism are manifold, as our collaborator was able to demonstrate using Entourage, when she chose *TNF-alpha* as her focus node, revealing all associated pathways. She found that one of the highest scoring and therefore most similar pathway to the *Graft-versus-host disease* map is the *Rheumatoid arthritis* pathway, shown as the focus pathway in Figure 5.9. Indeed, as clinical safety for *TNF-alpha* inhibitors had been proven in the initial trials for GVHD patients, the molecules were revisited and tested for their efficacy in patients suffering from rheumatoid arthritis. In this case, anti-*TNF alpha* therapy showed the desired clinical effect, and, today, *TNF* inhibitors are part of the standard treatment of rheumatoid arthritis. The domain expert pointed out that Entourage also ranks the *Apoptosis*, *MAPK signaling*, and *NF-kappa B signaling* pathways, which explain the controversial role of *TNF-alpha* in cancer. While the *Apoptosis* pathway shows the process by which *TNF-alpha* leads to cell death, the other two pathways point out how *TNF-alpha* contributes to cell survival. Accordingly, the benefit of *TNF-alpha* inhibitors in anti-cancer treatment remains an open question, and clinical trials are awaited to further explore the potential use of these molecules in malignancies.

5.5.3 Relating Genomic Features to Compound Sensitivity

To explain different compound sensitivities of cell lines, our collaborator used the previously introduced *CCLE* dataset. This dataset contains data on the inhibitory effects of 24 drugs against roughly 500 cell lines from different cancer tissues and genomic data. Ideally, a drug completely inhibits the growth of these cell lines at minimal concentrations. First, she wanted to investigate factors that sensitize cell lines to the drugs *Lapatinib* and *Erlotinib* that inhibit members of the *ErbB* gene family and are used in cancer treatment. The *ErbB* family is a family of epidermal growth factor receptors that are known to play an important role in tumor growth. The drug *Lapatinib* is a dual inhibitor of *EGFR* and *ErbB2*, while *Erlotinib* is a known inhibitor of only *EGFR*, all of which belong to the aforementioned family. Due to its immediate relevance, the expert started by loading the *ErbB signaling pathway* into Entourage. By searching for related pathways, she found several cancer-specific pathway maps. The pathways *Glioma* and *Non-small cell lung cancer* ranked among the top on the list (see Figure 5.10). She commented that this indicates that the *ErbB signaling pathway* is a key player in these diseases. For the *ErbB pathway map*, our collaboration partner was interested in the experimental data for the genes in the path that leads from *ErbB* receptors to *Myc*, a gene known to regulate cell growth. She also noticed that *ErbB2* was highlighted with a red exclamation mark indicating high variance in the copy number data. She thus selected the genes of this path for an in-depth analysis.

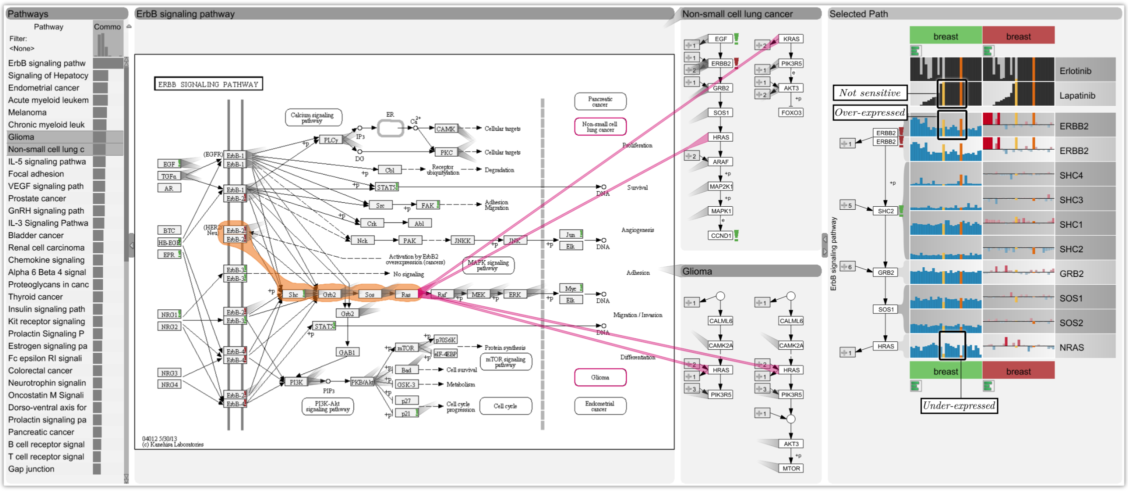


Figure 5.10: *The ErbB signaling pathway (the focus pathway) is a target of the drugs Lapatinib and Erlotinib, which are used for cancer treatment. As shown in the pathway list on the left, which results from a query for similar pathways, the ErbB signaling pathway is related to many cancer pathways. A signaling cascade from ErbB2 to Ras is selected. The integrated EnRoute view shows copy number and mRNA expression data for breast cancer cell lines. The sensitivity of the different cell lines to Lapatinib and Erlotinib is reported at the top. For the shown cell lines, increased copy numbers of ErB2 (high red bars in the ErbB2 row) result in over-expression of this gene (high blue bars). Furthermore, there is a strong relation between ErbB2 over-expression and sensitivity to Lapatinib (high blue bars for gene over-expression in the ErbB2 row coincide with low bars in the Lapatinib row). This means that Lapatinib is effective, if ErbB2 is highly expressed. There are, however, two exceptions - the highlighted cell lines (yellow and orange), for which an under-expression in Ras downstream in the pathway is observed, likely causing Lapatinib to be ineffective in these cases. While this observation was made for breast cancer tissue, exploring the related context pathways by setting the focus node to Ras reveals that the same signaling cascade (i.e., path) is also contained in the non-small cell lung cancer pathway. Thus, it would be interesting to explore the transferability of the observed resistance pattern to this tissue type.*

She then looked at this path's gene expression data in the embedded EnRoute view and combined it with sensitivities to *Erlotinib* and *Lapatinib*. For the analysis, cell lines were grouped by their tissue of origin (e.g., breast, ovary, liver, etc.) and sorted by sensitivity to *Lapatinib*. Her first observation when looking at the experimental data was that the two drugs displayed inhibitory activities across cell lines from many different tissues. The cell lines from lung, breast and three other tissues were in general most responsive. The set of cell lines that were responsive to *Erlotinib* and *Lapatinib* largely overlapped, although *Lapatinib* showed a broader spectrum of activity than *Erlotinib*. She found a strong co-occurrence between *ErbB2* mRNA over-expression and sensitivity to *Lapatinib* in lung and breast cancer cell lines, a trend that was less apparent or not observed at all for other responsive cell lines.

She then chose to focus on cell lines from breast and also investigated copy number variation for these cell lines. For most breast cancer cell lines that over-expressed *ErbB2*, high copy numbers of this gene were found, i.e., the increased expression could generally be traced back to an increased copy number. Interestingly, only two breast cancer cell lines that showed strong over-expression of *ErbB2* did not respond to *Lapatinib* treatment. The columns of these two cell lines are highlighted yellow and orange in Figure 5.10. She then tried to find the cause for this effect and examined the expression of downstream genes in the pathway. She found that for these two cell lines, the gene *Ras* was strongly under-expressed (also shown in Figure 5.10).

It is straightforward to assume that this under-expression further down the path counteracts the over-activation of the pathway by increased *ErbB2* expression, explaining the resistance to *Lapatinib* treatment that reduces the effects of ErbB2 expression. Our collaborator stated that this highlights the importance of being able to analyze genomic data in a pathway context, because compound sensitivities can often only be explained by the interplay of multiple genomic features.

Based on this observation, she started to investigate whether other cancer-related pathways contain the same signaling cascade, i.e., path. She selected *Ras* as the focus node of her analysis, which revealed several other pathways that contain the same *ErbB* signaling cascade. Figure 5.10 shows an example where it is obvious due to the many parallel stubs that the cascade is indeed identical. Our collaborator commented that it would be interesting to investigate in the future whether a similar gene expression pattern in these cancer types would also entail resistance to *Lapatinib*.

Overall, our collaboration partners were excited about the analytical capabilities of Entourage and its combination with EnRoute. They mentioned that they perceived a significant improvement over their previous tool-chain. While, for example, the KEGG interface could be used to conduct an analysis similar to the one described in the second case study, doing so would be very tedious, as KEGG provides no support for analyzing relationships of pathways. They highly valued the ability to immediately see all relevant related processes for a pathway and being able to compare them easily and to see experimental data in the context of pathways.

5.6 Summary

In this chapter we introduced Entourage, a visual analysis technique for the exploration of biological networks by connecting multiple pathways. From a visualization point of view, Entourage addresses two main challenges: how to visualize multiple pathways in a scalable way and how to indicate the potentially large amounts of relationships between them. The former challenge is addressed by the use of contextual subsets: one pathway is shown in full detail, but only contextually relevant paths are displayed for other pathways. As the focus of the analysis changes over time, users may flexibly adapt what is displayed in focus and context. To address the latter challenge, we use visual links that connect the nodes of different pathways. To avoid clutter, we only show stubs by default, but they can be expanded to links on demand.

In addition to introducing Entourage, we also we also discussed how EnRoute can complement Entourage in order to address scalability with respect to both graph size and attributes: While multiple pathways and their relationships can be analyzed with Entourage, EnRoute allows for the detailed inspection of associated experimental data. We developed Entourage and combined it with EnRoute to specifically support pharmacological research. Therefore, their utility was demonstrated in case studies reflecting goals in drug discovery, i.e., the judgment of side-effects, the identification of potential for drug repositioning, and the understanding of drugs' mechanisms of action.

Although Entourage was specifically designed to address issues in drug discovery, we argue that its concept can be applied to any partitioned graph, provided that each individual partition is small enough to be fully displayed. Also, in the case of pathways, the partitioning of the biological network is non-exclusive. Thus, nodes are shared by multiple partitions and these shared nodes represent the relationships between pathways. However, edges are equally conceivable to represent relationships between partitions when the partitioning is exclusive and there are no shared nodes.

The employment of contextual subsets, i.e., the flexible sub-setting of whole pathways to context paths, is the core concept of Entourage to achieve scalability. Although other forms of graph subsets are conceivable, we found the space-efficiency and flexibility of layouting paths particularly advantageous.

Up to this point, we introduced techniques that use paths as subsets to deal with scalability problems in graphs with regards to graph size and attributes. In the next chapter, we shift the focus from a more general exploration of graphs to the analysis of multiple paths and their relationships.

6 Visualizing Multiple Paths

Contents

The analysis of paths in graphs is highly relevant in many domains. For example, learning about how two suspects are connected in a criminal case or how one protein influences another are examples of important domain tasks that can be abstracted to path analysis tasks. In many scenarios, short or cheap paths between two nodes need to be identified. As discussed in Chapter 3, shortest paths can be computed with, for example, breadth-first search (BFS) or, when considering edge weights, with Dijkstra's algorithm. In practice, however, the shortest path might not be the *best* path, as other factors such as node or edge attributes or contextual knowledge play a role in identifying important paths. Also, in many cases, defining a cost-function to find the cheapest path is not straightforward and requires human judgment. Thus, visually exploring multiple candidate paths, which can be computed with k-shortest path algorithms, is important. In practice, such a visual exploration is difficult, due to the sheer size of the underlying graph, large amounts of attributes that are important for judging the paths, and also the potentially large number of path alternatives. In the previous chapter, we discussed the combination of the Entourage and EnRoute visualization techniques to deal with large partitioned graphs and also large numbers of attributes. However, this combination allows for a detailed inspection of only one single path at a time.

In this chapter, we introduce **Pathfinder**, a visual analysis technique for the investigation of multiple paths. To make Pathfinder scale to large graphs with tens of thousands of nodes and edges, our approach is based on queries: We provide visual methods to query for paths, while considering various constraints. As illustrated in Figure 6.1, the set of paths that matches a query is visualized in both a list and as a node-link diagram. To be able to judge paths based on attributes, the list displays both paths and associated attributes in a way that is reminiscent of the path representation in EnRoute. As the resulting paths can be numerous, they can be ranked to quickly identify relevant paths. The ranking can be based on various criteria, such as path length, average node degree, or scores derived from attributes. To explore relationships between different paths, we also employ various aids for path comparison. The node-link diagram complements the path list by showing the topology of paths and providing an overview. Overall, Pathfinder allows for an iterative analysis approach: Starting from an initial query for potentially interesting paths, these paths are ranked by user-defined criteria to identify most relevant paths. More detailed investigation of the paths with respect to attributes and topology might lead to refinements of the query, or re-ranking of paths.

Throughout this chapter, we illustrate Pathfinder with data from a coauthor network and biological pathways. In the following, we discuss requirements for effective multi-path analysis, introduce and discuss the Pathfinder visualization technique, and demonstrate Pathfinder in a pathway analysis case study.

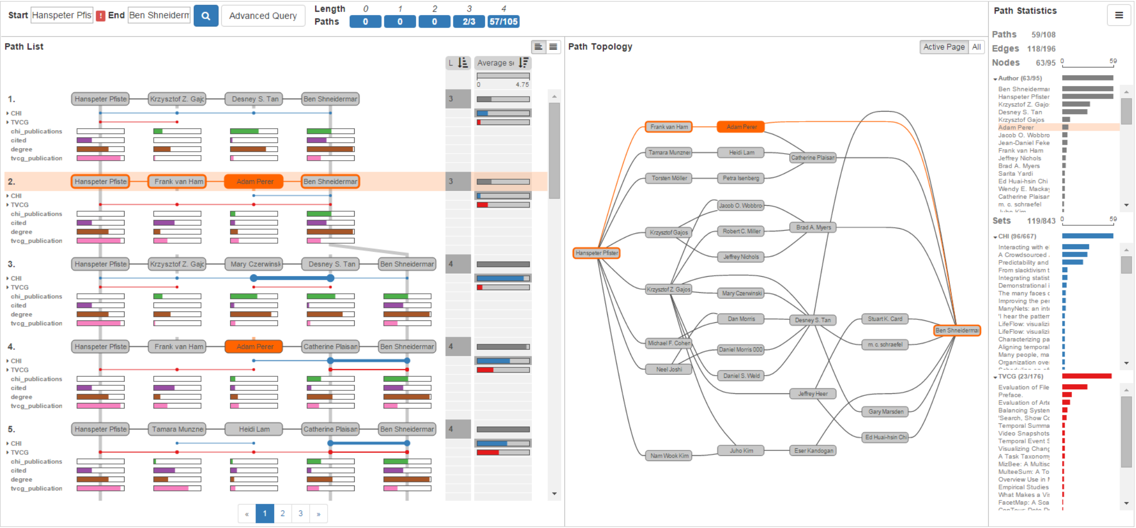


Figure 6.1: *Pathfinder visualizes multiple paths of a coauthor graph connecting Hanspeter Pfister and Ben Shneiderman. The paths are shown in a ranked list together with associated sets and attributes on the left (path list view). To its right, a node-link diagram shows the topology of the paths (path topology view). The path statistics view on the far right shows an overview of the properties of the paths.*

6.1 Requirement Analysis

The main objective of Pathfinder is to support the investigation of multiple path relationships. Thus, we introduce a set of requirements for a multi-path visualization technique. In our requirement analysis, we focus on path investigation in large, static networks that are highly multivariate, i.e., nodes and edges can be associated with rich attributes. Also, sets may be defined that capture additional relationships between the nodes. Our choice of requirements is based on discussions with potential end-users, our analysis of the literature on path visualization, and the task taxonomy for general graph visualization by Lee et al. [110]. As discussed in Section 3.1.1, Lee at al. distinguish topology-based tasks and attribute-based task, as well as browsing and overview tasks. The browsing tasks are related to topology (*follow path* and *revisit*), and the overview task is concerned with analyzing general properties of the graph, such as estimating the overall size. For our requirements, we assume an underlying fundamental task of exploring paths. Thus, some of the tasks introduced by Lee et al., especially regarding general connectivity (identifying clusters, connected components, bridges, and articulation points), cannot be addressed with a pure path-based approach.

C I: Query for paths. Users should be able to easily query for paths that adhere to some criteria. A simple query searches for the paths connecting two nodes. Other criteria, such as querying based on sets (find short paths that connect node A with any node in set S) or topological restrictions (find short paths from A to C that do not go through B), must also be supported. It should be easy to refine an existing query, as analysts often can identify restrictions once they see results matching their initial query.

C II: Visualize attributes. Many networks contain rich and heterogeneous attributes for nodes and edges. Understanding these attributes is often critical for judging paths. In a gene regulatory network, for example, low values for associated experimental data can tell analysts that the path is inactive for the given samples.

C III: Visualize group structures in paths. Group structures, such as set relationships and clusters, provide additional information about relationships between nodes. For example, they are important when judging the relevance of an edge in a path. If two connected genes occur in many pathways, for example, it is likely that their relationship is important.

C IV: Rank paths. A common goal of users querying for paths is to find “good” paths, according to some criteria. These criteria sometimes are as simple as finding the shortest path, but can also involve a more intricate combination of topological features and attributes. A path visualization technique should allow its users to dynamically define these criteria and rank paths according to them.

C V: Visualize topology context. The relevance of a path for an analysis can be influenced by its surrounding topology properties. For example, in a gene-regulatory network, feedback cycles are common, and it is important to know whether a path is involved in one of these cycles.

C VI: Compare paths. Comparing paths is important to evaluate similarity or dissimilarity between paths. As for path ranking, comparison can be based on many criteria, such as shared nodes, common set relationships, or similar attribute values. These similarities and differences are often not easy to spot, so a path visualization technique should make this comparison easier.

C VII: Group paths. It is common that many paths match a specific query, but multiple paths will be only slight variations of each other. Ideally, a system should convey which groups of paths are similar and provide an overview of the main path classes, as this can reveal important paths that do not rank at the very top based on other criteria.

In addition to these specific requirements, a path visualization system must be scalable, as the costs of using and learning a sophisticated path visualization technique mainly pay off for large graphs. Finally, the system also should follow best practices of conventional graph visualization, such as overview (e.g., graph statistics) and details on demand (e.g., full information about sets or nodes).

6.2 The Pathfinder Visualization Technique

Our goal was to develop a method for path analysis that is very scalable and addresses the requirements introduced in Section 6.1. The key to achieve scalability is **path queries**: we introduce a rich query interface that can cover a wide range of questions relevant to path analysis (C I). The visual representation of the graph is manifested in the **path list view**, the **path topology view**, and the **path statistics view**. These views are part of a multiple coordinated view setup that supports synchronized highlighting of paths and their elements. The path list is ideally suited to visualize attributes and sets (C II, C III) due to the linear layout of the individual paths, which allows us to juxtapose the nodes and edges with plots of the attributes. It is also a perfect match to dynamically rank paths and explore those rankings [66] (C IV). Additionally, the path list can be used to compare paths (C VI), especially with respect to attributes and sets. The shortcomings of the path list are addressed by the path topology view, which provides the topological context (C V) and covers the aspects of comparison with respect to topology (C VI). Also, the topology view is beneficial for an overview and for initial orientation.

In the following sections, we introduce the various components of Pathfinder in detail. We use two networks to illustrate Pathfinder. The first is a coauthor network that we created from extracting all ACM CHI and IEEE TVCG papers from the *DBLP computer science library*[1]. We treat authors as nodes and add an edge if two authors have coauthored

[1] `http://dblp.uni-trier.de/`

a publication. We also consider individual papers as sets, i.e., a paper is a set that contains all its authors. As attributes in this network, we provide node degrees, paper counts for each venue, and the number of citations to the visualization papers, which we extracted from the *Visualization Publication Dataset* [84]. This network contains about 34,000 nodes, 45,000 edges, and 13,000 sets.

The second network is a biological network extracted from the KEGG pathway database. We treat pathways as sets of nodes and supplement the network data with gene expression and copy number data from the CCLE dataset. In total, this network contains about 11,000 nodes, 71,000 edges, and 300 sets (pathways).

6.2.1 Path Queries

As queries are at the heart of the Pathfinder technique, we introduce a sophisticated graphical interface for querying, paired with choices of algorithms and concepts that make the analysis and querying process efficient. The most basic query in Pathfinder is for paths that connect two nodes. To make this simple use case as smooth as possible our default query interface consists of two input fields only: one for the start, and one for the end node.

Once such a query is triggered, we run a k-shortest path algorithm based on breadth-first search (BFS) to retrieve the shortest paths between two nodes. While BFS cannot consider edge weights, it is faster than other approaches and response time is critical for interactive analysis systems. Also, we can consider weights later in the analysis process through interactive rankings on the resulting set of paths.

Another method to improve scalability is incremental results. Incremental and approximate results have been shown to accelerate and open up the query process [51]. Consequently, we visualize each path and all its attributes, as soon as it becomes available, even if the search process is still ongoing.

We retrieve new paths, until we reach a fixed threshold of k paths. Once we have passed this number, we continue to fetch all paths that are of equal length l to the last path fetched. Using this approach, we can guarantee that we consider all paths of length l in the subsequent analysis. However, in rare cases, the resulting number of paths can be very large. To avoid excessive computation, we also define a maximum threshold on the number of paths, which is significantly larger than k. As shown in Figure 6.1, the fields to the right of the query interface give an overview of how many paths of which length are in the current result set.

We also support queries between sets of start and end nodes, which can be useful, for example, to find out how an author is connected to the authors of a paper of interest. In addition to the simple query interface, Pathfinder also provides an advanced interface for specifying more complex queries that consider topological restrictions, logical combinations of nodes, and node and edge properties. Topological restrictions can be used to consider, for example, only paths that go through a certain node, or contain a sequence of certain

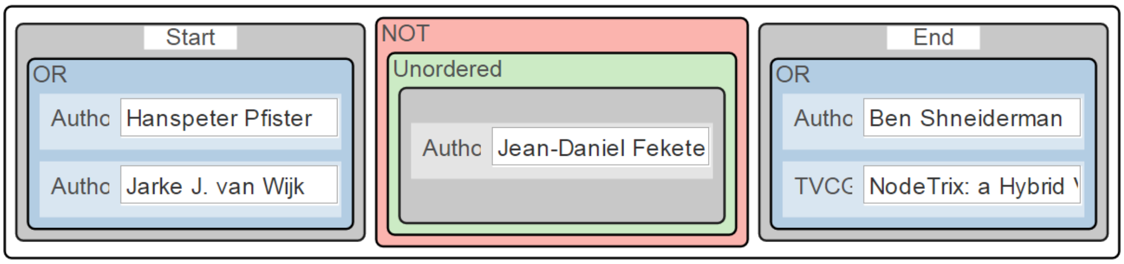


Figure 6.2: *The advanced query interface showing a query where the start node is either Hanspeter Pfister or Jarke van Wijk, the paths must not contain Jean-Daniel Fekete, and the last node is Ben Shneiderman or an author of the NodeTrix paper.*

nodes. Logical restrictions can combine nodes and sets through Boolean operations (AND, OR, NOT). Also, Pathfinder can treat set relationships as edges, if desired. Figure 6.2 shows a complex query example for a coauthor network, with an OR combination at both the start and the end node and a topological restriction to exclude all paths that go through the node between them.

Queries can be expressed either in the query interface or by interacting with elements in any of the views. For example, a user can exclude all paths through a certain node via its context menu in the path list view, which simplifies query refinement (C I).

Any change to the query acts as an immediate filter for the current set of paths, which guarantees rapid feedback. This is often sufficient for queries that restrict the result set, but queries that expand or change the result set have to be run against the whole network to produce reliable results.

Besides path queries, Pathfinder also supports querying for individual nodes and their neighbors. In this way, the network can be explored by iteratively expanding the shown subset in the topology view.

6.2.2 Path List View

As shown in Figure 6.1, Pathfinder's main view displays all paths that match a query in a ranked and paginated list of paths. Each path is displayed as a sequence of nodes and edges. For example, Figure 6.3 shows the path from *Jean-Daniel Fekete* to *Ben Shneiderman* in the coauthor network. A key benefit of this linear layout is that it allows us to easily show attributes associated with the nodes and edges (C II) and sets connecting the nodes below the path (C III). Conceptually, this display of paths is similar to the path representation in EnRoute, but as multiple paths are displayed in a list, the orientation of paths and attributes differ. Also, aggregation of attributes and sets and showing details on demand is more relevant, as the display of multiple paths exacerbates scalability issues. Additionally, attributes and sets can be hidden on demand, which allows analysts to focus on the paths exclusively.

Set-memberships of a node are indicated by a circle below the node. If two successive nodes are within the same set, the circles are connected by a line. As shown in red in Figure 6.3, tracing set relationships, such as common publications of authors in a coauthor network, is easy. To save space, the sets can be aggregated by type. These aggregated relationships are scaled according to the number of represented sets: Thicker lines and larger circles represent more sets. Figure 6.3, for example, shows that *Enrico Bertini* coauthored only a few CHI publications (blue) with *Catherine Plaisant*, but she coauthored many papers with *Ben Shneiderman*. By default, we show only sets that connect nodes. However, all available sets can be shown on demand.

When visualizing attributes, we can show either bar charts, as illustrated in Figure 6.3, or box plots, as shown in Figure 6.7. However, other representations, depending on the data type, the number, and the structure of the associated data, are equally conceivable. Attributes can be aggregated hierarchically. The box plots in Figure 6.7, for example, summarize hundreds of mRNA and copy number data values, either for the whole dataset, or for individual groups of samples taken from different tissue types.

By default, we assign colors to set types and attribute types, in order to facilitate identification across different views. However, coloring can be disabled on demand.

Taken together, the visual encodings that are made possible by the linear representation allow us to efficiently visualize group structures and attributes in graphs, which are both considered challenging at scale [98, 173].

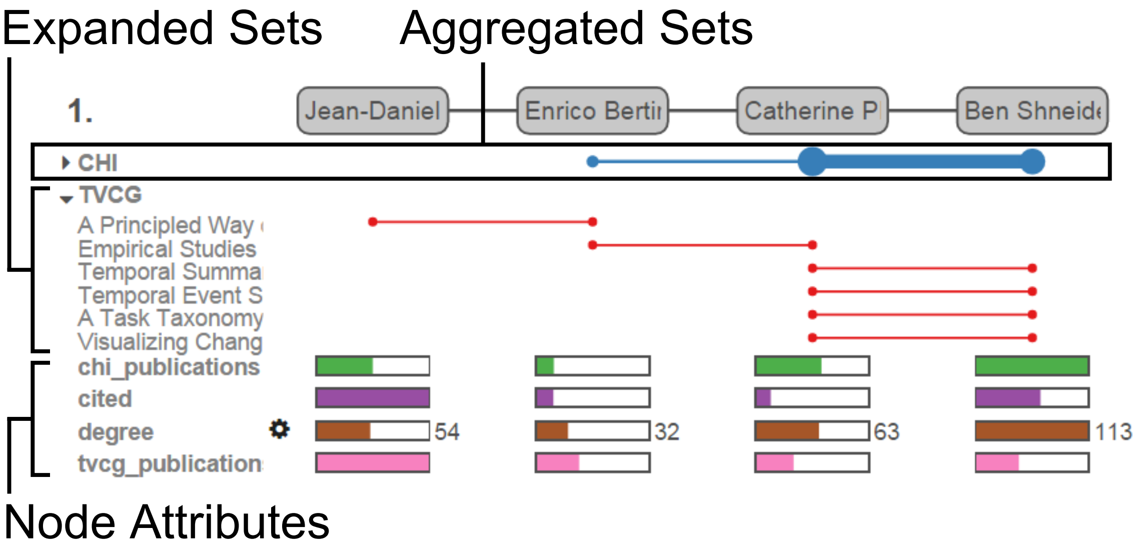


Figure 6.3: *A path from the coauthor network connecting Jean-Daniel Fekete and Ben Shneiderman. The coauthored papers are treated as sets and visualized below the path. The CHI papers are aggregated — the strength of lines and size of circles indicates the amount of papers. We see that Catherine Plaisant is highly connected with Ben Shneiderman via CHI papers. The TVCG papers show details: We can see who has coauthored which papers. Below the sets, numerical attributes, such as the number of publications in each venue, are shown for each node.*

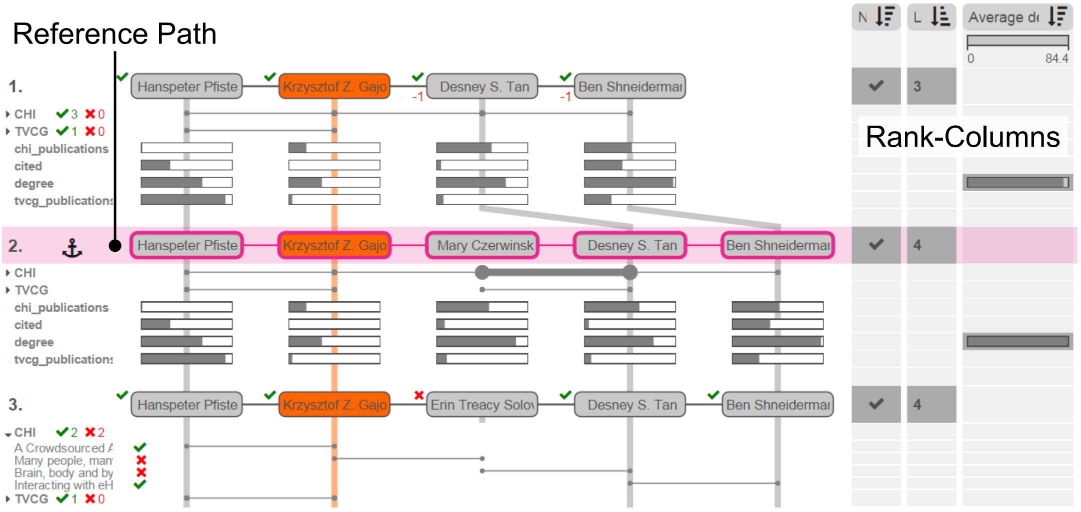


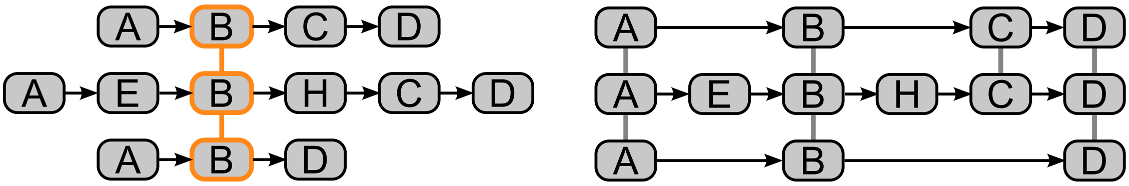
Figure 6.4: *Path comparison and ranking. The second path (violet) is the reference path. Icons next to the nodes and sets in all paths indicate whether they are shared with the reference path. For example, the third path shares two CHI papers, one TVCG paper, and four authors. The rank columns on the right show different scores by which the paths are ranked. Their order indicates score priority: First, paths are ranked by whether they contain Krzysztof Z. Gajos, second, by length, and third, by the average node degree.*

Path Ranking. A core aspect of our method is the dynamic ranking of paths (C IV). In concert with query refinement, dynamic ranking allows us to utilize human intelligence, contextual knowledge, and individual judgment for path analysis. It enables us to consider paths based not only on their topological properties or simple attributes, such as edge weights, but based on complex relationships of attributes, topology, and group structures.

By default, paths are ranked by their length, i.e., number of edges, but other rankings based on node attributes (e.g., minimum, maximum, average value across a path), the number of connecting sets, etc., are easily accessible. In addition, we empower users to write custom scoring functions using a script editor, which enables complex, targeted rankings as they are often desirable in advanced analysis cases.

Pathfinder also supports multiple scores for each path. Each score is represented by a rank-column displayed on the right of the paths. Figure 6.1, for example, shows two scores: The first score is the length of the path (represented by numbers), and the second score is based on set connection strength (average number of connecting sets, represented by bars). The second score is used to break ties. In this example, the ranking is first driven by the length, then by the connection strength score. An arbitrary number of scores is possible. In the example shown in Figure 6.4, paths are ranked on top, if they contain the selected node *Krzysztof Z. Gajos*, have a short length, and a high average node degree.

Path Comparison Being able to compare and judge the similarity of paths is an important task in path analysis (C VI). The path list view provides a variety of methods



(a) Align pivot node

(b) Align all

Figure 6.5: *Node-alignments that support path comparison. (a) All paths are aligned around a selected pivot node. (b) A layout that strives to put re-occurring nodes at the same horizontal position.*

that enable path comparison. Nodes shared between adjacent paths are connected with a gray line in the background. When a node is contained in a nonadjacent path, we draw a line-stub pointing in its direction. We also support different modes for node alignment to make shared nodes evident. By default, paths are drawn as compactly as possible, from left to right. On demand, paths can be aligned based on a selected pivot node or based on an optimized global alignment, as illustrated in Figure 6.5, which makes it easy to spot re-occurring nodes. A drawback of these layouts is that they are less compact than the default, left-aligned layout.

Pathfinder also introduces methods to compare a selected reference path to other paths. Figure 6.4 shows an example where the second path is the reference path; all other paths show icons next to nodes and sets indicating whether they are shared or not. Shared nodes also indicate their position relative to the same node in the reference path. To make these differences and similarities pop out, we disable the coloring of sets and attributes when a reference path is selected. The reference path can also be considered in path ranking: Instead of absolute scores, each score can be computed as a difference to the reference path. As a result, the reference path will be displayed first, followed by paths with decreasing similarity. Of course, more complex similarity calculations in ranking are possible by providing custom scoring functions.

6.2.3 Path Topology View

The path topology view complements the path list view by showing the topology of the nodes in the paths, which addresses the requirement of judging the topology and its context (C V), and providing an overview. By default, we use a layered layout that makes it easy to trace paths from their start at the left to their end at the right, as illustrated in Figure 6.1. When working with many paths, the topology view suffers from scalability issues. Zooming and panning helps to reveal details, but we can also show only the paths on the active page of the path list. This addresses the scalability issues and still shows the topology for the top-ranked paths.

To judge the path topology in the context of attributes, a user can select individual node attributes to be mapped onto the nodes. However, when there are many homogeneous attributes, an on-node mapping of average and variance similar to the mapping used in EnRoute is also conceivable.

Some usage scenarios demand contextual topological information that goes beyond the topology of the path result set (C V). Pathfinder provides two ways to add additional topological information: First, the links that connect a node to other nodes in the path result set, but are not covered by the paths, can be shown on demand. Second, the user may add all neighbors of a selected node, even if they are not in the set of paths, as shown in Figure 6.6. This can be done repeatedly with neighbors to enable a simple node-by-node graph exploration, which is especially desirable when only querying for start nodes. To better support this approach to graph exploration, Pathfinder also provides a force-directed layout.

6.2.4 Path Statistics View

The path statistics view provides basic statistics and gives an overview of the most important nodes and sets. For paths, nodes, edges, and sets, it displays the total numbers in the result-set, and the numbers considering active filters. To make the most important nodes and sets apparent, we show them in ranked lists. The nodes and sets are ranked by the number of paths of which they are part of, which is displayed in bar charts. Details about all these entities can be accessed by following links to external web-sites (if available), which can be accessed using the context menu.

Figure 6.1 shows an example where *Hanspeter Pfister* and *Ben Shneiderman* are ranked at the top, which is not surprising, as they are the start and end nodes, respectively, and thus occur in all paths. However, the highly ranked authors *Krzysztof Z. Gajos* and *Desney S. Tan* also play a key role in connecting these authors, as they are part of about half of the paths. In this example, the sets are grouped into TVCG and CHI publications, listing the top publications that connect nodes in the paths. Like in the list view, the path statistics view either shows only connecting sets or all sets present in the paths.

6.3 Implementation and Scalability

Pathfinder is implemented as a client-server web-application using Caleydo Web. We use *D3*[2] for the visualization components and D3's force-directed graph layout, which is based on Dwyer's algorithm [44], in the topology view. The layered graph layout is computed using the *dagre* library[3]. Dagre attempts to balance computation performance and quality of results by using a combination of different algorithms [18, 25, 56, 88, 139]. On the back-end, we store the graph data in a *Neo4j* graph database[4]. We use a custom Neo4j plugin written in Java for the k-shortest path search. It uses the built-in Neo4j BFS

[2] `http://d3js.org`

[3] `https://github.com/cpettitt/dagre`

[4] `http://neo4j.org`

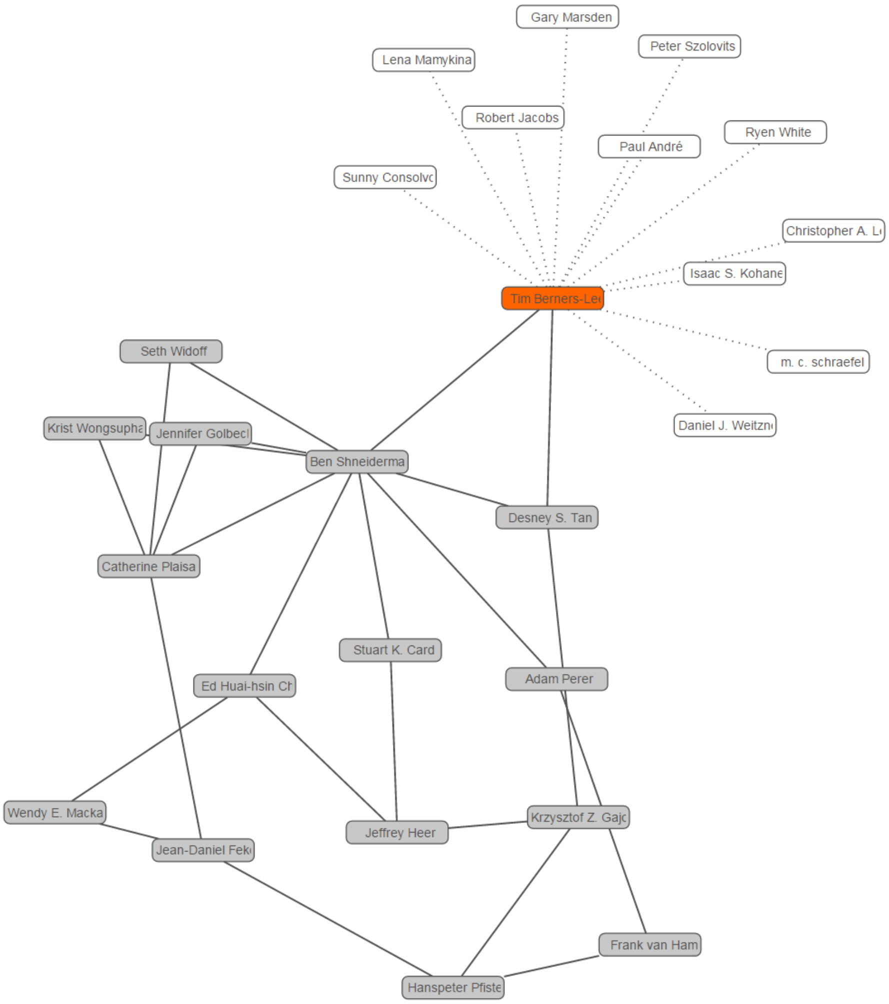


Figure 6.6: *The path topology view showing a force-directed layout to ease neighbor exploration. All neighbors of Tim Berners-Lee were added and are shown in white, indicating that they are not part of any path. Links to those nodes are stippled to distinguish them from links that occur in the path list.*

algorithms, streams intermediate results via WebSocket, and computes virtual edges for set relationships on-the-fly.

We tested Pathfinder for graphs with more than 30,000 nodes and more than 70,000 edges and found no significant scalability problems. Queries on our example datasets typically return results within a few seconds. Query performance is influenced by the server's computing power, but also by caching: Paths through frequently used nodes are retrieved faster. Pathfinder can handle at least 600 paths without problems, with the exception of the overview mode of the topology view, which can become cluttered, as any node-link diagram. The maximum length of paths that can be conveniently shown depends on both screen resolution and the node size configured by the user. On full-HD displays, paths of length 6-12 can be shown conveniently, but longer paths may require scrolling. However, we found that the paths we consider are rarely longer than 6-8 hops. The number of paths that can be shown at the same time in the list view largely depends on the amount of displayed attributes and sets. When all attributes and sets are hidden, up to 20 paths fit on a typical full-HD display. Especially combined with path ranking, which puts the most relevant paths for a specific question at the top of the list, we argue that this number accommodates most tasks.

6.4 Case Study

Up to this point, examples were given mainly on the coauthor network. Here, we also demonstrate Pathfinder's value for an intricate analysis of biological pathways. This case study was conducted by a chemical biologist, who is a researcher at a large pharmaceutical company. Pathfinder was developed with constant feedback from this researcher and other domain experts.

Our collaborator used Pathfinder to analyze biological signaling cascades driving cell proliferation and cancer formation. These signaling cascades consist of proteins that are encoded by genes. For simplicity, we use the terms protein and gene interchangeably. A well-known signaling cascade in biology is the so-called *ERK-MAPK pathway* [137], which transduces signals from the cell membrane to the cell nucleus. In the nucleus, these signals influence gene expression and can induce changes in the cell leading to cell division. Uncontrolled cell division, in turn, causes tumors. The gene *RAS* is the starting point of the *ERK-MAPK pathway*; *RAS* is attached to the cell membrane, where it is switched on by incoming signals. The end point of the signaling pathway is the gene *ERK*, which activates proteins that bind to the DNA and change gene expression. Accordingly, our collaborator queried for paths that connect *KRAS* (one representative member of the *RAS* gene family) and *MAPK3* (a member of the *ERK* family) in Pathfinder. As shown in Figure 6.7, she ranked the 149 paths that were returned by their average set connection strength. She chose this score because she was interested in paths that occur in many KEGG pathway maps. The top-ranked path was *KRAS-RAF1-MAP2K1-MAPK3*, which corresponds exactly to the *ERK-MAPK pathway.* Thus, Pathfinder was able to identify correctly the most important communication path between *RAS* and *ERK*. Expanding the set of associated pathways revealed that this path is present in many cancer pathways: *Colorectal cancer*, *pancreatic cancer*, *glioma*, *prostate cancer*, and *non-small cell lung cancer*, to name just a few. Similarly, the importance of this path was emphasized by the gene expression box plots showing data from cancer cell lines. As illustrated in Figure 6.7, the box plots revealed that these four genes are ubiquitously expressed across diverse tissues, which indicates that this signaling cascade is active in many different cell types.

Another important pathway involved in the formation of cancer is the so-called *mTOR pathway* [107]. Our collaborator used Pathfinder to detect cross-talk between these two pathways, i.e., she explored how one signal transduction pathway could affect the other.

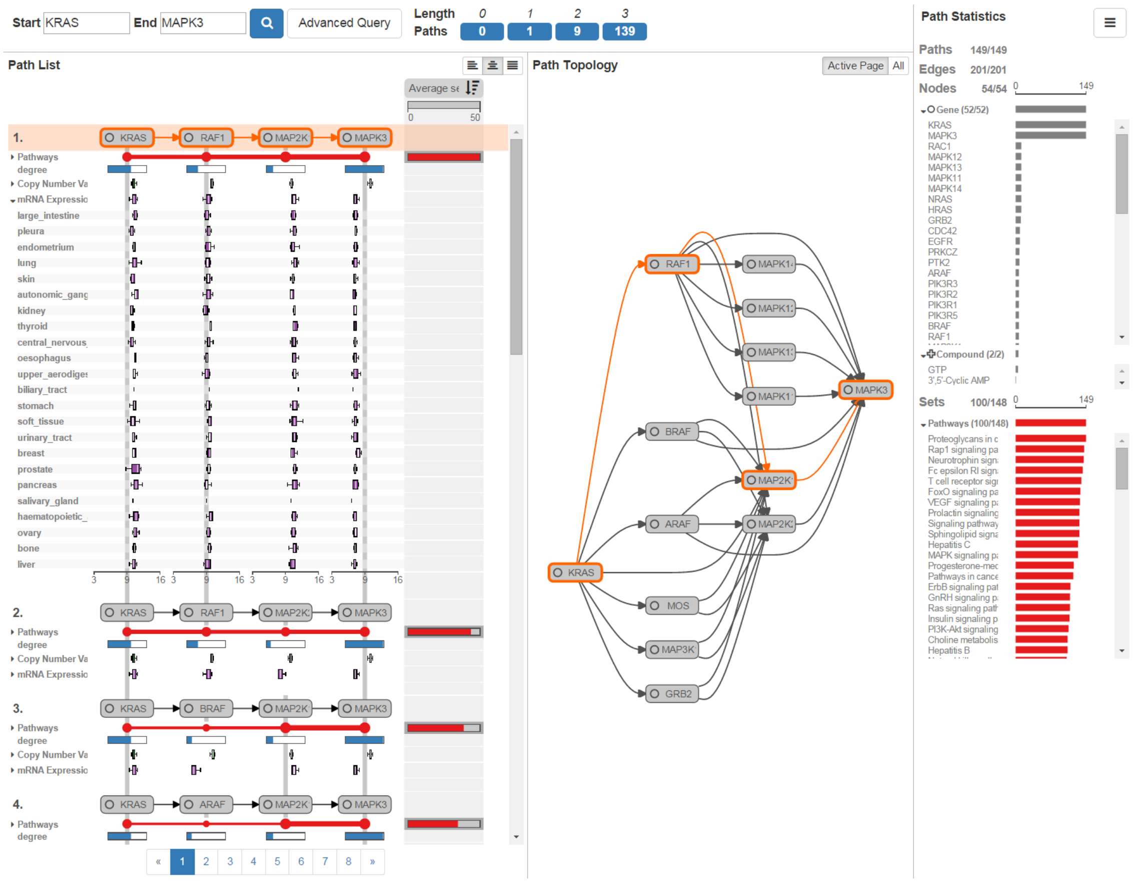


Figure 6.7: *The query result in Pathfinder for paths connecting KRAS and MAPK3. The paths are ranked by set connection strength, which places the path KRAS-RAF1-MAP2K1-MAPK3 on top (notice the thick lines for the aggregated pathways). This path corresponds to the ERK-MAPK signaling cascade. Associated copy number and mRNA expression data are shown as box plots. The expression dataset is expanded to investigate the expression across different tissue types in detail. The box plots show that the four genes are expressed in all displayed tissues, which emphasizes the importance of this path.*

For this purpose, she defined an advanced query, searching for connections between the four genes mentioned above and the five genes *MTOR*, *AKT1*, *TSC1*, *TSC2*, and *MLST8*, which are part of the *mTOR pathway*. She again ranked the paths by their average connection strength, which is illustrated in Figure 6.8. The resulting path topology view highlighted that the *ERK-MAPK pathway* can modulate the *mTOR pathway* through the gene *PI3K*, which in turn modulates *AKT1*. Note that there are different subtypes for this gene, which are all shown in the topology view. Furthermore, the top-ranked path revealed a second route for regulation of the *mTOR pathway* by the *ERK-MAPK pathway*: *MAPK3* can modulate *TSC2*. To learn more about this regulation, our collaborator chose the path *MAP2K1-MAPK3-TSC2* as reference path and sorted all other paths by their similarity to the reference path, as shown in Figure 6.9. This ranking revealed an

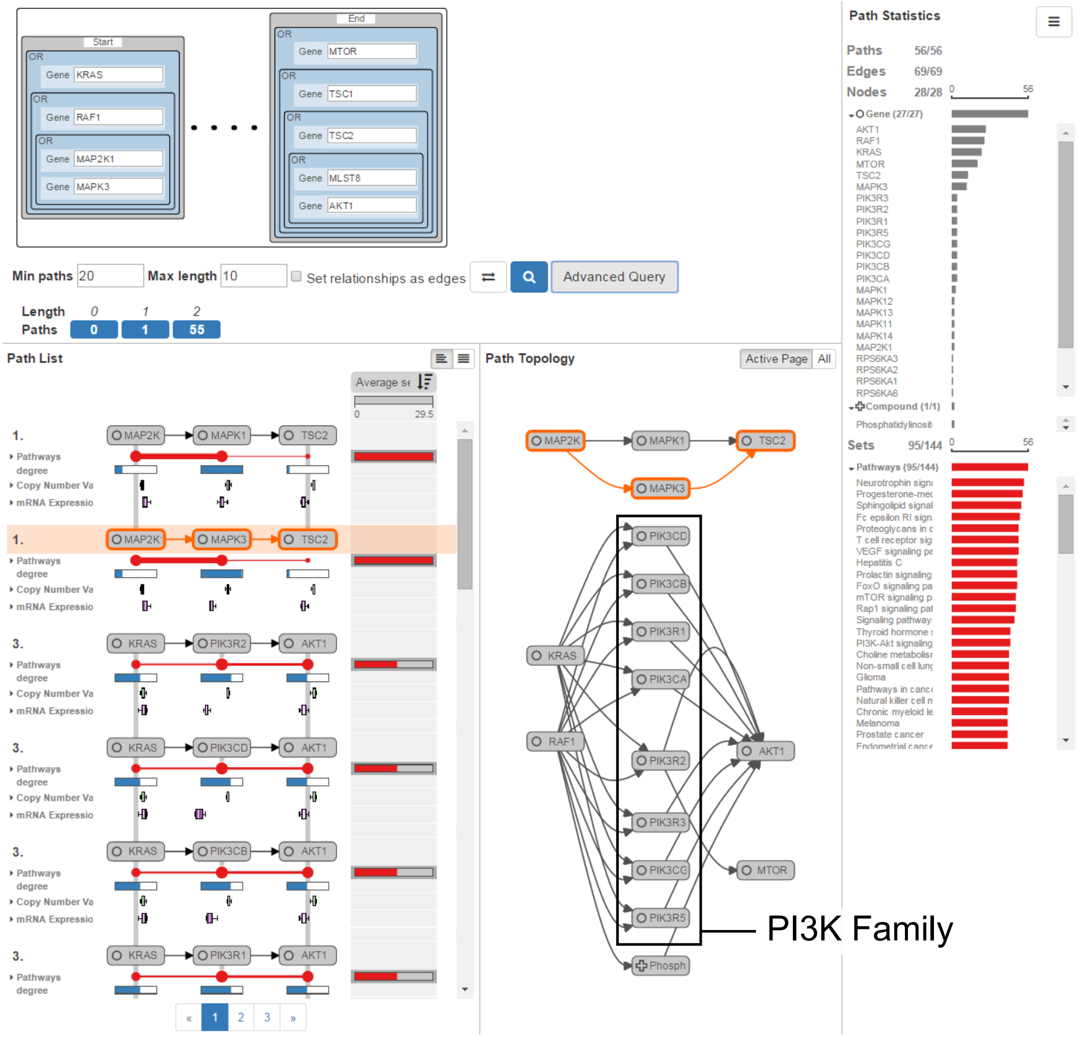


Figure 6.8: *To investigate whether there is any cross-talk between the ERK-MAPK pathway and the mTOR signaling cascade, which are both well-known cancer-related pathways, the analyst defines a complex query between multiple nodes of each path. The query is composed of several OR queries for the start and end, i.e., from (KRAS or RAF1 or MAP2K1 or MAPK3) to (MTOR or TSC1 or TSC2 or MLST8 or AKT1). The paths in the result set are ranked by set connection strength. The topology view reveals that the mTOR pathway can be influenced via the gene PI3K. In this case, the gene is represented by various subtypes (PIK3*). The mTOR pathway, specifically its node TSC2, can also be modulated via MAPK3, as shown by the path highlighted in orange.*

additional, indirect way for *MAPK3* to regulate *TSC2*: via *RPS6K*. Our collaborator was excited that Pathfinder was able to demonstrate the complexity of biological pathway regulation: The *ERK-MAPK pathway* can either modulate the *mTOR pathway* through *PIK3CA* or through *TSC2* [124]. Understanding cross-talk between signaling cascades is important for the development of cancer therapeutics, because they can contribute to drug resistance.

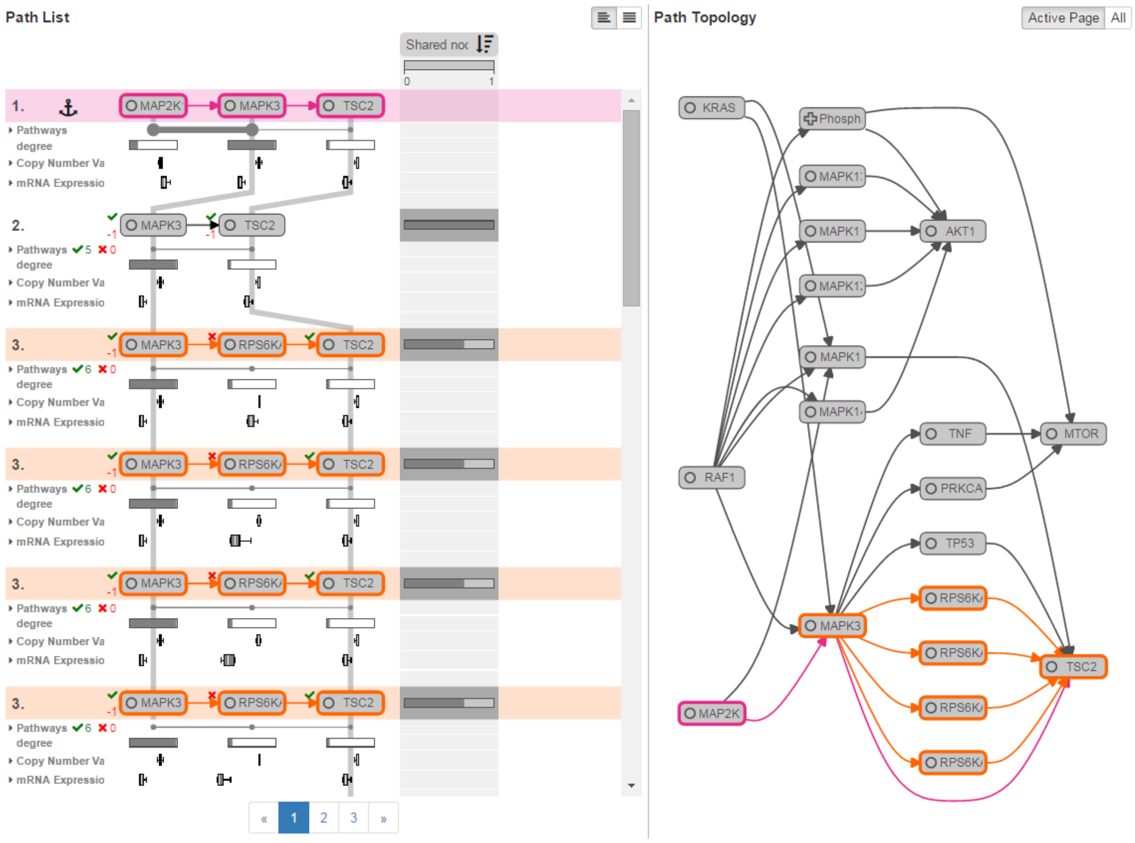


Figure 6.9: *To learn more about paths similar to the MAPK3 path, the analyst selects it as a reference. The list is ranked by the number of shared nodes, which reveals another way to modulate TSC2: via RPS6K. This gene is represented by various subtypes (RPS6K*). The corresponding paths are highlighted in orange.*

In summary, the feedback from our collaborator was very positive. She claimed that the analysis of multiple paths across pathways would have been difficult or even impossible with standard tools such as the KEGG web interface, especially when also considering experimental data.

6.5 Discussion

When developing Pathfinder, we set out to address two major goals: scalability with respect to the number of nodes and edges, and scalability with respect to associated sets and attributes. We deal with them in a number of ways. Foremost, the query-based approach reduces the complexity of analyzing the whole graph to analyzing resulting paths. Of equal importance is the list-based path view, which enables us to show ranked paths and rich attribute and set data that can be aggregated on demand. The computationally expensive and time-consuming path query process is mitigated by showing intermediate results, i.e., a path matching the query is shown as soon as it is found. As the resulting paths can be numerous, easily accessible methods to refine the query help narrow down the result-set, and path ranking helps to quickly identify relevant paths. The ranked list representation of paths scales very well to large numbers of paths. The pagination of the list helps to improve rendering performance and also defines meaningful subsets of the ranked paths to be explored in the topology view.

Besides scalability, a major strength of our approach is that it can be extended and customized to comply with requirements in different use cases without changing the overall workings of the technique. Complex scores can easily be added to support customized path rankings, and the sequential layout of nodes makes adding visualizations of attribute data easy.

In our design, we employ the most effective visual channels for the most important aspects of the data. For example, position, the strongest visual variable, is used to encode path rank, and connectedness is used to encode relationships. Also, all attribute visualizations use a position/size encoding (bar charts, box plots); color is used only as a redundant channel or for highlighting.

In the path list view, some information is duplicated — for example, the start and end nodes, and their associated attributes, occur in every path. Although this wastes space, we argue that the benefits of the linear representations (easy to rank, excellent for attribute visualization) outweigh the cost of non-optimal space usage.

Limitations We distinguish limitations of the technique from limitations of our implementation. Our technique addresses all requirements discussed in Section 6.1 save one: the exploration of path classes (C VII). Aggregating similar paths and showing their basic structure could give analysts a better overview of the variation of paths. Combined with revealing details on demand, this could be an alternative approach to ranking for tackling large lists of paths. Aggregations could also be driven by a user-defined combination of properties, such as topology, attributes, and sets.

Apart from the exploration of path classes, we argue that Pathfinder supports path-related tasks well. A wide range of other tasks discussed by Lee et al. [110] can be addressed by using the topology view and its capability to dynamically extend the network, as illustrated in Figure 6.6. Specifically, the topology view allows us to also support most adjacency, accessibility, common connection, and connectivity tasks, with the exceptions of finding clusters, connected components, bridges, and articulation points, i.e., tasks that are related to an overview of the graph. To support these tasks, we envision integrating Pathfinder with a general purpose, query-based graph visualization system such as Orion [75].

A limitation of the current implementation is the expressiveness of the query interface. For example, attributes of nodes and edges cannot be considered in the query. Also, the visual and interaction design of the advanced query interface could be improved. Another limitation is that we currently do not visualize edge attributes and edge types. The display of both, however, is straightforward, as they can be shown in the same way as node attributes: by adding dedicated rows below the paths in the path list view.

6.6 Summary

In this chapter, we presented Pathfinder, a technique for the visual analysis of multiple paths in large multivariate graphs. The query-based approach allows users to search for paths between a specified start and end. The immediate display of intermediate results allows for early query refinements and speeds up the analysis process. By showing paths in both a list and a node-link diagram, they can be judged and ranked holistically, taking topology, attributes, and grouping structures into account. Again, the linear visualization of paths proved to be useful: While attributes can be easily mapped to nodes and edges like in EnRoute, it also allows for the ranked display of multiple paths in a list. We showcased Pathfinder in context of a coauthor graph and a biological network, but we are confident that this technique can be useful for the analysis of networks from other domains, such as social or computer networks. In summary, Pathfinder is a viable technique for the analysis of multiple paths that is characterized by its iterative approach of querying, ranking, and detailed investigation of paths with respect to both attributes and topology.

Up to this point, our proposed visualization techniques mainly considered networks that feature many heterogeneous attributes, but rather few different types of nodes or items. In the next chapter, we introduce a visualization technique that focuses on the exploration of path relationships between different types of items.

7

Resolving Path Relationships in Linked Datasets

Contents

The need to investigate multiple heterogeneous linked datasets is common in many application domains. Up to now, we regarded heterogeneous datasets mainly as attributes that all map to one specific type of item in a graph; i.e., experimental datasets such as mRNA expression or copy number variation datasets were all mapped to genes. However, in many cases the datasets refer to different types of items. These items often exhibit complex relationships. For example, a drug can modulate multiple proteins, and a protein can be modulated by multiple drugs. Also, a protein is part of multiple pathways, and a pathway contains multiple proteins. Such kind of linked data is typically captured in relational databases. Analyzing path relationships that span across different types of items is essential to get a holistic understanding of all involved data.

In this chapter, we introduce **ConTour**, a visual analysis technique for the exploration of path relationships across heterogeneous datasets. ConTour is designed to tell whether items of different types are connected by paths rather than showing all connecting paths in detail like Pathfinder. In this way, many path relationships can be indicated simultaneously. For simplicity, we will mainly use the term “relationship” instead of path relationship and call path relationships with one hop “direct relationships”, and path relationships with multiple hops “indirect relationships” hereafter.

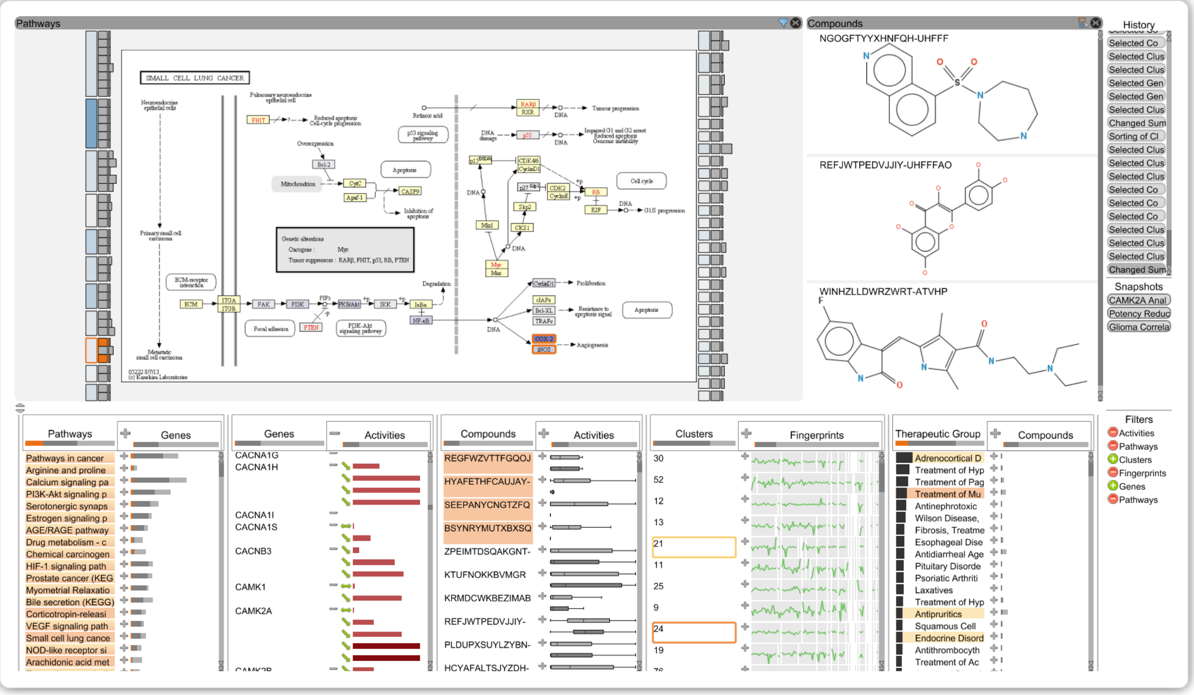


Figure 7.1: *ConTour shows a multitude of heterogeneous data items in several columns in the relationship view (bottom). The detail views display a selected pathway and selected chemical structures of compounds (top).*

Figure 7.1 shows the interface of ConTour: At the bottom, the relationship view lists all items of each type in separate columns. Relationships between items in different columns are revealed through interaction: selecting one or multiple items in one column highlights and re-sorts the items in other columns. Filters based on relationships enable drilling down into the large data space. Additionally, we support the interactive nesting of columns to show the related items of a child column for each item in the parent column. To identify interesting items in the first place, columns can be individually sorted based on attributes or connectivity to items of different types. Details about items can be either investigated within the columns, if enough space is available to show them, or on demand in distinct views at the top, as illustrated in Figure 7.1.

Although the concept of ConTour is domain-agnostic, it was developed in collaboration with domain experts to specifically support pharmaceutical research. Learning about the effects of chemical compounds, i.e., drugs or drug candidates, on cells or the whole organism relies on the generation of huge amounts of data in various experiments and the analysis of this data. Different visualization methods were developed to support structure-activity relationship analyses. For example, HiTSEE [159] helps finding correlations between the structure of chemical compounds and their activity in reactions with a single biological target, and Lounkine et al. [118] classify compounds based on their structure and visualize their interactions with pathway nodes. In addition to the data sources used in classical structure-activity relationship analyses, ConTour integrates various other data in order to provide a more holistic view on the effects that a compound has on a biological network. The overall goal of ConTour is to allow for a visual investigation of the inter-dependencies of all involved data in order to speed up and facilitate the analysis process.

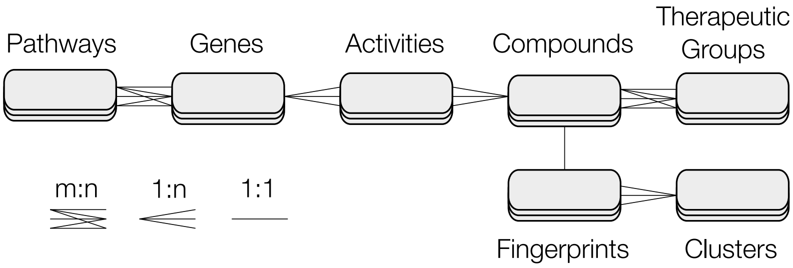


Figure 7.2: *Overview of pharmacological and biological datasets that are analyzed using ConTour. The edges of this graph indicate direct relationships between data items and the cardinality of these relationships. A pathway contains multiple genes and one gene can be part of multiple pathways. Activities connect genes and compounds. Compounds represent drugs, drug candidates, or other small molecules and are classified into therapeutic groups. Fingerprints describe biological properties of compounds. Fingerprint clusters can reveal biologically relevant similarities of compounds.*

As introduced in Section 2.3.3, our collaborators investigate data about chemical compounds, their biological fingerprints, fingerprint clusters, activity data of compounds against their protein targets, proteins, pathways that capture the biological roles of proteins, and therapeutic groups that classify the compounds. Note that proteins are mapped to their corresponding genes and we use the terms interchangeably hereafter. The structure of the available data can be described as a k-partite graph, where sets of items, such as pathways, genes, compounds etc., represent the partitions of the graph. This implies that the items within a set have no defined relationship, but that relationships are defined between items of different sets. As illustrated in Figure 7.2, the graph describing the set relationships is connected and acyclic, i.e., there are no sets that are not related to others, and there is exactly one path connecting any two sets. The relationships between the items of the sets can be of arbitrary cardinality (1:1, 1:n, or n:m). Though this graph only shows *direct* relationships, ConTour is designed to also consider *indirect* relationships via intermediate sets and items. Pathways, for example, are indirectly connected to compounds via genes and activities. Note that we only consider shortest path relationships between items, i.e., each item type must not occur more than once in a path. Otherwise, all items of the graph, with the exception of disconnected components, would be considered as related.

In the following, we first discuss the requirements for the visual analysis of relationships in heterogeneous pharmacological data. We then introduce the ConTour visualization technique on a conceptual level and its realization specifically for the drug discovery use case. Finally, we demonstrate ConTour in three drug discovery case studies.

7.1 Requirement Analysis

In repeated consultations with multiple domain experts over half a year we elicited a set of tasks an analyst has to perform to achieve the domain goals described in Section 2.3.3. From these tasks, we derived more general requirements for the ConTour visualization technique:

D I: Find and show related items. Given an item of type A, find all items of type B that are directly or indirectly related. A concrete example is to identify all pathways that contain a specific gene (direct), or all compounds that influence a pathway (indirect).

D II: Find and show items that share a relationship with a set of items. Given a set of items, find all items that are connected to all of the input items. In other words, identify the items that all of the input items are related to. The input items can be from the same set or from different sets. An example is to identify all genes that are shared between two pathways, or to identify all compounds that are connected to a specific cluster and that are also related to a specific pathway.

D III: Support network enrichment analysis. In highly relational datasets, many items are connected, directly or indirectly, to many others, which can lead to unspecific relationships. Our collaborators, however, are interested to identify the connections that are very specific. For example, they want to identify clusters of compounds where all compounds interact with only one specific pathway. More generally, for items of type A and B that are not directly related, one might want to judge how closely they are connected by considering items in the path between them.

D IV: Rank items. Being able to rank items is crucial to reveal the most important items out of a long list. Rankings can be based on item attributes or on derived measures such as network enrichment.

D V: Filter items. Analysts want to filter items, either based on attribute values or based on relationships. An example for the former is that an analyst might want to only consider activities that activate their interaction partner, and ignore inhibiting or binding drugs. The latter case depends on D I and D II - items that are not related to a specific selection of items should be filtered out.

D VI: Show items in detail. The relevance of data items can often only be judged by exploring their attributes. While some items are simple, such as activities, which only consist of the interaction type and the AC_{50} value, others, such as pathways or compounds, are complex entities. A central task is to investigate these complex entities in detail. Therefore, for example, a pathway should be viewable in all its complexity, or the chemical structure of compounds should be displayable.

7.2 The ConTour Visualization Technique

The tasks that need to be performed by our collaborators describe an analysis process that is highly exploratory in nature, rather than a rigid step-by-step process with well-defined starting, intermediate, and end points. To enable such an analysis for different item sets, a visual analysis technique needs to allow analysts to flexibly gain access to information encoded by items or item relationships at virtually any point during the analysis.

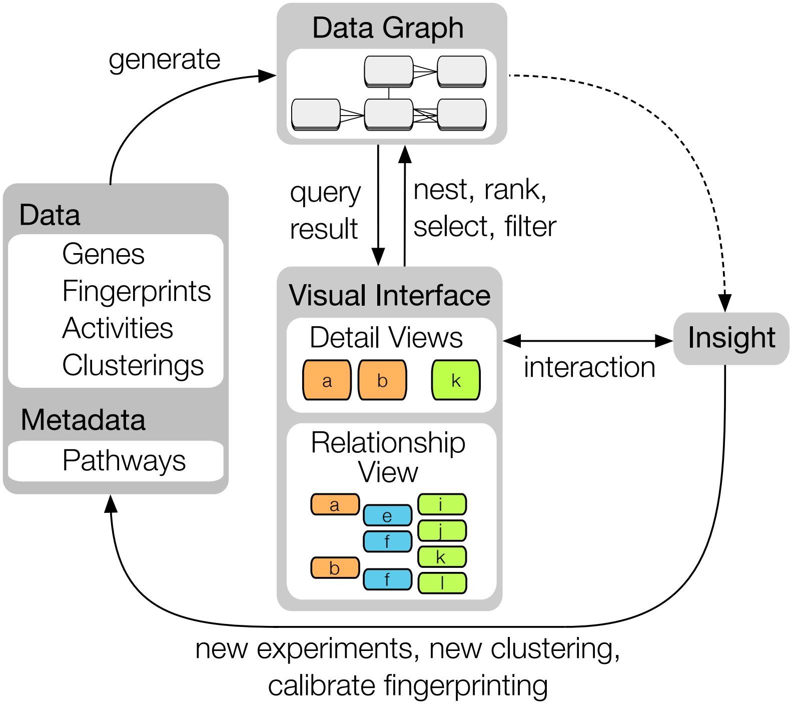


Figure 7.3: *The overall visual analysis process in ConTour. Data from internal or external (public) sources make up the data graph. The data graph is the underlying data structure for the visual interface and responds to its queries (e.g., nesting, ranking, etc.). Through interaction with the visual interface, domain experts generate insights. These insights can be of value on their own, or can lead to refined biochemical experiments, new ideas for clusterings and groupings, or to calibrated fingerprint algorithms, thus generating new data.*

Our approach to this problem is illustrated in Figure 7.3. The **data graph** component contains all data items, their relationships, as well as associated data present in the system. The items of this graph are presented to the user in the **visual interface**. The main component of this visual interface is the *relationship view*, which consists of a collection of columns, each listing the item set of a particular type. A second important component of the visual interface are *detail views*, which display detailed item information using representations specifically tailored to the item type. Based on individual items or whole item sets, several operations, such as selecting, filtering, or nesting, can interactively be triggered from the visual interface. Using graph information of the data structure, these operations are propagated to related items in other item sets, updating their representations in the relationship view and the detail views, e.g., by highlighting, hiding, showing, or reordering items. The tight interplay of the data graph and the visual interface realizes a highly interactive data-driven exploration of item relationships. In the following sections, we will discuss the components of the visual interface in more detail.

7.2.1 Relationship view

The relationship view is composed of several freely arrangeable columns that represent one item set each. Individual columns can be scrolled, sorted, and filtered independently. The layout is designed to enable arbitrary entry points into the analysis: every item in every column can be a starting point. The column's header displays relevant summary information, while the body displays the items. Items show either a label, or relevant attribute information in built-in visualizations, or both. Columns can be added, duplicated, and removed at any time. From a visual point of view, the relationship view is reminiscent of Jigsaw's list view [64, 151] and parallel node-link bands [62]. However, these techniques are restricted to text representations within the columns and explicitly draw links to indicate relationships between items in different columns. We refrain from drawing such links to avoid clutter and save horizontal space. Instead, we provide a rich set of operations that can be performed on individual items or whole columns to explore potentially thousands of items and relationships effectively. These operations are discussed in the following.

Item selection and highlighting. One simple yet effective method to find related items (D I) is highlighting. Selecting an item highlights all of its related items. We distinguish between two selection methods: Hovering over an item just highlights all related items, whereas clicking on an item also moves all related items in all columns to the top. As this reordering might be undesirable in some cases, for instance, if the items of a column use a meaningful sorting, it can be disabled. When selecting multiple items of a type, we employ one of two different modes, which are illustrated in Figure 7.4, to combine the highlights: In **union mode**, all items that are related to any selected item are highlighted. In **intersection mode**, only those items that are related to all selected items are highlighted.

Selection-based filters. Selection-based filters allow to reduce the whole data space to those items that are related to selected items (D I, D V). Applying multiple filters in succession gradually narrows down the data space. In essence, each newly applied filter is combined with the result of all previous filters using a Boolean `and` operation. However, after the data space is narrowed down, it might be desirable to expand it again. Therefore, we provide the possibility to add related items that have previously been filtered out. This additional operation can be regarded as a filter that is combined with the result of all previous filters using a Boolean `or`. The different selection modes (union, intersection)

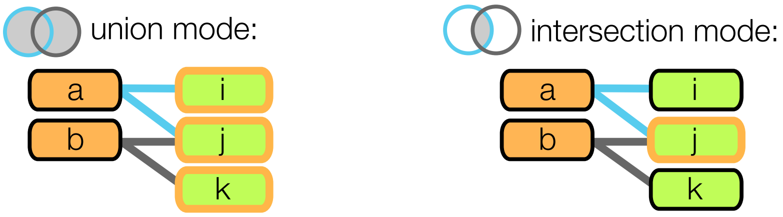


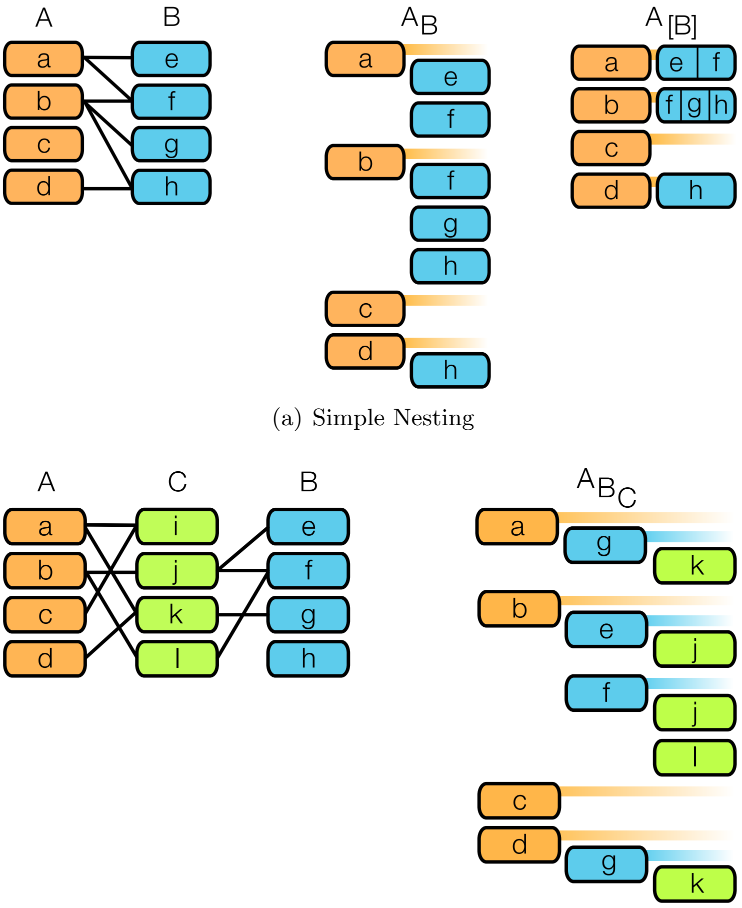
Figure 7.4: *Illustration of the two highlight modes. In union mode, selecting items a and b highlights all items related to either of them, i.e., items i, j, and k. In intersection mode, only items related to both a and b are highlighted, which is item j in this case.*

affect the filtering behavior. The data space gets reduced to the union or intersection of related items, respectively, and the union or intersection of related items is added.

Nesting. Nesting is an effective method to directly associate multiple related items of different sets. Columns can be nested to create parent-child relationships. Nesting two columns has the effect that for each item in the parent column, all related items of the child column are shown right next to it, as shown in Figure 7.5(a). Nesting a gene column within the pathway column, for example, displays all genes that a pathway contains next to the pathways. In contrast to highlighting, nesting always unambiguously shows what items are related, even if multiple items are selected. The downside of nesting is that it is less space efficient, as it results in redundant items. To remedy this, children can be collapsed, so that the relationships are shown only on demand. When child items are collapsed, a summary representation allows the analyst to gain an overview of these items, as shown on the right of Figure 7.5(a). This representation can show a simple count of the children or summary statistics about the children's attributes.

An interesting possibility is to use summary values of the children to sort the parent, opening up new opportunities to identify relevant items. To easily identify items with many relationships, for instance, the number of child items can be used as sorting criterion.

Nesting is a powerful way to investigate direct and indirect relationships between different items (D I), as items of a child column may be intuitively associated with their parents for multiple items of the parent column at once. ConTour also allows to nest multiple columns. Thus, child columns can be siblings or be nested recursively, as illustrated in Figure 7.5(b). However, a recursively nested item is only considered as a child, if it is related to all of its parent items in the chain of parent columns. This makes recursive nesting equivalent to a filter chain applied to the items of the nested columns, with the filters being defined over the relationships to the parent items. Recursive nesting of columns is an effective way for identifying items that are commonly related among items from different sets (D II). For example, in Figure 7.5(b) on the right, item *b* and *f* do have *j* and *l* in common.



(a) Simple Nesting

(b) Recursive Nesting

Figure 7.5: *Nesting. (a) On the left, direct relationships between the items of columns A and B are indicated as connecting lines. In the center, column B is nested in column A, placing related child items of column B right next to their parent item of column A. For example, items e and f are related to item a and are therefore displayed as its children. As item f is also related to b, it is shown next to both parents. On the right, the child items of a and b are collapsed into summary representations. (b) On the left, the direct relationships between items of column A and C and columns B and C are displayed. The items of columns A and B are indirectly related via items of column C. On the right, the columns are recursively nested. Column B is nested in A, and C is nested in B. This recursive nesting helps to find items in C that are commonly related among items in A and B. For example, items a and g are commonly related to item k, whereas items b and f are related to items j and l.*

Ranking and sorting. Ranking and sorting items in a column (D IV) is a simple method to identify the most interesting items quickly. The sorting criteria can be manifold. For example, items can be sorted alphabetically or by some numerical attribute. Rankings can also be based on scores (D III) that quantify certain network properties. Sorted items can easily be compared, if their representation reflects the sorting criterion, such as attribute values.

Column-based filters. Applying filters to columns (D V) is a simple method to remove uninteresting or irrelevant items. Similar to sorting, filters can be based on several criteria. For instance, attribute-based filters may define the value range for numerical attributes or filter items based on associated categories. A simple example is to remove all activities that are above a threshold in their AC_{50} values, which indicates that they are not potent. Filters may be applied locally or globally. **Local filters** only affect the item set of their column. **Global filters** affect all item sets by removing the items that are not connected to one of the remaining items in the source column. Global filters are efficient at reducing the complexity of the whole data space.

7.2.2 Detail views

Triggered from the relationship view, detail views show one or several items using suitable visualizations (D VI). Detail views are tailored to item types. Some may show all items, some a subset of items, and others may show only one item at a time. For example, the detail view for fingerprints shows all of them in one large parallel coordinates view. An example for a detail view that shows only selected items is the compound view, where only the selected compound structures are shown. Detail views can also integrate multiple item types. Our pathway view, for example, shows a pathway together with genes, compounds, and fingerprint clusters.

7.3 Detailed Design

We developed the prototype of ConTour in close cooperation with our collaborators, who gave feedback on a weekly basis. In this section, we describe the design decisions we made to represent the data, which algorithms we implemented to satisfy the analytical needs of our collaborators, and what additional tools we added to support the analysts in the data exploration process.

7.3.1 Relationship View

How we represent the various types of data items in the relationship view mainly depends on the amount of information held by each item. If an item has no additional data associated, we display its name or ID, which is the case for genes, clusters, and therapeutic groups. Also, if there is too much information available to fit in the columns, like in the case of pathways and compounds, we also only show their names or IDs. Although fingerprint items come with over 100 numerical values, it is still possible to visualize them in a compact way. To achieve this, we use centered bar charts with bars pointing up and down, as shown in Figure 7.6(d). This compact representation provides a good overview of the parameters and can be effectively used for comparative tasks. Fingerprints contain many missing values, which we encode by leaving the area empty, while adding a gray background for parameters that have a value. Activity items consist of only two parameters. The first one

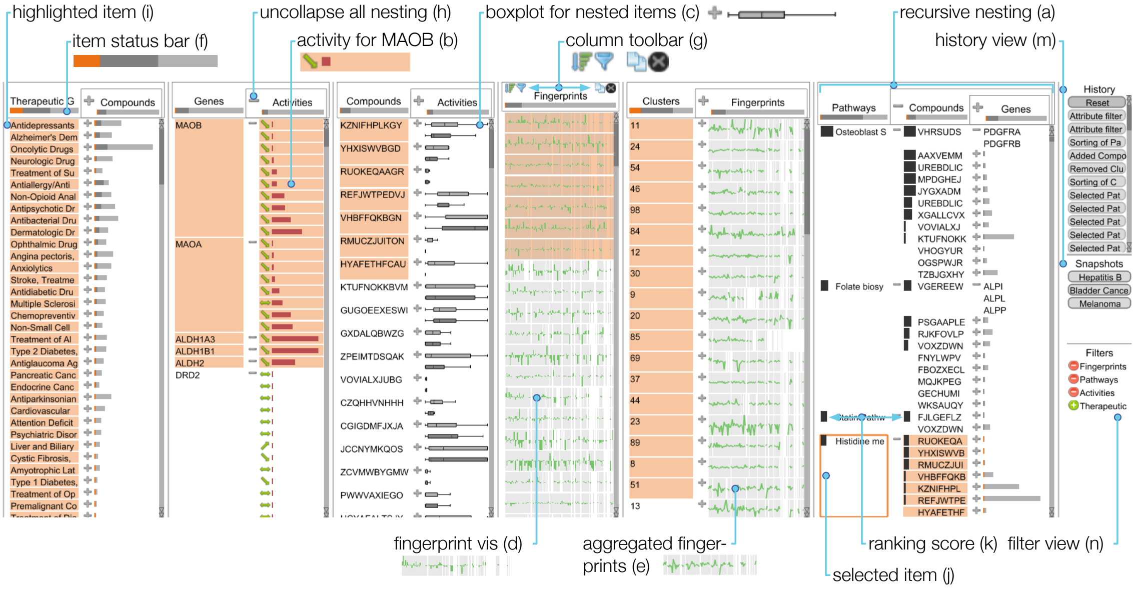


Figure 7.6: *Overview of visual components in the relationship view. Several different data items are listed in columns. (a) These columns can be nested recursively. The visual representations differ depending on the item type. By default items are represented as text. (b) Attributes of activity items are shown with bars and icons. (c) Box plots are used to summarize nested activity items. (d) Numerical fingerprint vectors show their data in bar charts. (e) Line charts summarize the median values of the fingerprints in child columns. (f) Composite bars in each column header show the number of all items (light gray), items not filtered out (dark gray), and selected items (orange) of a column. (g) The column toolbar is shown on demand and contains buttons to sort, filter, duplicate, and remove the column. (h) The items of a child column can be collapsed and expanded using the buttons in the column headers and next to the items. Highlighted items (i) indicate relationships to the selection source (j). (k) Columns that are ranked by enrichment scores represent the scores as bars right next to the items. (m) The history view records every analysis step and also shows snapshots taken. (n) The filter view displays the sequence of currently applied filters.*

is a categorical parameter, which describes the binding type of the associated compound to the associated gene. As shown in Figure 7.6(b), we visualize the three categories, *activation*, *inhibition*, and *binding*, by icons that show an arrow pointing up, down, and a horizontal double arrow, respectively. The second parameter describing the compound activity is the numerical AC_{50} value, which is encoded by a horizontal bar.

Nesting is a crucial concept in ConTour. To summarize nested items, we employ different encodings. The default summary representation that is available for any type of item provides an overview of children by indicating the number of child items using three bars that are drawn on top of each other. The light gray bar indicates the total number of children of a parent item, not considering any filters. The dark gray bar takes filters into account, indicating the number of children that will be shown if the user expands the summary representation. Finally, the orange bar indicates the number of children that are currently highlighted by selection. As illustrated in Figure 7.6(f), the same representation is used to give an overview of all the items in each column. As activity data is tightly coupled with genes and compounds, they are usually nested within these columns. Our collaborators emphasized that it is important to enable analysts to get a quick overview of all activity values associated with a gene or compound. To address this, we provide an additional summary view that encodes the activity value distribution of child items using two box plots, as illustrated in Figure 7.6(c). The box plots drawn in light gray show the distribution for all child items, not considering any filters, whereas the dark gray box plots show the distribution for only those child items that were not filtered out. As clusters are based on fingerprints, they are typically nested within the cluster column. In order to represent the characteristics of a cluster, the fingerprints are aggregated into a line plot that encodes the median values for the fingerprints' parameters, as shown in Figure 7.6(e). We decided to employ a line plot instead of a bar chart, to make the summary representation distinguishable from the representation of individual fingerprints.

7.3.2 Detail Views

As it is not possible to show all data associated with several item types in the relationship view, we provide a number of detail views to make this information accessible. As previously discussed, all detail views are tightly linked with the relationship view and also with each other. Thus, selections or filter operations are propagated to all other views.

Pathway view. The pathway view, shown in Figure 7.1, displays a selected pathway and its contained genes using a texture from one of the supported pathway databases. In addition, the pathway view also displays compounds that interact with at least one of its genes, as well as the fingerprint clusters the compounds are associated with. The pathway view is designed to support two analysis goals: identify which compounds interact with which genes within their cellular context, and identify which compounds and clusters are specific to a pathway. As previously mentioned, specificity is an important quality measure for the domain experts. The more specific a cluster is to a pathway, i.e., the more compounds of the cluster interact with the pathway, the more likely the compounds-pathway interaction is biologically relevant. Clusters are encoded as bars on the left and right of the pathway. The height of the bar encodes how many compounds of the cluster interact with the pathway. Its saturation indicates how specific a cluster is—highly specific clusters are dark blue, while unspecific clusters are white. Next to the clusters, smaller rectangles represent the compounds. By hovering or selecting a compound or cluster, all interaction partners within the pathway are highlighted, enabling analysts to identify the exact binding partners of each compound. The nodes in the pathway are shaded in yellow, if no compound interacts with them. Nodes with a white shading bind to one compound, whereas saturated purple nodes bind to many compounds. The compounds optionally adhere to the system-wide filters, which allows the domain expert to quickly assess the relevance of a pathway for the remaining items.

Compound view. Being able to access the chemical properties of compounds is important to our collaborators, as it provides crucial information when reasoning about, for example, why compounds fall into the same cluster, or why they bind to similar targets. To realize this, the compound view, which is shown in Figure 7.1, displays the chemical structures for multiple compounds together with their names.

Parallel coordinates view. Analysts can make use of the parallel coordinates view to visualize any kind of multi-dimensional data. In the context of the available data, only activity data and fingerprint data fall into this category. As activity data is already displayed in full detail in the relationship view, the parallel coordinates view is mainly used to display fingerprint data. By default the view shows all fingerprints, but can be toggled to respect applied filters.

7.3.3 Support Views

Two support views provide orientation and more flexibility during the data exploration.

History view. Every step taken in the exploration of items and their relationships is based on decisions made by the analyst. However, in some cases, the path taken might lead to a dead end, or the analyst just wants to explore the data in multiple directions without starting the analysis from scratch. As shown in Figure 7.6(m), we provide a history view to address this issue. The history view records every step taken during the analysis and allows the analyst to go back and forth within the analysis path as desired. Each step taken adds a new element to the history view; information about the step is shown on demand as a tooltip. Selecting an element reverts the system to the state when the element was recorded. In addition, analysts can take snapshots of the current state, which they can return to at any time.

Filter view. Filtering is an operation that is executed very frequently. However, keeping track of filters without support is hard. Therefore, ConTour tracks all applied filters in the filter view. As illustrated in Figure 7.6(n), every filter is represented by an element, which displays the name of the item set the filter was applied on. We use two symbols to indicate whether the data space was reduced by the filter, or items were added. A more detailed description of the filters is shown in a tooltip. Filters can be removed from the filter view on demand.

7.3.4 Enrichment Score

One crucial task of our collaborators is judging how specific two types of items are related considering a third item type. For example, they want to know what clusters show an enrichment in compounds that modulate a specific pathway. Abstracted to general set terms, they want to know for an item i of set I (clusters) the enrichment of items of set

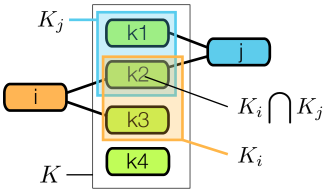


Figure 7.7: *Enrichment for i in j via K: $s_{i,j}(K) = (1/2)/(2/4) = 1$*

K (compounds) that reach one item j in set J (pathways). To answer this question, our collaborators use an enrichment score, which is defined as follows: Let K_i be the set of items in item set K that are related to i and K_j be the set of items in K that are related to j. For a pair of items (i, j), we calculate the enrichment score $s_{i,j}(K)$ by

$$s_{i,j}(K) = \frac{|K_i \bigcap K_j|/|K_j|}{|K_i|/|K|} \tag{7.1}$$

This is also illustrated in Figure 7.7. The numerator of this term describes how specific j is related to i via K. To account for the fact that items in I that are related to many items in K are more likely to also have common items with items in J, we divide by the given denominator. In the end, this score expresses the similarity of sets K_i and K_j and accounts for set sizes and total number of items. Alternative measures, such as overlap disproportionality [9], might also be feasible in this case.

As small overlaps of one or two items were generally not interesting for our collaborators, we include a threshold for the minimum number of common items. The enrichment score is calculated for all pairs (i, j), which in turn can be used to rank columns in the relationship view. However, a column only shows items of a single type, although the score is defined for pairs. Therefore, we use the maximum score, given by $r_{i,K,J} = \max_{\forall j \in J}(s_{i,j}(K))$ to determine the rank of every item i in its column. We display this score as a horizontal bar next to the item. To see the item pairs, the paired columns can be nested, as shown in Figure 7.6. While the parent items show the maximum score, the child items indicate the scores achieved with their parent.

7.4 Implementation and Scalability

Like EnRoute and Entourage, the prototype of the ConTour visualization technique is developed as a plugin for the Caleydo visualization framework. The chemical compound structures are rendered using the *Chemistry Development Kit (CDK)* [152], an open-source Java library for structural chemo- and bioinformatics.

As a tool for exploring multi-relational data, ConTour needs to scale with respect to the number of columns, the number of items inside the columns, the number of nested columns, and the number of detail views it can handle effectively. ConTour can comfortably deal with about a dozen columns. Depending on the kind of data, we observed a limit of about 20 columns on a full-HD display. To even further increase the upper limit for the number of displayable columns, it would be possible to add a level-of-detail approach that lets the user manage larger number of columns. In terms of scalability of items, we have successfully worked with multiple columns containing up to 14,000 items. However, conceptually the column-based approach in combination with our task-dependent sorting of items scales to datasets with many more items. Regarding the recursive nesting of columns, we found that more than four levels of nesting are rarely used in practice. Our current implementation supports about as many levels of recursions as columns for 1:1 relationships between datasets (i.e., about 20), while this number shrinks to about five for the n:m case. Of course, this heavily depends on the amount of items and relationships between them. The number of detail views that can be shown simultaneously depends on the nature and size of the data and the used visualization. For compounds, we observed up to eight simultaneously used views, while pathways were limited to one.

7.5 Case Studies

ConTour is the result of a user-centered development process, which included regular meetings with our collaborators to iteratively develop and refine the system according to their needs. Together with one of our collaborators, who are all chemical biologists, we conducted case studies to illustrate the applicability of ConTour on real-world problems. The overall goal of these case studies was to find out whether the biological fingerprints that were used as descriptors for compounds are able to detect meaningful biological similarities between compounds and reflect their effect on the cell and, ultimately, on the organism as a whole. If they prove to capture a compound's biological actions in a comprehensive manner, they can be used as a connecting module to identify relationships between compounds, targets, pathways, and diseases. In the following, we describe how our collaborators used ConTour to explore heterogeneous pharmaceutical and biological data and report on the gained insights.

7.5.1 Investigating Phosphodiesterase 4 Inhibitors and their Cluster Neighbors

A straightforward way to explore the ability of the used descriptors to group compounds in a biologically meaningful way is to analyze the fingerprints of compounds that are known to modulate the same protein target. Therefore, the expert started by focusing on a particular protein target, the enzyme *phosphodiesterase 4 (PDE4)*, which is represented by multiple different enzyme subtypes (*PDE4A-D*) in the dataset. She added a selection-based filter to limit all displayed items to those related to *PDE4A-D*. By applying an attribute filter to the activity data, our collaborator set an upper threshold for AC_{50} values of one μM, which resulted in ten different compounds that inhibit PDE4. Encouragingly, the fingerprints of four of the ten compounds belonged to cluster 56, prov-

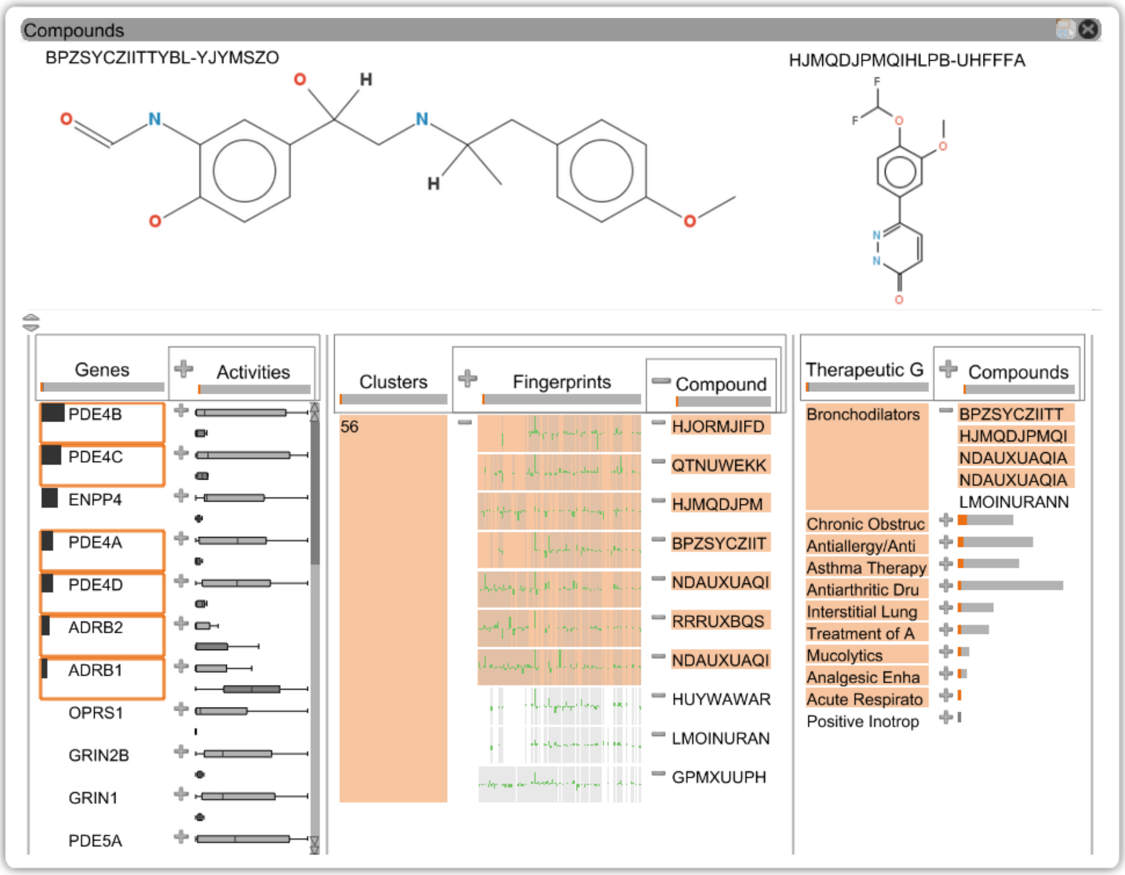


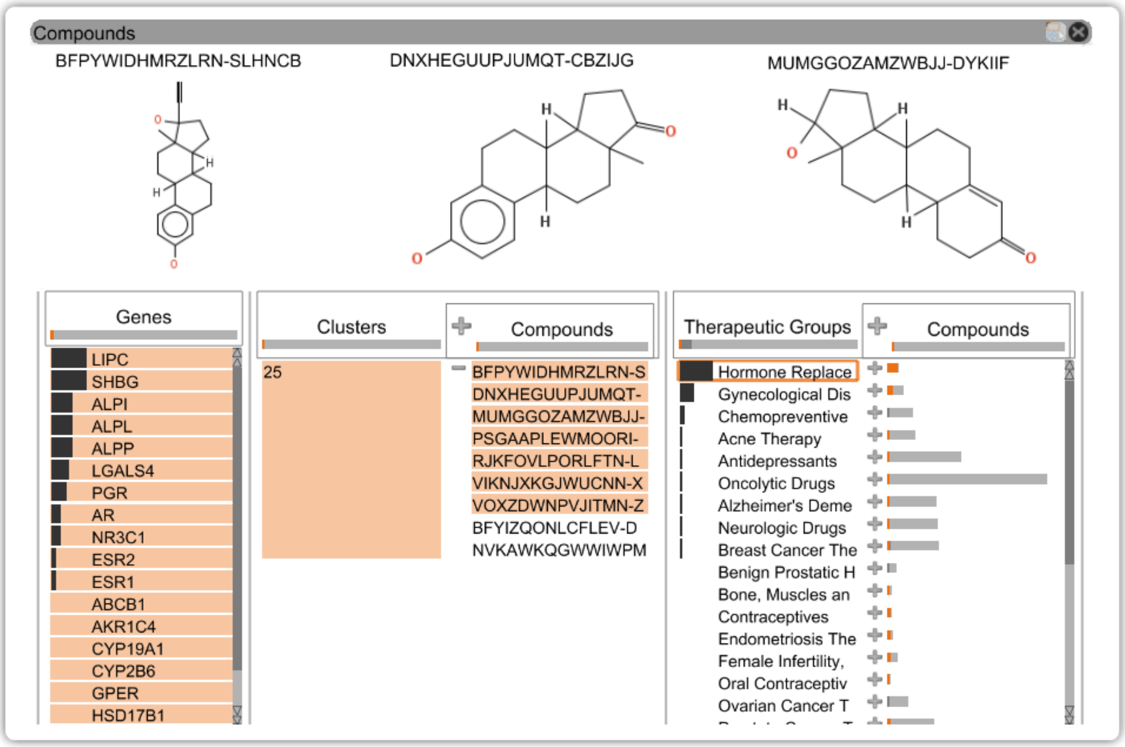
Figure 7.8: *The gene column on the left shows protein targets ranked by the enrichment of Cluster 56. The selected (orange frame) enzyme subtypes of phosphodiesterase 4 (PDE4A-D) and beta-adrenergic receptors (ADRB1, ADRB2) rank among the top. The highlights in the compound column, which is nested within the fingerprint and cluster columns, reveal that compounds binding to these targets fall into the same fingerprint cluster, Cluster 56. Differences in the chemical structures between compounds binding to ADRB1-2 and those binding to PDE4A-D can be seen in the detail view. When looking at the compound column nested within the therapeutic groups column, the highlights indicate that four out of the seven compounds that modulate ADRB or PDE4 map to the group bronchodilators. This group refers to compounds that are used for the treatment of breathing difficulties. Thus, the common bronchodilatory effects of compounds modulating these two target classes might be the reason why their fingerprints cluster together.*

ing that their shared target activity was reflected by similar fingerprint activity patterns. She was then interested what other compounds were found in Cluster 56. Therefore, she added all items related to Cluster 56. Overall, the cluster consisted of ten compounds. By ranking the protein targets by their enrichment of Cluster 56, she learned that two of the newly added compounds bind to *beta-adrenergic receptors (ADRB1, ADRB2)*, which are evolutionary unrelated to *PDE4*. Also, when she displayed the compound structures, she saw that these compounds were structurally very distinct from the *PDE4* inhibitors. At first glance, it seemed surprising that structurally diverse compounds binding to different proteins have similar biological fingerprints and cluster together. However, this observation became better understandable when she integrated the therapeutic group column into the analysis. As shown in Figure 7.8, five compounds from Cluster 56, among them modulators of both *PDE4* and *adrenergic receptors*, are mapped to the group *bronchodilators*, which refers to compounds that are used for the treatment of breathing difficulties. Indeed, it is known that both *phosphodiesterases* and *adrenergic receptors* are protein targets that are involved in airway diseases [1, 136]. Hence, as hoped for by the expert, the biological descriptor, i.e., the clustered fingerprints were able to find therapeutic effect similarities between compounds. In summary, ConTour helped to reveal complex relationships between compounds, their evolutionary unrelated targets, and diseases, and it supported the hypothesis that compounds in Cluster 56 induced therapeutically relevant, similar cellular phenotypes. Based on this finding, it would be straightforward to suggest experiments that test the other compounds in the cluster for their bronchodilatory effects and their use for treating airway diseases.

7.5.2 Exploring Overrepresentation of Compounds with the same Therapeutic Effect

The previous case study has shown that compounds used for the treatment of airway diseases cluster together. To find out whether this observation could also be made for other therapeutic groups, our collaborator defined an enrichment score to systematically calculate the enrichment of therapeutic groups in the fingerprint clusters. By setting a minimum overlap threshold, she constrained the results to only contain therapeutic groups that are related to at least three different compounds in a cluster. The ranking of therapeutic groups by the enrichment score revealed that many different therapeutic groups were overrepresented in individual clusters. For example, seven therapeutic groups showed a strong, more than 20-fold enrichment in at least one cluster. Thus, the observation made for airway diseases was not a singular case, but clustering of compounds that are administered for the same therapeutic indication is generally observed across the data set.

The highest enrichment score (87.9) was obtained for the group *hormone replacement therapy.* Of eight compounds mapping to this therapeutic group, seven fell into Cluster 25 and were grouped together with four more compounds, of which two were annotated for hormone-related therapies. In accordance with the therapeutic role of the compounds, known targets that were enriched in this cluster were *progesterone*, *androgen*, and *estrogen* (sex hormone) receptors. Furthermore, displaying chemical structures revealed that all eleven compounds were chemically very similar steroids, as evident in Figure 7.9. Hence, in this case, the chemical similarity of compounds correlates very well with their biological similarity.



Compounds

BFPYWIDHMRZLRN-SLHNCB

DNXHEGUUPJUMQT-CBZIJG

MUMGGOZAMZWBJJ-DYKIIF

Genes

LIPC
SHBG
ALPI
ALPL
ALPP
LGALS4
PGR
AR
NR3C1
ESR2
ESR1
ABCB1
AKR1C4
CYP19A1
CYP2B6
GPER
HSD17B1

Clusters

25

Compounds

BFPYWIDHMRZLRN-S
DNXHEGUUPJUMQT-
MUMGGOZAMZWBJJ-
PSGAAPLEWMOORI-
RJKFOVLPORLFTN-L
VIKNJXKGJWUCNN-X
VOXZDWNPVJITMN-Z
BFYIZQONLCFLEV-D
NVKAWKQGWWIWPM

Therapeutic Groups

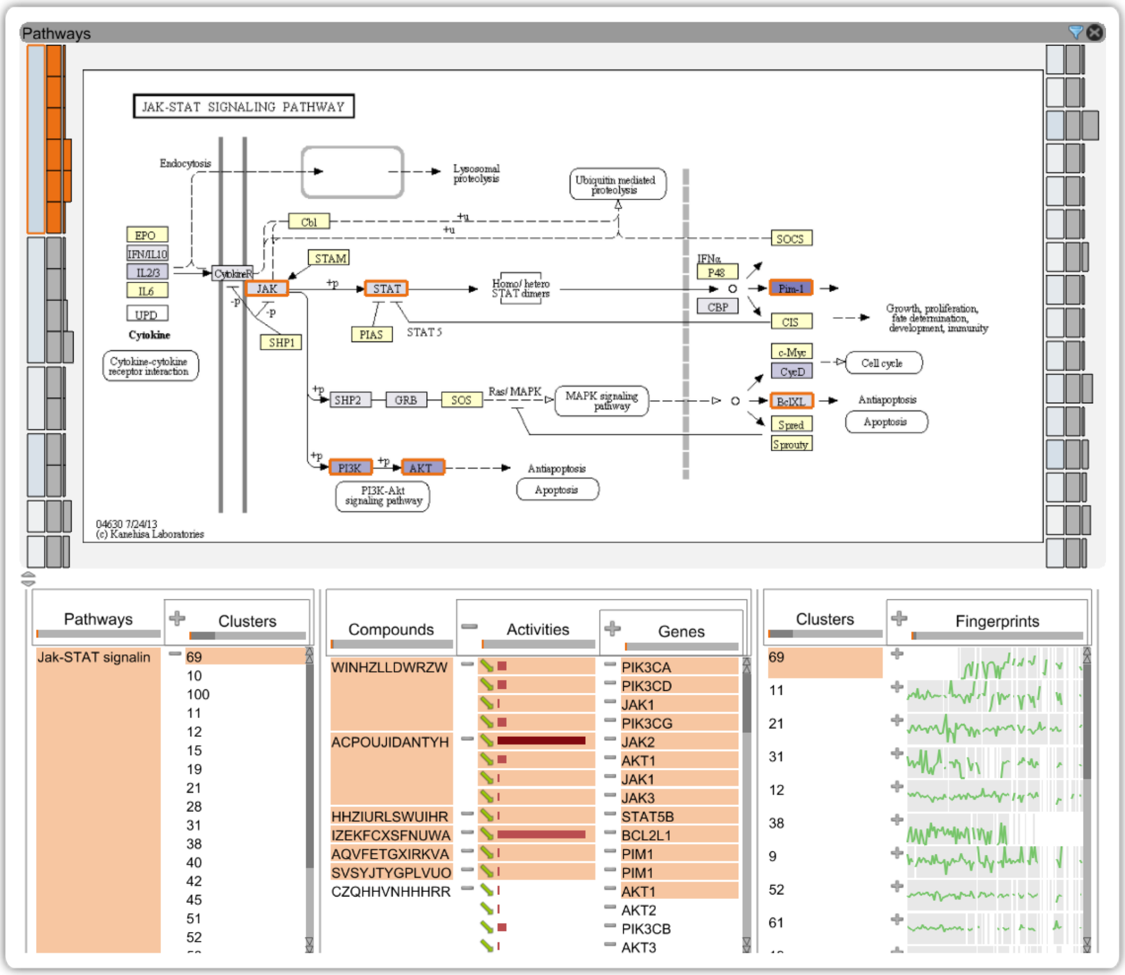
Hormone Replace
Gynecological Dis
Chemopreventive
Acne Therapy
Antidepressants
Oncolytic Drugs
Alzheimer's Deme
Neurologic Drugs
Breast Cancer The
Benign Prostatic H
Bone, Muscles an
Contraceptives
Endometriosis The
Female Infertility,
Oral Contraceptiv
Ovarian Cancer T

Compounds

Figure 7.9: *To find out whether similarities in the fingerprints of compounds correlated with the compounds' classification in therapeutic groups, an enrichment score was used to calculate the enrichment of therapeutic groups in the fingerprint clusters. The highest score was achieved by Cluster 25 and the group hormone replacement therapy (selected item with orange frame), which was thus ranked on top of the therapeutic group column. As indicated by the highlight, this group and Cluster 25 have 7 compounds in common. The detail view of these compounds reveals that they are also structurally similar.*

7.5.3 Explaining Cluster Compositions by Pathways

Finally, our collaborator wanted to find out whether compounds modulating different targets in the same biological pathway had similar biological fingerprints and thus cluster together. This could be expected considering that all targets in a pathway work together and that modulations of different targets in the same signaling cascade of a pathway should be propagated, ultimately leading to similar cellular phenotypes. Our collaborator started the analysis by selecting the important *JAK-STAT signaling pathway* shown in Figure 7.10. This pathway is evolutionary conserved across many different species and plays a role in the regulation of the immune system. When she opened the pathway in the detail view, she found four clusters that contained at least two compounds mapping onto the pathway. Interestingly, for all four clusters, it could be observed that their compounds modulated different targets in the pathway. The most prominent example was Cluster 69, in which six of overall 17 compounds in the cluster modulated one or more targets in the pathway. Of the six compounds, four inhibited different members of a protein family known as kinases, i.e., *Janus kinases (JAKs)* and the *Pim-1 kinase*, in the pathway. The other two compounds modulated the pathway by inhibiting the signal transducer *STAT* and the protein *BCL* that regulates transcription, i.e., gene expression. The collaborator confirmed that, without pathway information, clustering of compounds with diverse target activities is often difficult to reconcile. However, ConTour makes it possible to explain similar biological fingerprints of compounds with diverse target activities by showing that their targets act in a concerted manner in the cell by contributing to the same signaling event.



Pathways	Clusters
Jak-STAT signalin	69
	10
	100
	11
	12
	15
	19
	21
	28
	31
	38
	40
	42
	45
	51
	52

Compounds	Activities	Genes
WINHZLLDWRZW		PIK3CA
		PIK3CD
		JAK1
		PIK3CG
ACPOUJIDANTYH		JAK2
		AKT1
		JAK1
		JAK3
HHZIURLSWUIHR		STAT5B
IZEKFCXSFNUWA		BCL2L1
AQVFETGXIRKVA		PIM1
SVSYJTYGPLVUO		PIM1
CZQHHVNHHHRR		AKT1
		AKT2
		PIK3CB
		AKT3

Clusters	Fingerprints
69	
11	
21	
31	
12	
38	
9	
52	
61	

Figure 7.10: *The detail view shows the JAK-STAT signaling pathway. Selecting the block of Cluster 69 on the top left shows that the compounds of this cluster map to several different targets (graph nodes highlighted with an orange frame) in the pathway. The selection also highlights the genes and compounds in the relationship view. The (recursive) nesting of the gene column within the compound column displays the exact compound-target mappings and reveals that the compounds bind to different targets. This observation suggests that despite the compounds binding to different targets, their fingerprints clustered together, because their targets are part of the same pathway.*

7.6 Discussion

By using ConTour, our collaborators gained several insights that support the hypothesis that biological fingerprints indeed reflect similarities among compounds and their biological effects on both the target protein and pathway level. By observing our collaborator during the analyses, we found that she picked up ConTour's concepts quickly. She used selections to identify relationships of individual items and filters to focus on the data of interest. The expert excessively used nesting, as she argued it helped understanding relationships of multiple items much better when she had to identify relationships across columns. To our surprise, she used nesting not in a static "set up once" approach, but constantly refined, removed, and added nestings to answer specific questions. We observed that she continuously used a combination of reasoning based on visualization and refinement using analytical processes and queries. For example, she relied on ranking by various scores and filtering to identify interesting items, but then refined her queries and adjusted her next steps based on the visual representations.

The combination of the query-driven relationship view and the various detail views proved highly valuable to our collaborators. Especially the compound detail view was frequently used to reason about whether observations made for compounds may be caused by their chemical properties, but also the pathway view was employed to contextualize the findings. In summary, the case studies confirm that the interplay of ConTour's building blocks is an effective approach for exploring data relationships in drug discovery.

7.7 Summary

In this chapter, we introduced ConTour, a visual analysis technique designed to facilitate the exploration of path relationships in large cohorts of biological and pharmacological data. The main interface of ConTour displays items of diverse datasets in a simple, yet effective column-based layout. We indicate the presence of path relationships between items of different sets through interactive selection and highlighting, filtering, and careful alignment of items in nested columns. As we do not display the whole paths, large amounts of relationships between multiple item sets can be indicated simultaneously. However, whole path relationships may also be resolved, for example, by repeated nesting or filtering. Ranking and sorting of items allow for quick identification of relevant relationships and items, which can be investigated in detail in tightly interlinked views. Case studies in drug discovery confirmed the practical applicability of our technique: Using ConTour, our collaborators detected correlations between fingerprint clusters and therapeutic groups and were able to explain the composition of fingerprint clusters by common targets in pathways. These findings are valuable indicators in support of our collaborators' hypothesis that biological fingerprints can be used to detect meaningful compound similarities and that fingerprints also reflect the effect of compounds on the cell or organism.

Although we have designed ConTour specifically for the requirements in drug discovery, the underlying concept is very generic, such that the approach can be applied to both other biological and non-biological domains, if the data is structured similarly. Besides customizing the representation of items accordingly, providing an overview of how sets are related, for example, in the form of entity-relationship diagrams [30], might be necessary when users are not as intrinsically familiar with the data structure as our collaborators.

In conclusion, ConTour represents an effective visualization technique for the analysis of heterogeneous linked data. Similarly to Pathfinder, the main concept to deal with large amounts of related data items is iterative querying (through selection, filtering, and nesting) ranking of items, and detailed inspection of individual items on demand.

8 Conclusion and Future Directions

The analysis of graphs and linked data is a long-studied, but rich research domain in visualization, which is relevant to many application domains. In molecular biology, large amounts of heterogeneous data are generated and analyzed to get an understanding of biological processes in cells or whole organisms, which may ultimately lead to, for example, the development of more effective drugs, more accurate diagnoses, and more targeted therapies. From a visualization point of view the main challenges to be faced are the large amounts of data, their complex relationships, and data heterogeneity.

In this thesis, we contributed four visualization techniques for the analysis of paths and the path-based exploration of graphs. These techniques were primarily developed to support data analysis in molecular biology. More specifically, with EnRoute, we enable the investigation of pathways in context of large amounts of heterogeneous genomic data, which helps reasoning about observed effects in this data. Entourage supports the simultaneous analysis of multiple pathways and their relationships, which is particularly useful for judging side-effects of drugs or finding potential for drug repositioning. Pathfinder allows for the detailed analysis of relationships between biomolecules by supporting the inspection of multiple alternative paths with associated genomic data. With ConTour, we support uncovering relationships in a network of diverse types of biomolecular and pharmacological data to identify the mechanism of action of drugs and their therapeutic indications. Our visualization techniques were evaluated in case studies that demonstrated how they were used by domain experts for data analysis. We chose case studies to evaluate our techniques, as they are particularly suitable to determine how well visualizations support visual data analysis and reasoning, and frequently used in visualization research [85, 105].

When regarding our proposed visualization techniques from a domain-independent point of view, we make two primary observations:

1. We have shown that paths can effectively be used as subsets of graphs to achieve scalability in graph visualization. With EnRoute, we address scalability and also heterogeneity with respect to graph attributes. We enable users to interactively select individual paths in a node-link diagram, extract them, and visualize them

side-by-side with large amounts of heterogeneous attributes. With Entourage, we address scalability in terms of graph size. We introduced the concept of contextual subsets, which allows us to simultaneously show multiple partitions of a graph by reducing them to contextually relevant paths. We believe that paths are especially well-suited for sub-setting due to two main reasons: First, paths are highly relevant in graph analysis, as many tasks that are related to connectivity can be broken down to the analysis of paths. Second, due to the sequential nature of paths, they can be laid out flexibly. Especially the linear layout proved to be advantageous, as it is very space-efficient when aligned horizontally or vertically, and allows for intuitive juxtaposition of node and edge attributes. Additionally, as illustrated in Pathfinder, linear path representations can be efficiently displayed in a list, which in turn can be reordered to reflect path rankings.

2. We have shown that iterative querying, ranking, and detailed investigation is a viable strategy for the visual exploration of multiple paths and data items connected by path relationships. This strategy was successfully applied in both Pathfinder and ConTour. Pathfinder supports the analysis of whole paths in graphs with rather few different node types, whereas ConTour indicates only the presence of path relationships between data items, but for many different types of items at the same time. The key to quickly identify relevant information in both approaches is to enable users to flexibly express their interest by means of user-driven querying, which reduces the whole data space to a smaller, but potentially still large result set, and user-driven ranking in order to identify the most important elements in this result set. The visualization of details on demand allows users to further judge the results and plan next analysis steps. Overall, this exploration strategy can be regarded as an instance of Keim's Visual Analytics Mantra [95]: Users *analyze first* by formulating an initial query. Initial results *show the important.* Repeated application of ranking and query refinement allow users to *zoom, filter, and analyze further.* Eventually, we reveal *details on demand.*

In summary, our visualization techniques proved to be valuable for the analysis of complex biomolecular data. However, as the underlying concepts of these techniques are generic, we argue that they can be equally applied in many other domains to deal with large partitioned graphs, large amounts of attributes, and to conduct comprehensive analyses of multiple paths.

Future Directions In the future, we consider to further combine our visualization techniques to widen the coverage of analysis tasks. Besides the already discussed combination of EnRoute and Entourage, Pathfinder might be combined with Entourage as well: Entourage could serve as query interface for Pathfinder, where start- and end nodes can be specified in the node-link diagrams. Furthermore, Entourage could also serve as pathway detail view for ConTour that is capable of showing multiple pathways simultaneously.

When it comes to improving individual techniques, we believe that Pathfinder would benefit from an overview of similar paths. Pathfinder currently gives an overview of all paths in the topology view and also provides various methods to judge the similarity of individual paths. The identification of similar paths would be greatly alleviated by providing an overview with respect to path similarity that can be parameterized with different distance measures. Such an overview could be generated by clustering paths into similar groups, which show a summary representation of contained paths in the path list that can be expanded on demand. Alternatively, dimensionality reduction techniques such as multidimensional scaling could be used to reduce paths to points in a scatterplot, similar to the approach by van den Elzen [169]. Similar paths would then be shown as points that are in close proximity, whereas points of dissimilar paths are further apart.

The current realization of ConTour focuses on the exploration of relationships between items of different types. Considering connections between items of the same type would severely increase the analytical capabilities of ConTour. For example, the judgment of indirect effects of drugs can be facilitated by including relationships among their binding partners, i.e., proteins.

Although the exploration of paths is an important part of graph analysis, many tasks cannot be performed by only considering paths. For example, branches or backward links may influence signaling cascades in biological networks, but are hard to cover by path-centric exploration, like in Pathfinder. Consequently, it is desirable to complement Pathfinder with methods that allow users to query for more complex graph structures that are eventually added to the topology view. Such structures might also be considered as contextual subsets of pathways in Entourage. As the display of these structures as node-link diagrams is less space efficient than paths, an alternative display of focus and context pathways, where they are merged into a single node-link diagram, might be conceivable. Approaches like Extended Line Sets [129] or other set visualization techniques could be adopted to identify individual pathways. Especially with the availability of algorithms that produce layouts reminiscent of manually curated pathway maps [100], we believe that interactive (sub-)pathway merging could be interesting research direction.

Finally, we intend to integrate statistical analysis methods. Although humans are good at recognizing patterns in visualizations, judging whether an observation is an actual effect in the data or just an artifact due to chance is difficult. Tight integration of statistics in the visual analysis process allows users to immediately quantify their observations and thus gives them confidence about their findings.

Bibliography

[1] Abbott-Banner, K. H. and Page, C. P. (2014). Dual PDE3/4 and PDE4 inhibitors: Novel treatments for COPD and other inflammatory airway diseases. *Basic & Clinical Pharmacology & Toxicology*, 114(5):365–376. (page 120)

[2] Abello, J., Hadlak, S., Schumann, H., and Schulz, H.-J. (2014). A modular degree-of-interest specification for the visual analysis of large dynamic networks. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 20(3):337–350. (page 25)

[3] Adar, E. (2006). GUESS: A language and interface for graph exploration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*, pages 791–800, New York, NY, USA. ACM. (page 22)

[4] Advani, A. S. and Pendergast, A. M. (2002). Bcr–Abl variants: Biological and clinical aspects. *Leukemia Research*, 26(8):713–720. (page 80)

[5] Agrawala, M. and Stolte, C. (2001). Rendering effective route maps: Improving usability through generalization. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*, pages 241–249, New York, NY, USA. ACM. (page 35)

[6] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2007). *Molecular Biology of the Cell*. Garland Science, 5th edition. (page 7, 8)

[7] Albrecht, M., Kerren, A., Klein, K., Kohlbacher, O., Mutzel, P., Paul, W., Schreiber, F., and Wybrow, M. (2010). On open problems in biological network visualization. In Eppstein, D. and Gansner, E., editors, *Graph Drawing*, volume 5849 of *Lecture Notes in Computer Science*, pages 256–267. Springer Berlin Heidelberg. (page 8, 27, 30)

[8] Aleman-Meza, B., Halaschek-Weiner, C., Arpinar, I., Ramakrishnan, C., and Sheth, A. (2005). Ranking complex relationships on the semantic web. *IEEE Internet Computing*, 9(3):37–44. (page 38)

[9] Alsallakh, B., Aigner, W., Miksch, S., and Hauser, H. (2013). Radial Sets: Interactive visual analysis of large overlapping sets. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 19(12):2496–2505. (page 117)

[10] Amar, R., Eagan, J., and Stasko, J. (2005). Low-level components of analytic activity in information visualization. In *Proceedings of the IEEE Symposium on Information Visualization*, InfoVis '05, pages 111–117, Minneapolis, MN, USA. IEEE. (page 18)

[11] Anyanwu, K., Maduko, A., and Sheth, A. (2005). Semrank: Ranking complex relationship search results on the semantic web. In *Proceedings of the 14th International Conference on World Wide Web (WWW '05)*, pages 117–127, New York, NY, USA. ACM. (page 38)

[12] Ashburn, T. T. and Thor, K. B. (2004). Drug repositioning: Identifying and developing new uses for existing drugs. *Nature Reviews Drug Discovery*, 3(8):673–683. (page 12)

[13] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000). Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25(1):25–29. (page 25)

[14] Auber, D. (2004). Tulip: A huge graph visualization framework. In Jünger, M. and Mutzel, P., editors, *Graph Drawing Software*, Mathematics and Visualization, pages 105–126. Springer Berlin Heidelberg. (page 22)

[15] Baldonado, M. Q. W., Woodruff, A., and Kuchinsky, A. (2000). Guidelines for using multiple views in information visualization. In *Proceedings of the ACM Conference on Advanced Visual Interfaces (AVI '00)*, pages 110–119. ACM. (page 22, 23)

[16] Barretina, J., Caponigro, G., Stransky, N., Venkatesan, K., Margolin, A. A., Kim, S., Wilson, C. J., Lehár, J., Kryukov, G. V., Sonkin, D., Reddy, A., Liu, M., Murray, L., Berger, M. F., Monahan, J. E., Morais, P., Meltzer, J., Korejwa, A., Jané-Valbuena, J., Mapa, F. A., Thibault, J., Bric-Furlong, E., Raman, P., Shipway, A., Engels, I. H., Cheng, J., Yu, G. K., Yu, J., Aspesi, P., de Silva, M., Jagtap, K., Jones, M. D., Wang, L., Hatton, C., Palescandolo, E., Gupta, S., Mahan, S., Sougnez, C., Onofrio, R. C., Liefeld, T., MacConaill, L., Winckler, W., Reich, M., Li, N., Mesirov, J. P., Gabriel, S. B., Getz, G., Ardlie, K., Chan, V., Myer, V. E., Weber, B. L., Porter, J., Warmuth, M., Finan, P., Harris, J. L., Meyerson, M., Golub, T. R., Morrissey, M. P., Sellers, W. R., Schlegel, R., and Garraway, L. A. (2012). The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603–607. (page 13, 15)

[17] Barsky, A., Munzner, T., Gardy, J., and Kincaid, R. (2008). Cerebral: Visualizing multiple experimental conditions on a graph with biological context. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '08)*, 14(6):1253–1260. (page 28)

[18] Barth, W., Mutzel, P., and Jünger, M. (2004). Simple and efficient bilayer cross counting. *Journal of Graph Algorithms and Applications*, 8(2):179–194. (page 96)

[19] Bastian, M., Heymann, S., and Jacomy, M. (2009). Gephi: An open source software for exploring and manipulating networks. In *Third International AAAI Conference on Weblogs and Social Media*. (page 36, 41)

[20] Behrisch, M., Bach, B., Henry Riche, N., Schreck, T., and Fekete, J.-D. (2016). Matrix reordering methods for table and network visualization. *Computer Graphics Forum*, 35(3):693–716. (page 20)

[21] Bellman, R. (1958). On a routing problem. *Quarterly of Applied Mathematics*, 16(1):87–90. (page 35, 53)

[22] Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping Multidimensional Data*, pages 25–71. Springer, Berlin, Heidelberg. (page 22)

[23] Bertin, J. (2010). *Semiology of Graphics: Diagrams, Networks, Maps*. ESRI Press, Redlands, CA, USA. First published in French in 1967. (page 20)

[24] Bezerianos, A., Chevalier, F., Dragicevic, P., Elmqvist, N., and Fekete, J. D. (2010). GraphDice: A system for exploring multivariate social networks. *Computer Graphics Forum (EuroVis '10)*, 29(3):863–872. (page 28)

[25] Brandes, U. and Köpf, B. (2001). Fast and simple horizontal coordinate assignment. In Mutzel, P., Jünger, M., and Leipert, S., editors, *Graph Drawing*, number 2265 in Lecture Notes in Computer Science, pages 31–44. Springer Berlin Heidelberg. (page 96)

[26] Brandes, U. and Wagner, D. (1998). Dynamic grid embedding with few bends and changes. In *Algorithms and Computation*, Lecture Notes in Computer Science, pages 90–99. Springer Berlin Heidelberg. (page 20)

[27] Brewer, C. A. (2009). Colorbrewer. last accessed Jun. 03, 2018. (page 42)

[28] Cantley, L. C. and Neel, B. G. (1999). New insights into tumor suppression: PTEN suppresses tumor formation by restraining the phosphoinositide 3-Kinase/AKT pathway. *Proceedings of the National Academy of Sciences*, 96(8):4240–4245. (page 12)

[29] Carr, D. B., Littlefield, R. J., and Nichloson, W. L. (1986). Scatterplot matrix techniques for large N. In *Proceedings of the Symposium on the Interface of Computer Sciences and Statistics*, pages 297–306. Elsevier North-Holland. (page 26)

[30] Chen, P. P. S. (1976). The entity-relationship model toward a unified view of data. *ACM Transactions on Database Systems (TODS '76)*, 1(1):9–36. (page 124)

[31] Cheng, G., Zhang, Y., and Qu, Y. (2014). Explass: Exploring associations between entities via top-k ontological patterns and facets. In *Proceedings of the 13th International Semantic Web Conference - Part II (ISWC '14)*, pages 422–437, New York, NY, USA. Springer. (page 38)

[32] Chernoff, H. (1973). The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68(342):361–368. (page 26)

[33] Cimikowski, R. (2006). An analysis of some linear graph layout heuristics. *Journal of Heuristics*, 12(3):143–153. (page 19)

[34] Collins, C., Penn, G., and Carpendale, S. (2009a). Bubble Sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '09)*, 15(6):1009–1016. (page 51)

[35] Collins, C., Viegas, F., and Wattenberg, M. (2009b). Parallel Tag Clouds to explore and analyze faceted text corpora. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST '09)*, pages 91–98. IEEE. (page 73)

[36] Delling, D., Gemsa, A., Nöllenburg, M., and Pajor, T. (2010). Path schematization for route sketches. In Kaplan, H., editor, *Algorithm Theory - SWAT 2010*, number 6139 in Lecture Notes in Computer Science, pages 285–296. Springer Berlin Heidelberg. (page 35)

[37] Di Battista, G., Eades, P., Tamassia, R., and Tollis, I. (1999). *Graph drawing: Algorithms for the visualization of graphs.* Prentice Hall, Upper Saddle River N.J. (page 19)

[38] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271. (page 35)

[39] Dinkla, K. and Westenberg, M. A. (2012). Network visualization in cell biology. *Tsinghua Science and Technology*, 17(4):365–382. (page 8)

[40] Dix, A. and Ellis, G. (2002). By chance: Enhancing interaction with large data sets through statistical sampling. In *Proceedings of the ACM Conference on Advanced Visual Interfaces (AVI '02)*, pages 167–176. ACM. (page 22)

[41] Dork, M., Riche, N., Ramos, G., and Dumais, S. (2012). PivotPaths: Strolling through faceted information spaces. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '12)*, 18(12):2709–2718. (page 23, 24)

[42] Duncan, C. A. and Goodrich, M. T. (2013). Planar orthogonal and polyline drawing algorithms. In Tamassia, R., editor, *Handbook of Graph Drawing and Visualization*, pages 223–246. CRC Press, Boca Raton, FL, USA. (page 20)

[43] Dunne, C., Henry Riche, N., Lee, B., Metoyer, R., and Robertson, G. (2012). GraphTrail: Analyzing large multivariate, heterogeneous networks while supporting exploration history. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, pages 1663–1672. ACM. (page 24)

[44] Dwyer, T. (2009). Scalable, versatile and simple constrained graph layout. *Computer Graphics Forum*, 28(3):991–998. (page 20, 96)

[45] Dwyer, T., Marriott, K., Schreiber, F., Stuckey, P., Woodward, M., and Wybrow, M. (2008). Exploration of networks using overview+detail with constraint-based cooperative layout. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1293–1300. (page 22)

[46] Eades, P. (1984). A heuristic for graph drawing. In *Congressus Numerantium*, volume 42, pages 149–160. (page 19)

[47] Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences USA*, 95(25):14863–14868. (page 26, 42)

[48] Eppstein, D. (1999). Finding the k shortest paths. *SIAM Journal on Computing*, 28(2):652–673. (page 35)

[49] Fabregat, A., Sidiropoulos, K., Garapati, P., Gillespie, M., Hausmann, K., Haw, R., Jassal, B., Jupe, S., Korninger, F., McKay, S., Matthews, L., May, B., Milacic, M., Rothfels, K., Shamovsky, V., Webber, M., Weiser, J., Williams, M., Wu, G., Stein, L., Hermjakob, H., and D'Eustachio, P. (2016). The reactome pathway knowledgebase. *Nucleic Acids Research*, 44(D1):D481–D487. (page 9, 25)

[50] Ferreira de Oliveira, M. C. and Levkowitz, H. (2003). From visual data exploration to visual data mining: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):378–394. (page 21)

[51] Fisher, D., Popov, I., Drucker, S., and schraefel, m. (2012). Trust me, i'm partially right: Incremental visualization lets analysts explore large datasets faster. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, pages 1673–1682, New York, NY, USA. ACM. (page 91)

[52] Floyd, R. W. (1962). Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345. (page 35)

[53] Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3):75–174. (page 22)

[54] Freeman, J. L., Perry, G. H., Feuk, L., Redon, R., McCarroll, S. A., Altshuler, D. M., Aburatani, H., Jones, K. W., Tyler-Smith, C., Hurles, M. E., Carter, N. P., Scherer, S. W., and Lee, C. (2006). Copy number variation: New insights in genome diversity. *Genome Research*, 16(8):949 –961. (page 8)

[55] Furnas, G. W. (1986). Generalized fisheye views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '86)*, pages 16–23. ACM. (page 25)

[56] Gansner, E., Koutsofios, E., North, S., and Vo, K.-P. (1993). A technique for drawing directed graphs. *IEEE Transactions on Software Engineering*, 19(3):214–230. (page 96)

[57] Gansner, E. R. and North, S. C. (2000). An open graph visualization system and its applications to software engineering. *Software - Practice and Experience*, 30(11):1203–1233. (page 41)

[58] Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B., and Overington, J. P. (2012). ChEMBL: A large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–1107. (page 15)

[59] Gehlenborg, N., O'Donoghue, S. I., Baliga, N. S., Goesmann, A., Hibbs, M. A., Kitano, H., Kohlbacher, O., Neuweger, H., Schneider, R., Tenenbaum, D., and Gavin, A.-C. (2010). Visualization of omics data for systems biology. *Nature Methods*, 7(3):56–68. (page 27)

[60] Gene Ontology Consortium (2015). Gene Ontology Consortium: Going forward. *Nucleic Acids Research*, 43(D1):D1049–1056. (page 25)

[61] Gerasch, A., Kaufmann, M., and Kohlbacher, O. (2014). Rebuilding KEGG maps: Algorithms and benefits. In *2014 IEEE Pacific Visualization Symposium (PacificVis '14)*, pages 97–104. IEEE. (page 34)

[62] Ghani, S., Kwon, B. C., Lee, S., Yi, J. S., and Elmqvist, N. (2013). Visual analytics for multimodal social network analysis: A design study with social scientists. *IEEE Transactions on Visualization and Computer Graphics (VAST '13)*, 19(12):2032–2041. (page 41, 110)

[63] Ghoniem, M., Fekete, J.-D., and Castagliola, P. (2005). On the readability of graphs using node-link and matrix-based representations: A controlled experiment and statistical analysis. *Information Visualization*, 4(2):114 –135. (page 20)

[64] Görg, C., Tipney, H., Verspoor, K., Jr, W. A. B., Cohen, K. B., Stasko, J., and Hunter, L. E. (2010). Visualization and language processing for supporting analysis across the biomedical literature. In Setchi, R., Jordanov, I., Howlett, R. J., and Jain, L. C., editors, *Knowledge-Based and Intelligent Information and Engineering Systems*, number 6279 in Lecture Notes in Computer Science, pages 420–429. Springer. (page 40, 110)

[65] Gratzl, S., Gehlenborg, N., Lex, A., Strobelt, H., Partl, C., and Streit, M. (2015). Caleydo Web: An integrated visual analysis platform for biomedical data. In *Poster Compendium of the IEEE Conference on Information Visualization (InfoVis '15)*, Chicago, IL, USA. IEEE. (page 42)

[66] Gratzl, S., Lex, A., Gehlenborg, N., Pfister, H., and Streit, M. (2013). LineUp: Visual analysis of multi-attribute rankings. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 19(12):2277–2286. (page 42, 90)

[67] Greilich, M., Burch, M., and Diehl, S. (2009). Visualizing the evolution of compound digraphs with TimeArcTrees. *Computer Graphics Forum*, 28(3):975–982. (page 36)

[68] Haag, F., Lohmann, S., Bold, S., and Ertl, T. (2014). Visual SPARQL querying based on extended filter/flow graphs. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI '14)*, pages 305–312, New York, NY, USA. ACM. (page 23, 25)

[69] Hadlak, S., Schulz, H. J., and Schumann, H. (2011). In situ exploration of large dynamic networks. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '11)*, 17(12):2334–2343. (page 22)

[70] Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107. (page 35)

[71] Hartmann, J. T., Haap, M., Kopp, H.-G., and Lipp, H.-P. (2009). Tyrosine kinase inhibitors - a review on pharmacology, metabolism and side effects. *Current drug metabolism*, 10(5):470–481. (page 81)

[72] Healey, C. G. (1996). Choosing effective colours for data visualization. In *Proceedings of the IEEE Conference on Visualization (Vis '96)*, pages 263–270. IEEE. (page 37)

[73] Healy, P. and Nikolov, N. S. (2013). Hierarchical drawing algorithms. In Tamassia, R., editor, *Handbook of Graph Drawing and Visualization*, pages 409–453. CRC Press, Boca Raton, FL, USA. (page 19)

[74] Hearst, M., Elliott, A., English, J., Sinha, R., Swearingen, K., and Yee, K.-P. (2002). Finding the flow in web site search. *Communications of the ACM*, 45(9):42–49. (page 24)

[75] Heer, J. and Perer, A. (2014). Orion: A system for modeling, transformation and visualization of multidimensional heterogeneous networks. *Information Visualization*, 13(2):111–133. (page 22, 102)

[76] Heer, J. and Robertson, G. G. (2007). Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '07)*, 13(6):1240–1247. (page 76)

[77] Heim, P., Ertl, T., and Ziegler, J. (2010). Facet Graphs: Complex semantic querying made easy. In *The Semantic Web: Research and Applications*, Lecture Notes in Computer Science, pages 288–302. Springer Berlin Heidelberg. (page 24)

[78] Heim, P., Hellmann, S., Lehmann, J., Lohmann, S., and Stegemann, T. (2009). RelFinder: Revealing relationships in RDF knowledge bases. In Chua, T.-S., Kompatsiaris, Y., Mérialdo, B., Haas, W., Thallinger, G., and Bailer, W., editors, *Semantic Multimedia*, number 5887 in Lecture Notes in Computer Science, pages 182–187. Springer Berlin Heidelberg. (page 37)

[79] Henry, N. and Fekete, J.-D. (2007). MatLink: Enhanced matrix visualization for analyzing social networks. In Baranauskas, C., Palanque, P., Abascal, J., and Barbosa, S. D. J., editors, *Human-Computer Interaction – INTERACT 2007*, number 4663 in Lecture Notes in Computer Science, pages 288–302. Springer Berlin Heidelberg. (page 39)

[80] Henry, N., Fekete, J. D., and McGuffin, M. J. (2007). NodeTrix: A hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '07)*, 13(6):1302–1309. (page 22)

[81] Hoffmann, R., Baudisch, P., and Weld, D. S. (2008). Evaluating visual cues for window switching on large screens. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*, pages 929–938. ACM. (page 51)

[82] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441. (page 21)

[83] Inselberg, A. and Dimsdale, B. (1990). Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proceedings of the IEEE Conference on Visualization (Vis '90)*, pages 361–378. IEEE. (page 26, 42)

[84] Isenberg, P., Heimerl, F., Koch, S., Isenberg, T., Xu, P., Stolper, C., Sedlmair, M., Chen, J., Möller, T., and Stasko, J. (2015). Visualization publication dataset. http://vispubdata.org. (page 91)

[85] Isenberg, T., Isenberg, P., Chen, J., Sedlmair, M., and Moller, T. (2013). A systematic review on the practice of evaluating visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2818–2827. (page 125)

[86] Jankun-Kelly, T. J., Dwyer, T., Holten, D., Hurter, C., Nöllenburg, M., Weaver, C., and Xu, K. (2014). Scalability considerations for multivariate graph visualization. In Kerren, A., Purchase, H. C., and Ward, M. O., editors, *Multivariate Network Visualization*, number 8380 in Lecture Notes in Computer Science, pages 207–235. Springer. (page 21, 22, 23)

[87] Johnson, B. and Shneiderman, B. (1991). Tree-Maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the IEEE Conference on Visualization (Vis '91)*, pages 284–291. (page 21)

[88] Jünger, M. and Mutzel, P. (1997). 2-Layer straightline crossing minimization: Performance of exact and heuristic algorithms. *Journal of Graph Algorithms and Applications*, 1(1):1–25. (page 96)

[89] Junker, B. H., Klukas, C., and Schreiber, F. (2006). VANTED: A system for advanced data analysis and visualization in the context of biological networks. *BMC Bioinformatics*, 7(1):109. (page 27)

[90] Jusufi, I. (2013). *Multivariate Networks: Visualization and Interaction Techniques.* PhD thesis, Linnaeus University, Växjö, Sweden. (page 26)

[91] Jusufi, I., Klukas, C., Kerren, A., and Schreiber, F. (2012). Guiding the interactive exploration of metabolic pathway interconnections. *Information Visualization*, 11(2):136–150. (page 34)

[92] Kamada, T. and Kawai, S. (1989). An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15. (page 19)

[93] Kanehisa, M., Goto, S., Sato, Y., Furumichi, M., and Tanabe, M. (2012). KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Research*, 40(1):D109–114. (page 9, 15, 25, 31, 32)

[94] Kashofer, K., Tschernatsch, M. M., Mischinger, H. J., Iberer, F., and Zatloukal, K. (2009). The disease relevance of human hepatocellular xenograft models: Molecular characterization and review of the literature. *Cancer Letters*, 286(1):121–128. (page 15)

[95] Keim, D. A., Mansmann, F., Schneidewind, J., and Ziegler, H. (2006). Challenges in visual data analysis. In *Proceedings of the Conference on Information Visualisation (IV '06)*, pages 9–14. IEEE. (page 126)

[96] Kelder, T., Iersel, M. P. v., Hanspers, K., Kutmon, M., Conklin, B. R., Evelo, C. T., and Pico, A. R. (2011). WikiPathways: Building research communities on biological pathways. *Nucleic Acids Research*, 40(D1):D1301–D1307. (page 9, 15)

[97] Keller, R., Eckert, C. M., and Clarkson, P. J. (2006). Matrices or node-link diagrams: Which visual representation is better for visualising connectivity models? *Information Visualization*, 5(1):62 –76. (page 20)

[98] Kerren, A., Purchase, H. C., and Ward, M. O. (2014). Introduction to multivariate network visualization. In Kerren, A., Purchase, H. C., and Ward, M. O., editors, *Multivariate Network Visualization*, number 8380 in Lecture Notes in Computer Science, pages 1–9. Springer. (page 19, 93)

[99] Kerzner, E., Lex, A., Sigulinsky, C., Urness, T., Jones, B., Marc, R., and Meyer, M. (2017). Graffinity: Visualizing connectivity in large graphs. *Computer Graphics Forum (EuroVis '17)*, 36(3):251–260. (page 39)

[100] Kieffer, S., Dwyer, T., Marriott, K., and Wybrow, M. (2016). HOLA: Human-like orthogonal network layout. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):349–358. (page 30, 31, 32, 127)

[101] Klukas, C. and Schreiber, F. (2006). Dynamic exploration and editing of KEGG pathway diagrams. *Bioinformatics*, 23(3):344–350. (page 33, 34)

[102] Kobourov, S. G. (2013). Force-directed drawing algorithms. In Tamassia, R., editor, *Handbook of Graph Drawing and Visualization*, pages 383–408. CRC Press, Boca Raton, FL, USA. (page 19)

[103] Kono, N., Arakawa, K., Ogawa, R., Kido, N., Oshita, K., Ikegami, K., Tamaki, S., and Tomita, M. (2009). Pathway Projector: Web-based zoomable pathway browser using KEGG Atlas and Google Maps API. *PLoS ONE*, 4(11):e7710. (page 31)

[104] Kruskal, J. B. and Landwehr, J. M. (1983). Icicle plots: Better displays for hierarchical clustering. *The American Statistician*, 37(2):162. (page 21)

[105] Lam, H., Bertini, E., Isenberg, P., Plaisant, C., and Carpendale, S. (2012). Empirical studies in information visualization: Seven scenarios. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1520–1536. (page 125)

[106] Lambert, A., Dubois, J., and Bourqui, R. (2011). Pathway preserving representation of metabolic networks. *Computer Graphics Forum (EuroVis '11)*, 30(3):1021–1030. (page 30, 31, 32)

[107] Laplante, M. and Sabatini, D. M. (2012). mTOR signaling in growth control and disease. *Cell*, 149(2):274–293. (page 98)

[108] Law, V., Knox, C., Djoumbou, Y., Jewison, T., Guo, A. C., Liu, Y., Maciejewski, A., Arndt, D., Wilson, M., Neveu, V., Tang, A., Gabriel, G., Ly, C., Adamjee, S., Dame, Z. T., Han, B., Zhou, Y., and Wishart, D. S. (2014). Drugbank 4.0: Shedding new light on drug metabolism. *Nucleic acids research*, 42(D1):D1091–1097. (page 15)

[109] Lee, B., Parr, C., Plaisant, C., Bederson, B., Veksler, V., Gray, W., and Kotfila, C. (2006a). TreePlus: Interactive exploration of networks with enhanced tree layouts. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1414–1426. (page 25)

[110] Lee, B., Plaisant, C., Parr, C. S., Fekete, J.-D., and Henry, N. (2006b). Task taxonomy for graph visualization. In *Proceedings of the AVI Workshop on BEyond time and errors: novel evaluation methods for information visualization (BELIV '06)*, pages 1–5. ACM. (page 18, 47, 89, 102)

[111] Lex, A., Partl, C., Kalkofen, D., Streit, M., Gratzl, S., Wassermann, A. M., Schmalstieg, D., and Pfister, H. (2013). Entourage: Visualizing relationships between biological pathways using contextual subsets. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 19(12):2536–2545. (page 6)

[112] Lex, A., Schulz, H.-J., Streit, M., Partl, C., and Schmalstieg, D. (2011). VisBricks: Multiform visualization of large, inhomogeneous data. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '11)*, 17(12):2291–2300. (page 23, 54)

[113] Lex, A., Streit, M., Kruijff, E., and Schmalstieg, D. (2010). Caleydo: Design and evaluation of a visual analysis framework for gene expression data in its biological context. In *Proceedings of the IEEE Symposium on Pacific Visualization (PacificVis '10)*, pages 57–64. IEEE. (page 33)

[114] Lex, A., Streit, M., Schulz, H.-J., Partl, C., Schmalstieg, D., Park, P. J., and Gehlenborg, N. (2012). StratomeX: Visual analysis of large-scale heterogeneous genomics data for cancer subtype characterization. *Computer Graphics Forum (EuroVis '12)*, 31(3):1175–1184. (page 29, 42)

[115] Lieberman, M. D., Taheri, S., Guo, H., Mir-Rashed, F., Yahav, I., Aris, A., and Shneiderman, B. (2010). Visual exploration across biomedical databases. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(2):536–550. (page 41)

[116] Lindroos, H. and Andersson, S. G. E. (2002). Visualizing metabolic pathways: Comparative genomics and expression analysis. *Proceedings of the IEEE*, 90(11):1793–1802. (page 27)

[117] Liu, Z., Navathe, S., and Stasko, J. (2011). Network-based visual analysis of tabular data. In *IEEE Conference on Visual Analytics Science and Technology (VAST '11)*, pages 41–50. IEEE. (page 22)

[118] Lounkine, E., Kutchukian, P., Petrone, P., Davies, J. W., and Glick, M. (2012). Chemotography for multi-target SAR analysis in the context of biological pathways. *Bioorganic & Medicinal Chemistry*, 20(18):5416–5427. (page 106)

[119] Luger, S., Stitz, H., Gratzl, S., Gehlenborg, N., and Streit, M. (2015). Interactive visualization of provenance graphs for reproducible biomedical research. In *Poster Compendium of the IEEE Conference on Information Visualization (InfoVis '15)*. IEEE. (page 25)

[120] Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2):110–141. (page 56)

[121] Marriott, K., Purchase, H., Wybrow, M., and Goncu, C. (2012). Memorability of visual features in network diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2477–2485. (page 20)

[122] Martins, R. M., Andery, G. F., Heberle, H., Paulovich, F. V., Lopes, A. d. A., Pedrini, H., and Minghim, R. (2012). Multidimensional projections for visual analysis of social networks. *Journal of Computer Science and Technology*, 27(4):791–810. (page 28)

[123] McGill, R., Tukey, J. W., and Larsen, W. A. (1978). Variations of box plots. *The American Statistician*, 32(1):12–16. (page 57)

[124] Mendoza, M. C., Er, E. E., and Blenis, J. (2011). The Ras-ERK and PI3K-mTOR pathways: Cross-talk and compensation. *Trends in biochemical sciences*, 36(6):320–328. (page 100)

[125] Mestres, J., Gregori-Puigjané, E., Valverde, S., and Solé, R. V. (2008). Data completeness—the achilles heel of drug-target networks. *Nature Biotechnology*, 26(9):983–984. (page 13)

[126] Meyer, M., Wong, B., Styczynski, M., Munzner, T., and Pfister, H. (2010). Pathline: A tool for comparative functional genomics. *Computer Graphics Forum (EuroVis '10)*, 29(3):1043–1052. (page 29, 30, 49, 56, 72)

[127] Mlecnik, B., Scheideler, M., Hackl, H., Hartler, J., Sanchez-Cabo, F., and Trajanoski, Z. (2005). PathwayExplorer: Web service for visualizing high-throughput expression data on biological pathways. *Nucleic Acids Research*, 33(Suppl 2):W633–W637. (page 27)

[128] Olston, C. and Woodruff, A. (2000). Getting portals to behave. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '00)*, pages 15–25. IEEE Computer Society Press. (page 22)

[129] Paduano, F. and Forbes, A. G. (2015). Extended LineSets: A visualization technique for the interactive inspection of biological pathways. *BMC Proceedings*, 9(6):S4. (page 33, 127)

[130] Partl, C., Gratzl, S., Streit, M., Wassermann, A. M., Pfister, H., Schmalstieg, D., and Lex, A. (2016). Pathfinder: Visual analysis of paths in graphs. *Computer Graphics Forum (EuroVis '16)*, 35(3):71–80. (page 6)

[131] Partl, C., Lex, A., Streit, M., Kalkofen, D., Kashofer, K., and Schmalstieg, D. (2012). enRoute: Dynamic path extraction from biological pathway maps for in-depth experimental data analysis. In *Proceedings of the IEEE Symposium on Biological Data Visualization (BioVis '12)*, pages 107–114. IEEE. (page 6)

[132] Partl, C., Lex, A., Streit, M., Kalkofen, D., Kashofer, K., and Schmalstieg, D. (2013). enRoute: Dynamic path extraction from biological pathway maps for exploring heterogeneous experimental datasets. *BMC Bioinformatics*, 14(Suppl 19):S3. (page 6)

[133] Partl, C., Lex, A., Streit, M., Strobelt, H., Wassermann, A. M., Pfister, H., and Schmalstieg, D. (2014). ConTour: Data-driven exploration of multi-relational datasets for drug discovery. *IEEE Transactions on Visualization and Computer Graphics (VAST '14)*, 20(12):1883–1892. (page 6)

[134] Petrone, P. M., Simms, B., Nigsch, F., Lounkine, E., Kutchukian, P., Cornett, A., Deng, Z., Davies, J. W., Jenkins, J. L., and Glick, M. (2012). Rethinking molecular

similarity: Comparing compounds on the basis of biological activity. *ACS Chemical Biology*, 7(8):1399–1409. (page 14)

[135] Pretorius, A. J., Purchase, H. C., and Stasko, J. T. (2014). Tasks for multivariate network analysis. In Kerren, A., Purchase, H. C., and Ward, M. O., editors, *Multivariate Network Visualization*, number 8380 in Lecture Notes in Computer Science, pages 77–95. Springe. (page 18)

[136] Proskocil, B. J. and Fryer, A. D. (2005). Beta2-agonist and anticholinergic drugs in the treatment of lung disease. *Proceedings of the American Thoracic Society*, 2(4):305–310. (page 120)

[137] Roberts, P. J. and Der, C. J. (2007). Targeting the Raf-MEK-ERK mitogen-activated protein kinase cascade for the treatment of cancer. *Oncogene*, 26(22):3291–3310. (page 98)

[138] Rohrschneider, M., Heine, C., Reichenbach, A., Kerren, A., and Scheuermann, G. (2010). A novel grid-based visualization approach for metabolic networks with advanced focus&context view. In Eppstein, D. and Gansner, E., editors, *Graph Drawing*, volume 5849 of *Lecture Notes in Computer Science*, pages 268–279. Springer Berlin Heidelberg. (page 33, 34)

[139] Sander, G. (1996). Layout of compound directed graphs. Technical report, FB Informatik, Universität des Saarlandes. (page 96)

[140] Saraiya, P., North, C., and Duca, K. (2005). Visualizing biological pathways: Requirements analysis, systems evaluation and research agenda. *Information Visualization*, 4(3):191–205. (page 27, 30)

[141] Sarkar, M. and Brown, M. H. (1994). Graphical fisheye views. *Communications of the ACM*, 37(12):73–83. (page 23)

[142] Schulz, H.-J. (2010). *Explorative Graph Visualization.* PhD thesis, University of Rostock. (page 19)

[143] Schulz, H.-J., Hadlak, S., and Schumann, H. (2011). The design space of implicit hierarchy visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):393–411. (page 21)

[144] Schulz, H.-J., John, M., Unger, A., and Schumann, H. (2008). Visual analysis of bipartite biological networks. In *Proceedings of the Eurographics Workshop on Visual Computing for Biomedicine (VCBM '08)*, pages 135–142. The Eurographics Association. (page 40)

[145] Shannon, R., Holland, T., and Quigley, A. (2008). Multivariate graph drawing using parallel coordinate visualisations. Technical report, University of St Andrews. (page 27, 28)

[146] Shen, Z. and Ma, K.-L. (2007). Path visualization for adjacency matrices. In *Eurographics/ IEEE-VGTC Symposium on Visualization*. The Eurographics Association. (page 39)

[147] Shendure, J. and Ji, H. (2008). Next-generation DNA sequencing. *Nature Biotechnology*, 26(10):1135–1145. (page 11)

[148] 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)*, pages 336–343. IEEE. (page 25)

[149] Shneiderman, B. and Aris, A. (2006). Network visualization by semantic substrates. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '06)*, 12(5):733–740. (page 41)

[150] Smoot, M. E., Ono, K., Ruscheinski, J., Wang, P.-L., and Ideker, T. (2011). Cytoscape 2.8: New features for data integration and network visualization. *Bioinformatics*, 27(3):431–432. (page 30, 36, 41, 80)

[151] Stasko, J., Görg, C., and Liu, Z. (2007). Jigsaw: Supporting investigative analysis through interactive visualization. In *Proceedings of the IEEE Symposium on Visual Analytics in Science and Technology (VAST '07)*, pages 131–138. IEEE. (page 22, 40, 110)

[152] Steinbeck, C., Han, Y., Kuhn, S., Horlacher, O., Luttmann, E., and Willighagen, E. (2003). The chemistry development kit (CDK): An open-source java library for chemo- and bioinformatics. *Journal of Chemical Information and Computer Sciences*, 43(2):493–500. (page 117)

[153] Steinberger, M., Waldner, M., Lex, A., Streit, M., and Schmalstieg, D. (2011). Context-preserving visual links. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '11)*, 17(12):2249–2258. (page 73)

[154] Stitz, H., Luger, S., Streit, M., and Gehlenborg, N. (2016). AVOCADO: Visualization of workflow–derived data provenance for reproducible biomedical research. *Computer Graphics Forum (EuroVis '16)*, 35(3):481–490. (page 25)

[155] Strebhardt, K. and Ullrich, A. (2008). Paul Ehrlich's magic bullet concept: 100 years of progress. *Nature reviews. Cancer*, 8(6):473–480. (page 13)

[156] Streit, M., Kalkusch, M., Kashofer, K., and Schmalstieg, D. (2008). Navigation and exploration of interconnected pathways. *Computer Graphics Forum (EuroVis '08)*, 27(3):951–958. (page 27, 33)

[157] Streit, M., Lex, A., Gratzl, S., Partl, C., Schmalstieg, D., Pfister, H., Park, P. J., and Gehlenborg, N. (2014). Guided visual exploration of genomic stratifications in cancer. *Nature Methods*, 11(9):884–885. (page 42)

[158] Streit, M., Lex, A., Kalkusch, M., Zatloukal, K., and Schmalstieg, D. (2009). Caleydo: Connecting pathways and gene expression. *Bioinformatics*, 25(20):2760–2761. (page 28, 33, 42)

[159] Strobelt, H., Bertini, E., Braun, J., Deussen, O., Groth, U., Mayer, T. U., and Merhof, D. (2012). HiTSEE KNIME: A visualization tool for hit selection and analysis in high-throughput screening experiments for the KNIME platform. *BMC Bioinformatics*, 13(Suppl 8):S4. (page 106)

[160] Sugiyama, K., Tagawa, S., and Toda, M. (1981). Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man and Cybernetics*, 11(2):109–125. (page 19)

[161] Tamassia, R., editor (2013). *Handbook of graph drawing and visualization.* Number 81 in Discrete Mathematics and Its Applications. Chapman & Hall/CRC, London. (page 20)

[162] Tekušová, T. and Kohlhammer, J. (2008). Visual analysis and exploration of complex corporate shareholder networks. In *Proceedings of the SPIE Conference on Visualization and Data Analysis (VDA '08)*, volume 6809, pages 68090F–1–68090F–10. (page 37)

[163] The International SNP Map Working Group (2001). A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature*, 409(6822):928–933. (page 8)

[164] Tomczak, K., Czerwińska, P., and Wiznerowicz, M. (2015). The cancer genome atlas (TCGA): An immeasurable source of knowledge. *Contemporary Oncology*, 19(1A):A68–A77. (page 15)

[165] Tominski, C., Abello, J., and Schumann, H. (2009a). CGV—an interactive graph visualization system. *Computers & Graphics*, 33(6):660–678. (page 22, 32)

[166] Tominski, C., Abello, J., and Schumann, H. (2009b). Two novel techniques for interactive navigation of graph layouts. In *Poster Compendium of the Eurographics/IEEE Symposium on Visualization (EuroVis '09).* (page 23)

[167] Tufte, E. (1983). *The Visual Display of Quantitative Information.* Graphics Press, Cheshire, CT, USA, 2nd edition. (page 28)

[168] Valiati, E. R. A., Pimenta, M. S., and Freitas, C. M. D. S. (2006). A taxonomy of tasks for guiding the evaluation of multidimensional visualizations. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV '06)*, pages 1–6, New York, NY, USA. ACM. (page 18)

[169] van den Elzen, S., Holten, D., Blaas, J., and van Wijk, J. (2016). Reducing snapshots to points: A visual analytics approach to dynamic network exploration. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):1–10. (page 21, 127)

[170] van den Elzen, S. and van Wijk, J. (2014). Multivariate network exploration and presentation: From detail to overview via selections and aggregations. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '14)*, 20(12):2310–2319. (page 28)

[171] van Ham, F. and Perer, A. (2009). "Search, show context, expand on demand": Supporting large graph exploration with degree-of-interest. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '09)*, 15(6):953–960. (page 25)

[172] van Ham, F. and van Wijk, J. J. (2002). Beamtrees: Compact visualization of large hierarchies. In *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002*, pages 93– 100. IEEE. (page 21)

[173] Vehlow, C., Beck, F., and Weiskopf, D. (2015). The state of the art in visualizing group structures in graphs. In *Eurographics Conference on Visualization (EuroVis) - STARs*, pages 21–40, Cagliari, Italy. The Eurographics Association. (page 93)

[174] Vital-Lopez, F. G., Memišević, V., and Dutta, B. (2012). Tutorial on biological networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(4):298–325. (page 8)

[175] Wang, J., Li, Z.-x., Qiu, C.-x., Wang, D., and Cui, Q.-h. (2012). The relationship between rational drug design and drug side effects. *Briefings in Bioinformatics*, 13(3):377–382. (page 12)

[176] Wang, Y., Wang, Y., Sun, Y., Zhu, L., Lu, K., Fu, C. W., Sedlmair, M., Deussen, O., and Chen, B. (2018). Revisiting stress majorization as a unified framework for interactive constrained graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):489–499. (page 20)

[177] Wassermann, A. M., Lounkine, E., and Glick, M. (2013). Bioturbo similarity searching: combining chemical and biological similarity to discover structurally diverse bioactive molecules. *Journal of chemical information and modeling*, 53(3):692–703. (page 14)

[178] Wattenberg, M. (2002). Arc diagrams: Visualizing structure in strings. In *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002*, pages 110– 116. IEEE. (page 19)

[179] Westenberg, M. A., Van Hijum, S. A. F. T., Kuipers, O. P., and Roerdink, J. B. T. M. (2008). Visualizing genome expression and regulatory network dynamics in genomic and metabolic context. *Computer Graphics Forum (EuroVis '08)*, 27(3):887–894. (page 27)

[180] Yamada, T., Letunic, I., Okuda, S., Kanehisa, M., and Bork, P. (2011). iPath2.0: Interactive pathway explorer. *Nucleic Acids Research*, 39(Suppl 2):W412–W415. (page 31)

[181] Yee, K.-P., Swearingen, K., Li, K., and Hearst, M. (2003). Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*, pages 401–408. ACM. (page 24)

[182] Yen, J. Y. (1971). Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716. (page 35)

[183] Young, D. and Shneiderman, B. (1993). A graphical filter/flow representation of boolean queries: A prototype implementation and evaluation. *Journal of the American Society for Information Science*, 44(6):327–339. (page 25)

[184] Young, F. W. and Hamer, R. M., editors (1987). *Multidimensional scaling: History, theory, and applications.* Lawrence Erlbaum Associates, Hillsdale, NJ, US. (page 21)

[185] Zhang, Y., Cheng, G., and Qu, Y. (2013). Towards exploratory relationship search: A clustering-based approach. In Kim, W., Ding, Y., and Kim, H.-G., editors, *Semantic Technology*, number 8388 in Lecture Notes in Computer Science, pages 277–293. Springer. (page 38)

[186] Zhao, J., Collins, C., Chevalier, F., and Balakrishnan, R. (2013). Interactive exploration of implicit and explicit relations in faceted datasets. *IEEE Transactions on Visualization and Computer Graphics (VAST '13)*, 19(12):2080–2089. (page 24)