Graz University of Technology

Institute for Computer Graphics and Vision

Dissertation

Guided Visual Analysis of Heterogeneous Data

Dipl.-Ing. **Marc Streit** Bakk.rer.soc.oec.
Graz, Austria, February 2011

Thesis supervisors

Univ. Prof. DI Dr. techn. **Dieter Schmalstieg**
Prof. Dr.-Ing. habil. **Heidrun Schumann**

To My Family

Visual Analytics: a Grand Challenge in Science - Turning Information Overload into the Opportunity of the Decade

Jim Thomas (1946 - 2010)

Abstract

Gaining insights by exploring massive, complex data is the grand challenge of Visual Analytics – the science of analytical reasoning. As heterogeneous data from different sources is being increasingly linked, it becomes difficult for users to understand how the data is connected, to identify what means are suitable to analyze a given data set, or to find out how to proceed for achieving a given analysis task. The analyst support a user needs is twofold: first, the analyst needs to be oriented within the information landscape – orientation support; and secondly, with orientation as a prerequisite, the user can be guided towards a specific analysis goal by following concrete recommended steps – guidance support.

In order to realize analyst support on both levels, a unified representation of the knowledge about the infrastructure and the workflow to be performed is required. This dissertation proposes a new model-driven design process that effectively co-designs aspects of data, view, analytics and tasks. A workflow, composed of individual tasks, is used as a trajectory through data, interactive views and computational tools.

Drawing upon information from a well-defined model and using the Caleydo visualization framework as infrastructure, this thesis introduces novel visualization techniques that are targeted at assisting users in terms of orientation. With this focus, three novel techniques are proposed – the Matchmaker, the Jukebox and the Bucket – all of which particularly address orientation support in the classic setups of visual analysis systems where one or multiple data sets are loaded in a multi-view system. The techniques presented rely on visual links as an additional visual cue – making the relations within and between the data sets more explicit. In addition to these traditional setups, further setup characteristics are discussed: first, a system that is targeted at orientation support in an analysis that spans multiple existing applications; and secondly, the analysis of heterogeneous data in a collaborative scenario. This thesis concludes with the Stack'n'flip system that utilizes the information captured in the model for realizing comprehensive analyst support on both support levels: orientation and guidance.

The theoretical model as well as the visualization techniques in this work are motivated and demonstrated by means of real world data from the biomedical domain.

Keywords: Visual analytics, guidance, orientation, authoring, visual links, information linking, multiple data sets, Caleydo visualization framework.

Kurzfassung

Die explorative Analyse großer, vernetzter Datenmengen ist eine zentrale Herausforderung, mit der sich das junge Wissenschaftsgebiet Visual Analytics beschäftigt. Aufgrund der Masse, Heterogenität und Komplexität der Daten gestaltet es sich für den Analysten schwierig zu verstehen, wie die Daten miteinander verbunden sind, welche Möglichkeiten zur Analyse für einen Datensatz geeignet sind, oder wie für eine bestimmte Aufgabe vorgegangen werden soll. Hierbei benötigt der Analyst Unterstützung in zweifacher Ausprägung: In erster Linie gilt es diesen in der Informationslandschaft zu orientieren. Basierend darauf kann ein System den Benutzer durch das Vorschlagen konkreter, zukünftiger Schritte durch die Analyse führen (anleitende Unterstützung).

Um die Benutzerunterstützung auf beiden Ebenen umsetzen zu können, ist eine einheitliche Repräsentation des Wissens über die Infrastruktur und den Analyseprozess notwendig. Diese Dissertation schlägt einen modellbasierten Ansatz vor, welcher effektiv die zugrundeliegenden Daten, die visuellen Repräsentationen, die analytischen Werkzeuge sowie die Analyseaufgaben erfasst und miteinander verbindet.

Basierend auf einem definierten Modell und unter Verwendung des Visualisierungssystems Caleydo führt diese Arbeit zunächst neue Visualisierungstechniken ein, welche auf orientierende Unterstützung abzielen. Mit diesem Fokus werden drei Techniken vorgeschlagen: Matchmaker, Jukebox und Bucket, welche speziell die Orientierung in klassischen Analyseszenarien mit einem oder mehreren Datensätzen in einem Multiple-View-System adressieren. Die Techniken verwenden Visual Links als zusätzliches Wahrnehmungselement, um die Beziehungen in den Daten explizit zu veranschaulichen. Neben den traditionellen Setups werden weitere Aspekte beleuchtet: einerseits ein System, das Orientierungsunterstützung in einem sich über mehrere existierende Applikationen umspannende Analyse-Setup zur Verfügung stellt; andererseits die Analyse von heterogenen Daten in einem kollaborativen Ansatz. Diese Dissertation schließt mit der Vorstellung des Stack'n'flip-Systems, welches die im Modell erfassten Informationen für orientierende, aber auch anleitende Unterstützung nutzt.

Sowohl das theoretische Modell als auch die Visualisierungstechniken werden anhand von realen Daten aus der Biomedizin motiviert und demonstriert.

Schlüsselwörter: Visuelle Datenanalyse, Analyseprozess, Orientierung, Autorensystem, heterogene Daten, visuelle Verknüpfung von Information, Caleydo Visualisierungssystem.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Place Date Signature

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.

Ort Datum Unterschrift

Acknowledgments

I would like to express my gratitude to all the people who have supported me over the last several years. First of all, I want to thank my supervisor Prof. Dieter Schmalstieg who believed in me and gave me the freedom to follow my ideas, but at the same time guided me in the right directions. While working in his group at TU Graz I learned a lot about structured thinking, leadership and professional scientific work.

My thanks are also extended to my second supervisor Prof. Heidrun Schumann for her advice and valuable input that influenced my research in a very positive way. In particular, I was inspired by her special motivating skills that pushed me forward.

I am grateful to my dear friend and colleague Alexander Lex with whom I have collaborated closely over the last couple of years. I want to thank him for the myriad professional and personal discussions, the squash fights for relieving the stress and the many other activities we did together. It is very special to be able to work in a team with a close friend who is always there for you.

My sincerest thanks also go to my colleagues who made the Institute for Computer Graphics and Vision an extraordinary place to work. In particular, I would like to thank Manuela Waldner, Bernhard Kainz, Denis Kalkofen, Daniel Wagner, Markus Grabner, Markus Steinberger, Gerhard Schall, Erick Mendez, Hans-Jörg Schulz, Mark Dokter, Andreas Wurm, Albert Walzer, Renate Hönel, Christina Fuchs, Manuela Reinisch and Prof. Horst Bischof – all of whom, in one or the other way, influenced this thesis and are part of an environment that is professional and fun to work in.

Lastly, and most importantly, I want to express my deepest thanks to my family. Thank you mom and dad for supporting me in every respect of my life. Thank you Katja for being such a great sister. Thank you Grandma for being the kindhearted person you are. Finally, I want to thank my beloved girlfriend Barbara for tolerating my long working hours, giving me stability and making me happy. I'm glad to have you all in my life!

This work was funded in part by the Austrian Research Promotion Agency (FFG) through the *InGenious* project (385567) and the *Genoptikum* project (813398), the *VIPEM* project (L427-N15) granted by the Austrian Science Fund (FWF), and the German Research Foundation (DFG) research training school 1387 *dIEM oSiRiS*.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

Over the last few decades many fields of science have been confronted with tremendous amounts of data and continuously increasing annual growth rates. However, it is an undisputed fact that generating the data is not the grand challenge anymore, but analyzing it [Nielsen, 2009, Thomas and Cook, 2005]. Raw data by itself is useless if we lack the tools to analyze it [Keim et al., 2010]. The central question is how to turn data into meaningful information, which can be transformed into knowledge, in the sense of [Ackoff, 1989, Chen et al., 2009, Wang et al., 2009]. This is particularly relevant for the fields of biology and medicine.

The sheer amount of data, however, is only one aspect of the problem. In order to solve complex analysis questions, a deep understanding of the problem itself is of vital importance. Assuming this as a precondition, in many cases the information that leads to the correct answers is present, but often hidden in the unstructured mass of data. In this connection it is necessary to cope with heterogeneous data sets from different sources, on distinct levels of scale and stored in various formats and types (*e.g.*, text, maps, graphs, images, *etc.*).

Visualization has proven its value for the interactive analysis of homogeneous data. However, in order to analyze these enormous amounts of heterogeneous interrelated sets of data, visualization alone is often not sufficient. In contrast to visualization, automated methods scale better to big data sets but sometimes fail or get stuck in local optima.

Furthermore, traditional algorithms run in a black box fashion, where the user has no means to actively intervene once the process has been triggered. *Visual Data Mining*, which can be considered as the predecessor to Visual Analytics, addresses the shortcoming of each of the single approaches by bringing them together and exploiting the strength of both (*cf.*, [Kreuseler and Schumann, 2002]). However, the field of *Visual Analytics* takes the next step and goes beyond this combination. It strives to combine a series of scientific disciplines: visualization, data management, data mining, spatio-temporal data analysis, evaluation, human perception and cognition (*cf.*, [Keim et al., 2010]). All these separate fields are tied together by humans, which have the ability of combining and reasoning, but also have intuition and background knowledge [Keim et al., 2009].

In 2005 a road map for the young field of Visual Analytics was formulated [Thomas and Kielman, 2009]. Since then, a lot of research has been conducted in this direction, making Visual Analytics one of the most vital and quickly developing fields of science. Although profound progress has been made on multiple topics and sub-issues related to Visual Analytics, a lot of open challenges still remain. More recently, the book “Mastering the Information Age – Solving Problems with Visual Analytics” [Keim et al., 2010] reviews the advances in the field and summarizes the current challenges. The following section outlines which of these challenges are addressed by this dissertation in particular.

1.1 Problem Statement and Goals

The aforementioned challenging data characteristics of Visual Analytics problems increase the danger of the user of getting lost within the analysis [Keim et al., 2010, p.1]. A disoriented analyst without a correct mental map cannot conduct an effective, targeted analysis. In the sense-making process, users need to compare, evaluate and interpret pieces of information. However, these information intensive tasks are error-prone, tedious and time-consuming. Consequently, it is up to the visual analysis system to facilitate the understanding of the relations within as well as across data sets and in turn also the association to and connections between the visual representations involved – thus, providing **orientation support**.

Assuming that a user is aware of these relations, his being confronted with an overwhelmingly rich set of choices is an additional challenge. The analyst needs to decide which visual or computational interface is appropriate for performing a certain task of the analysis. It is this degree of freedom that makes it problematic to effectively run the series of steps it takes to reach a specific analysis goal. However, this is not foremost an issue of traditional user interface design and improvement, as it is unrealistic to think that this is possible by just providing an intuitive user interface. In fact, it is essential to assist the user in making the right decisions – providing **guidance** – without restraining him in any way.

In order to be able to provide this kind of analysis support in an automated fashion, an externalization of the knowledge about the infrastructure and the workflow to perform is required. In particular, the interplay between data, view, and task needs to be captured in a unified representation. Schulz elaborated on the dependencies between these three levels for explorative graph visualization [Schulz, 2010]. However, this management of semantics, as discussed in [Keim et al., 2010, p.33 ff.], needs to be formalized in a general way, so that it can potentially be applied to a wider spectrum of analysis scenarios. Only this externalized knowledge can serve as a basis for a system that guides analysts through a heterogeneous, complex set of data. Guidance for exploratory data analysis is a hot topic in visual analytics research, see for instance [Perer and Shneiderman, 2008] as well as the work on DimStiller [Ingram et al., 2010] that evolved in parallel to this thesis, which address guidance tailored to dimensional analysis and reduction. However, the ultimate goal lies in the provision of a "walk-up usable" interface [Keim et al., 2009] that actively assists analysts at all stages of their work and in turn speeds up the analysis process. In a first step, this is done by providing the means for keeping a user oriented during the analysis; and in a second step, with an intact mental map to build on, it is done by visually guiding a user through the analysis towards a specific goal.

1.2 Contributions

This thesis is centered around the question of how a visual analysis system can actively assist a user in the analysis of a interwoven heterogeneous set of data. The analyst support that a user needs is twofold: first, the analyst needs to be oriented in order to keep up the mental map of the analysis setup – **orientation support**; and secondly, with orientation as a prerequisite, the user can be guided towards a specific analysis goal – **guidance support**.

Providing support on both levels can only be achieved by externalizing the information of the setup, so that a visual analysis system can dynamically employ this information. The first major contribution of this thesis is a **Three-Stage Model-Driven Design Process** that realizes this externalization. It makes it possible to specifically define in detail

- **what** can be analyzed (*data set*)
- by **whom** (*expert user*)
- in **which way** (*visual and computational interfaces*)
- with **which goal** (*aim of this analysis task*), and
- in **which order** (*workflow*).

Thus, the model captures the entire analysis session, going well beyond the pure definition of individual analysis tasks. In a first authoring stage, a basic model of the given setup is created which considers data sets from different sources, relations between them and visual as well as computational interfaces operating on them. A visual analysis system employing these models can support the analyst by providing orientation within the conglomerate of data sets, where not only previous but also possible future analysis steps are shown. On top of the setup model, a set of domain-specific tasks are defined, forming the domain model. In the last stage, the analysis session model, which contains a workflow targeted at a concrete analysis goal, is defined. In combination, these three models can be used to actively guide the analyst along a predefined path. However, as the models cover both support levels, the user has always the opportunity to leave the suggested path at any time and switch to orientation support.

The thesis proceeds by introducing the **Caleydo Visual Analysis Framework**. The framework serves as the general infrastructure for realizing the proposed interactive visualization techniques throughout this thesis and therefore constitutes the second major contribution of this work.

Drawing upon information from a well-defined model and using the Caleydo visualization framework as infrastructure, this thesis then goes on to introduce a series of novel visualization techniques that are specifically targeted at assisting users in terms of orientation. With this focus, three novel techniques are proposed, the **Matchmaker**, the **Jukebox** and the **Bucket**, which particularly address orientation support in the classic setups of visual analysis systems, *i.e.*, one or multiple data sets loaded in a multi-view system. After elaborating on these traditional setups, further setup characteristics are introduced and discussed: first, a system that is targeted at the provision of orientation support in an analysis that comprises multiple existing applications; and secondly, the analysis of complex, heterogeneous data in a collaborative visual analysis setup. However, both topics, the analysis across multiple applications as well as across multiple users, are not the focus of this thesis and therefore only covered briefly with special attention to the potential influences of the model on this kind of heterogeneous setups.

In contrast to orientation support, hardly any work has been published aiming at systematic guidance towards a specific analysis goal based on defined unified representation – a model. As a first step to fill this gap, the thesis concludes with the presentation of the Stack'n'flip system that realizes both levels of support by utilizing the information of all three stages of the model. These visualization techniques which realize the orientation as well as guidance support form the third major contribution of this dissertation.

1.3 The Role of Life Science Research in Visual Analytics

Life science researchers are confronted with highly complex, interconnected pieces of data. In order to understand how organisms work and what factors cause diseases, these pieces of information need to be considered in combination. These challenges in terms of sense-making and knowledge discovery make the field of biology a prime area of application for Visual Analytics. This trend in the InfoVis community towards biology is also reflected by a series of specially designated yearly events that were established within the visualization community over the last couple of years: the *Eurographics Workshop on Visual Computing* *in Biology and Medicine (VCBM)* and the *Workshop on Visualizing Biological Data (VizBi)* are only two examples. Also the panel discussion at the VisWeek '10 on the topic *Challenges in Visualizing Biological Data* as well as the *Visual Analytics in Health Care* workshop underline the importance and topicality in our community. With the newly established *IEEE Symposium on Biological Data Visualization (BioVis)* that will be held the first time at VisWeek '11 this strong trend will be continued.

Moreover, the life science community itself identified Visual Analytics as a central enabling technology that has become indispensable. Only recently, Nature Methods devoted a special issue to "Visualizing Biological Data" [Evanko, 2010] which, besides reviewing the current state-of-the-art, aims to increase the awareness of the necessity of Visual Analytics for the life sciences.

Although the concepts and techniques presented in this thesis are targeted at solving general Visual Analytics problems, they are demonstrated by means of biological data, addressing open research questions from the life sciences. Especially the understanding of genetic functions was identified as a particularly complex Visual Analytics problem [Keim et al., 2010].

1.4 Collaboration Statement

The list of paper co-authors from the next section provides a good overview of the people without whom this thesis would not have been possible. However, the following list of colleagues and institutions deserve to be specially mentioned, as a significant part of the presented work was done in close collaboration with them:

- **Alexander Lex** from Graz University of Technology has definitely been the most important collaborator over the last couple of years. We have both committed our scientific work to the Caleydo project in which we jointly designed and developed the Caleydo framework. His work influences all parts of the results, except the early work before 2008.
- **Hans-Jörg Schulz** from University of Rostock contributed to the conception of the model-driven design for the analysis of heterogeneous data, especially to the theoretical foundation.
- **Manuela Waldner** from Graz University of Technology was the driving force of the work on visual links across applications as well as the developments in the direction of co-located collaborative data analysis.
- **Michael Kalkusch**, former researcher from Graz University of Technology, initiated the Caleydo project in 2005 and laid the groundwork for the collaboration with the Medical University of Graz.
- **Werner Puff** did his master's thesis on collaborative information visualization in a multi-desktop environment within the Caleydo project. He worked on a distributed

version of the Caleydo framework which was in turn the basis for the visual links across applications.

- **Bernhard Schlegl** was involved in the Caleydo project as a master's student. His main contribution was the implementation of the hierarchical heat map approach as well as the integration of clustering algorithms into the framework.
- **Christian Partl** also participated as a student. He supported the implementation of the Matchmaker approach and also worked on the implementation of Caleydo's flexible ID-mapping mechanism.
- **Ernst Kruijff** from Graz University of Technology was involved in the design and realization of the user study evaluating the Bucket visualization technique and the integrated visual linking approach.
- **Institute of Pathology, Medical University of Graz** was a main collaboration partner over the last few years. As life science experts with real needs in terms of visualization and data analysis, Prof. Dr. Kurt Zatloukal, Dr. Karl Kashofer and Dr. Martin Asslaber contributed valuable domain knowledge and requirements, which were the starting point for numerous solutions developed in the course of the thesis.
- **Ludwig Boltzmann Institute for Experimental and Clinical Traumatology** was a collaboration partner that contributed domain-specific requirements and analysis needs in the context of sepsis research. In particular, Prof. Dr. Heinz Redl, Dr. Gudrun Schmidt-Gann, Monika Schuller and Dr. Katharina Schmid were involved in the interdisciplinary research activities.

1.5 Related Publications

The following list of peer-reviewed, co-authored papers gives an overview of the publication activities done in the course of this thesis. The author's contributions are stated explicitly.

Primary Publications

The publications listed here are strongly related to the dissertation's topic. The contribution to these manuscripts can be found on all stages: the idea generation phase, its realization and evaluation (if contained) as well as substantial paper writing. The publications, sorted by date, are:

- **Marc Streit**, Hans-Jörg Schulz, Alexander Lex, Dieter Schmalstieg, Heidrun Schumann. *Model-Driven Design for the Visual Analysis of Heterogeneous Data.* Accepted with major revision to: IEEE Transactions on Visualization and Computer Graphics, 2011 [Streit et al., 2011].

This paper covers the three-stage model-driven approach as well as the Stack'n'flip system. Chapter 4 and 8 of this thesis are based on material from this manuscript.

- Alexander Lex, **Marc Streit**, Ernst Kruijff, Dieter Schmalstieg. *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), pp. 57-64. IEEE Computer Society Press, 2010. ISBN 424466856 [Lex et al., 2010a].

  This paper serves as core material for the Bucket visualization technique, presented in Section 6.3.3.

- **Marc Streit**, Alexander Lex, Michael Kalkusch, Kurt Zatloukal, Dieter Schmalstieg. *Caleydo: Connecting Pathways and Gene Expression.* Bioinformatics, Oxford Journals, vol. 25, no. 20, pp. 2760-2761, 2009 [Streit et al., 2009a].

  This application paper was targeted at the Bioinformatics community. It describes the concrete benefit of the Caleydo software for domain experts, in particular the Bucket visualization technique.

- **Marc Streit**, Hans-Jörg Schulz, Dieter Schmalstieg, Heidrun Schumann. *Towards Multi-User Multi-Level Interaction.* In Technical Report LMU-MI-2010-2, Ludwig Maximilias University Munich: Proceedings of the Workshop on Collaborative Visualization on Interactive Surfaces (part of VisWeek '09), pp. 5-8., 2009. ISSN 1862-5207 [Streit et al., 2009c].

  This position paper contains preliminary work on the model-driven design approach (see Chapter 4). It is the initial outcome that originated from the collaboration with the University of Rostock.

- **Marc Streit**, Michael Kalkusch, Karl Kashofer, Dieter Schmalstieg. *Navigation and Exploration of Interconnected Pathways.* Computer Graphics Forum (EuroVis '08), vol. 27, no. 3, pp. 951-958, 2008 [Streit et al., 2008].

  This manuscript serves as core material for the Jukebox visualization technique, presented in Section 6.2.2.

- **Marc Streit**, Michael Kalkusch, Dieter Schmalstieg. *Interactive Visualization of Metabolic Pathways.* In Poster Compendium of the IEEE Conference on Visualization (Vis '07), IEEE Computer Society Press, 2007 [Streit et al., 2007].

  The poster abstract contains preliminary work on the Jukebox technique.

Secondary Publications

The following list contains publications which are related but thematically not at the heart of this thesis. The author's own contribution is explicitly stated for each paper.

- Alexander Lex, **Marc Streit**, Christian Partl, Karl Kashofer, Dieter Schmalstieg. *Comparative Analysis of Multidimensional Quantitative Data.* IEEE Transactions on Visualization and Computer Graphics (InfoVis '10), vol. 16, no. 6, pp. 1027-1035, 2010 [Lex et al., 2010b].

  This paper presents the Matchmaker technique which is introduced in a compact manner in Section 6.1.2. The author of this thesis supported the idea finding process and contributed significantly to the implementation as well as the writing of the paper.

- Manuela Waldner, Werner Puff, Alexander Lex, **Marc Streit**, Dieter Schmalstieg. *Visual Links across Applications* - Best student paper award. In Proceedings of the Conference on Graphics Interface (GI '10). Canadian Human-Computer Communications Society, 2010 [Waldner et al., 2010].

  This publication serves as additional material for Section 7.1.1. In addition to the contributions made during the generation of the initial ideas, the author supported the integration of the Caleydo software into the general *Visual Links Across Application* system.

- **Marc Streit**, Alexander Lex, Helmut Doleisch, Dieter Schmalstieg. *Does software engineering pay off for research? Lessons learned from the Caleydo project.* In Poster Compendium of the Eurographics Workshop on Visual Computing for Biomedicine (VCBM '10). Eurographics, 2010 [Streit et al., 2010].

  This poster abstract serves as additional material for the Chapter on the Caleydo system. Major parts of the manuscript were written by the author of this thesis.

- Manuela Waldner, Alexander Lex, **Marc Streit,** Dieter Schmalstieg. *Design Considerations for Collaborative Information Workspaces in Multi-Display Environments.* In Technical Report LMU-MI-2010-2, Ludwig Maximilias University Munich: Proceedings of the Workshop on Collaborative Visualization on Interactive Surfaces (part of VisWeek '09), pp. 36-39, 2009. ISSN 1862-5207 [Waldner et al., 2009].

  The author's own contribution lies in the support during the realization of the prototype system, around which the ideas of the position paper revolve (see Section 7.1).

- Heimo Müller, Robert Reihs, Stefan Sauer, Kurt Zatloukal, **Marc Streit**, Alexander Lex, Bernhard Schlegl, Dieter Schmalstieg. *Connecting Genes with Diseases.* In Proceedings of the Conference on Information Visualisation (IV '09). IEEE Computer Society Press, 2009. ISBN 0769537337 [Mueller et al., 2009].

  The author contributed the sections on selected visualization techniques, part of the Caleydo framework.

- **Marc Streit**, Alexander Lex, Heimo Müller, Dieter Schmalstieg. *Gaze-Based Focus Adaption in an Information Visualization System.* In Proceedings of the Conference on Computer Graphics and Visualization and Image Processing (CGVCVIP '09), 2009 [Streit et al., 2009b].

The paper serves as additional material for Section 6.3.5. The author played a leading role in the realization of the gaze-based interaction in the Caleydo system and also contributed a major part of the written text.

- Gudrun Schmidt-Gann, Katharina Schmid, Monika Uehlein, Joachim Struck, Andreas Bergmann, Dieter Schmalstieg, **Marc Streit**, Alexander Lex, Douw G. van der Nest, Martijn van Griensven and Heinz Redl. *Gene- and Protein Expression Profiling in Liver in a Sepsis-Baboon Model.* In Proceedings of the Conference on Shock, 2009 [Schmidt-Gann et al., 2009].

  The work presented in this manuscript was carried out with the help of the Caleydo analysis software which served as enabling technology for generating the results.

- Heimo Müller, Kurt Zatloukal, **Marc Streit**, Dieter Schmalstieg. *Interactive Exploration of Medical Data Sets.* In Proceedings of the Conference on BioMedical Visualisation, pp. 29-35. IEEE Computer Society Press, 2008. ISBN 0769532844 [Mueller et al., 2008].

  The author's main contribution to this manuscript lies in the section on pathway analysis.

- Manuela Waldner, Christian Pirchheim, **Marc Streit**, Dieter Schmalstieg. *Multiple View Visualization On A Multi Display Setup.* In Poster Compendium of the Workshop on Giga-Pixel Displays & Visual Analytics (GIANT), 2008 [Waldner et al., 2008].

  This poster's content serves as additional material for Chapter 7. The author contributed in all aspects that concern the collaborative data analysis as well as the design of the case study.

1.6 Structure

The thesis is structured as follows:

Chapter 2 introduces the necessary life science background which serves as the basis in terms of input for the rest of the thesis. The demanding data characteristics as well as the complex domain problems provide the perfect playground for the development of new solutions.

Chapter 3 starts by defining the different levels of analyst support: as a first step, the user needs to be oriented in the information landscape, and secondly, the oriented user can be actively guided towards a concrete analysis goal. It continues by discussing the related work on guided visual analysis and which visual means can be utilized to convey the information to the analyst. The chapter finishes by categorizing existing work on supported visual analysis as well as the novel visualization techniques, which will be introduced throughout the subsequent chapters.

Chapter 4 introduces the three-stage model-driven design concept that aims to manage the semantics of visual data analysis. The model constitutes the theoretical foundation of this dissertation.

Chapter 5 presents the Caleydo visual analysis framework developed in the course of this thesis. The software framework serves as the main infrastructure that enabled the realization of the visualization techniques and case studies introduced in this thesis.

Chapter 6 introduces a series of visualization techniques that operate on a defined analysis setup model. The techniques aim to assist users by explicitly showing the relations within and across multiple data sets. The utilization of visual links, a particularly strong visual cue, plays a central role in achieving this goal.

Chapter 7 addresses two new challenges in terms of a setup's heterogeneity: on the one hand, the need for combining stand-alone applications in order to be able to cover all aspects of the analysis of heterogeneous data; and on the other hand, the demand for the inclusion of multiple users who jointly perform an analysis. Both aspects impose a whole new set of requirements in terms of orientation. In addition to the discussion of these factors for guided visual analysis, initial work in both directions is presented.

Chapter 8 introduces the Stack'n'flip system which employs the full three-stage model for providing analyst support on both support levels: orientation as well as guidance towards a particular analysis goal.

Chapter 9 concludes this thesis by summarizing the work, discussing further implications and indicating promising directions for future research.

Chapter 2

Life Science Background

Contents

This chapter gives an overview of the biological data that lay the groundwork in terms of input for this thesis – ranging from cellular networks, to molecular data, and further clinical data resources. While each homogeneous data set and each domain on its own is subject of intensive research, a simultaneous consideration of multiple aspects of the biological system poses new challenges in terms of data analysis – making more sophisticated support of the analyst necessary.

2.1 Multi-Level, Multi-Scale Biological and Biomedical Data

Organisms are complex, natural systems of which only a fraction is currently understood. In order to gain a deeper knowledge of these systems, research in biology and medicine is centered around discovering the missing links. However, the more we find out about how these systems function, the harder it gets to capture their complexity and to make sense of the incomplete and even partially erroneous body of knowledge. In a nutshell, the central question is: "How will big pictures emerge from a sea of biological data?" (*cf.*, [Pennisi, 2005]).

An essential concept used in coping with the complexity of these systems lies in investigating it on different semantic levels and on multiple levels of granularity. Figure 2.1 illustrates some of the levels in the context of biological systems. A very similar multi-level representation showing concrete data examples (*cf.*, Figure 2.2) was published in

2010 [O'Donoghue et al., 2010]. Data sets assigned to each of the levels cover distinct aspects of the natural system and therefore vary in format and type. While biological processes on the cellular level for instance are represented as graphs, imaging data shows structures on the organ level.

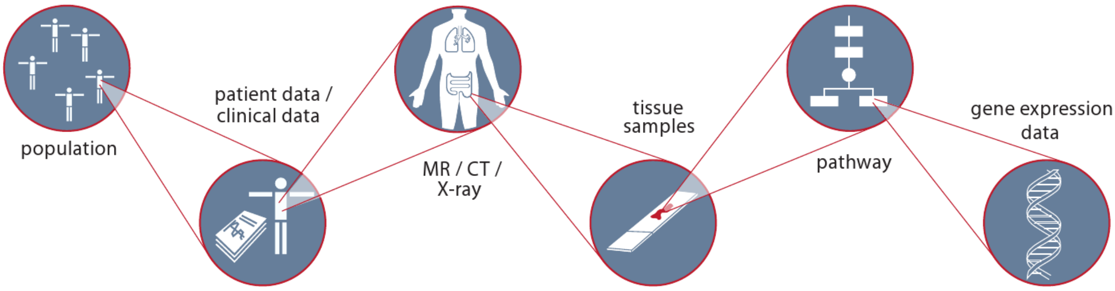


Figure 2.1: Patient-centered data hierarchy starting with a population on the top and going down to the gene regulation data of each individual patient [Streit et al., 2009c].

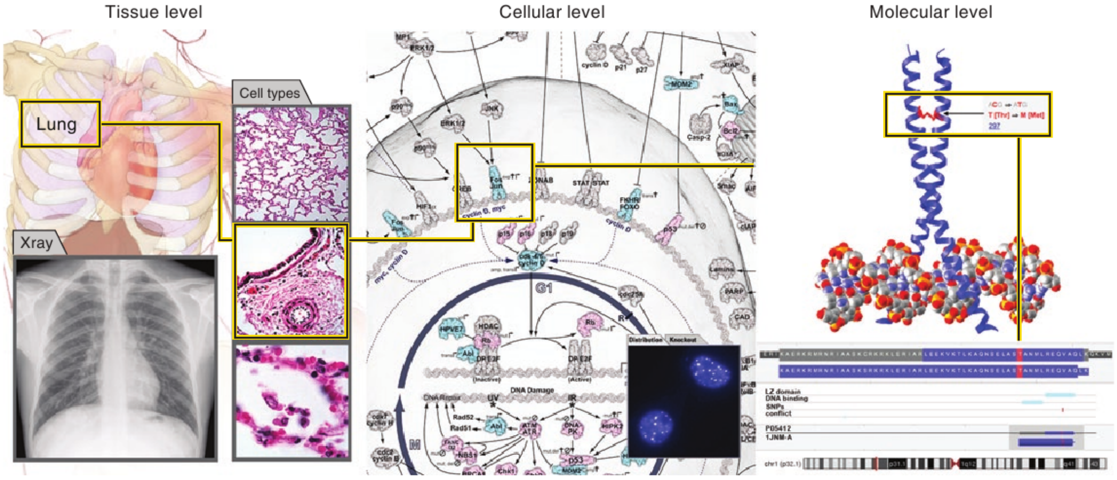


Figure 2.2: Concrete sample hierarchy for biological data [O'Donoghue et al., 2010].

However, this multi-level hierarchy is not meant as a universally valid data collection, as it contains only a subset of possible data sources and scales. It is rather defined by a research question. However, addressing different research questions may require the investigation of another set of data, leading to a differently composed hierarchy.

A main distinguishing factor of biological resources is whether or not a data set is associated with a concrete individual, animal model or experiment. Experiment-independent resources contain knowledge about generic functions. However, in order to understand the functioning behind organisms and diseases, both kinds of resources are essential. The subsequent sections introduce data of both kinds. Note that the selection of data resources presented here gives an impression of the wide spectrum of data types involved, ranging from high-resolution images over structured multi-dimensional data to networks and written text.

2.2 Patient-Independent Resources

The available body of generic knowledge in biology is covered by large database, mostly financed by public funds. As a multitude of different institutions offer these services, the data sources are scattered over the web and therefore hard to merge and uniformly access. Probably the most comprehensive collection of generic resources on all kinds of levels is the *Entrez*[1] database network hosted by the *National Center for Biotechnology Information (NCBI)*, illustrated in Figure 2.3.

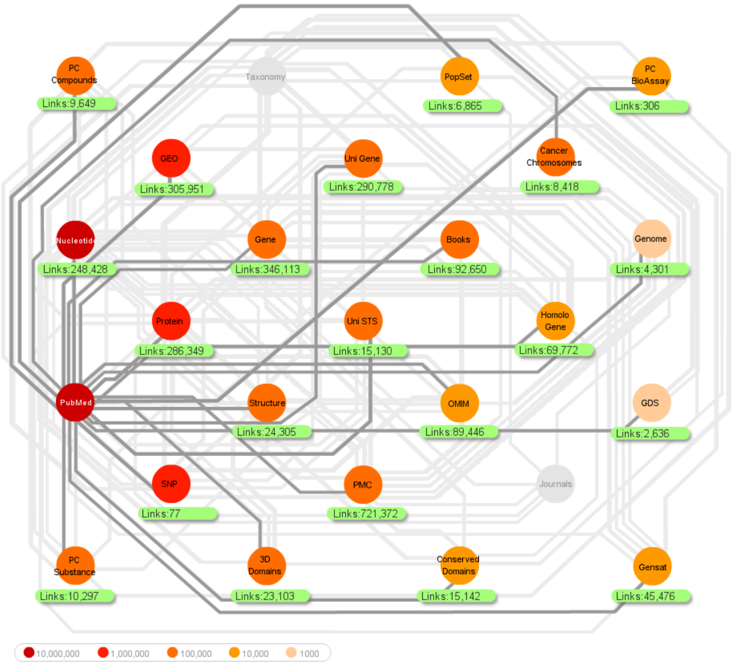


Figure 2.3: Interactive representation of the *Entrez* database model. By selecting a particular database, the links to other resources in the database network are highlighted. The color (red to light orange) encodes the number of records contained in each of the databases. The green labels specify how many links interconnect the separate databases. Source: `http://www.ncbi.nlm.nih.gov/Database`

When working with biomedical data, one has to keep in mind that the underlying data is imperfect. The knowledge is subject to a constant update process: new relations are discovered, existing connections are changed or invalidated, and many entities and relations are currently still unknown. The user must be aware that the current body of knowledge is only an incomplete snapshot of the complete and comprehensive systems. This volatile data basis has to be considered when designing new methods.

[1] `http://www.ncbi.nlm.nih.gov/Entrez`

The following paragraphs briefly introduce generic, patient-independent resources that are relevant in the context of this thesis.

Gene/protein database Information about genes and proteins is stored in public databases such as *GeneCards*[2] and *Entrez Gene*[3]. These web sources include meta-information such as short names, alternative identifiers, a detailed description, references to publications and disease classifications.

Disease database Diseases and health-related conditions are classified according to various disease schemes. The most commonly used classification is the International Statistical Classification of Diseases and Related Health Problems (ICD)[4] published by the WHO.

Publications Published articles about genes, proteins, pathways and diseases play an important role during the analysis, as an analyst can gain deeper knowledge of the topics if needed. The most commonly used database for literature research in the biomedical domain is *PubMed*[5].

Cellular Networks: Pathways Pathways are graphs representing biological processes in cells. They describe states of and transformations between molecular entities. These entities are, for instance, genes, proteins, enzymes and/or metabolites – depending on the scope of the pathway (*e.g.*, signaling pathways, biochemical pathways, protein-protein interaction networks, *etc.*). Nodes represent the biological entities, and edges the reactions within a cell or chemical signals between or inside the cells. A complete definition of pathways in terms of graph theory is given in [Klukas and Schreiber, 2006]. Pathways are generic models valid for a whole species and therefore independent of specific individuals.

2.3 Patient-Specific Resources

In contrast to these public, generic databases, a wide range of data exists which can be directly associated with a concrete patient or experiment. The following paragraphs give a short overview of the most relevant patient-related resources.

Anamnesis The anamnesis is a patient's medical history – including illnesses, allergies, hereditary diseases, *etc.*

Lab results Lab results include blood levels, urine and stool sample results, *etc.*

[2] `http://www.genecards.org`

[3] `http://www.ncbi.nlm.nih.gov/sites/entrez?db=gene`

[4] `http://www.who.int/classifications/icd/en`

[5] `http://www.ncbi.nlm.nih.gov/pubmed`

MR / CT / X-ray Magnetic resonance (MR), computer tomography (CT) and X-ray data is acquired using imaging techniques. For cancer patients, a tumor might be visible in one, several or all of the imaging data sets. In some cases, computer-based analysis, such as automatic tumor segmentation, is employed.

Tissue samples When a tumor is discovered, the standard procedure is to take a biopsy. The acquired tissue is sliced, applied to a glass slide and stained. When magnified, cell conglomerates as well as individual tumor cells become visible. In addition to their investigation under the microscope, high-resolution scans are acquired and stored in a database.

Gene-/protein expression Since the human genome as well as the genome of many other species is completely sequenced, a major thrust of scientific effort has shifted towards the identification of gene functions [Pellegrini et al., 2001]. High-throughput techniques like DNA micro-arrays [Schena et al., 1995] enable biomedical experts to measure the regulation of ∼omics data (genomics, proteomics, metabolomics, *etc.*– for details see [Gehlenborg et al., 2010]) for a patient (or any other sequenced organism) at a specific point in time. This snapshot of the expression tells a life scientist how active a gene or protein is, which influences the cellular processes and in turn the diseases themselves. Again, whether the expression data describes the regulation of genes or proteins is a question of the biological level and depends on the analysis and the questions asked. However, from the perspective of data processing and visualization, the difference is not essential, as the data structure remains the same: multi-dimensional, numerical data.

2.4 Biobanks

Collections of human material of patients, such as tissue, blood and further patient-related data on various levels, are called *Biobanks* [Hagen and Carlstedt-Duke, 2004, Asslaber and Zatloukal, 2007]. These comprehensive databases have the potential to serve as a basis for large scale comparison analyses between diseased and healthy individuals. The goals behind the establishment of Biobanks are ambitious: finding cure for widespread diseases like Alzheimer's disease, cancer, diabetes and heart disease [Collins et al., 2003].

Traditionally, each hospital used to have and often still has a local, rather small data collection. One reason for this are the strict privacy regulations stating that the biological material and the associated information needs to be held confidential and can only be accessed by authorized clinical and research institutions. However, over the last several years, intensive efforts have been taken to consolidate these local collections by advancing the integration to joint, international databases. While the collection and consolidation of the data in these Biobanks remains a huge and only partially solved engineering and legal challenge, it is still unclear which tools and methods can cope with the data in order to gain new insights and find some of the missing links.

The Institute of Pathology at the Medical University of Graz serves as the hosting institution of the Austrian Biobanking infrastructure. In addition, as a coordinator of the EU-wide *Biobanking and Biomolecular Resources Research Infrastructure (BBMRI)*[6] project, they initiated and advanced the data fusion process among the European countries. As the main collaboration partner during the thesis project, they approached us with their needs in terms of data analysis in the context of Biobanking and also provided us with anonymized real-world data for developing and testing our solutions. Due to their specialized focus on gene expression data, pathways and the combination of the two, a part of the visualization techniques proposed throughout this thesis address the analysis of these data resources.

2.5 Summary

This chapter was dedicated to the collection of biological and biomedical data that comprises the basis for this dissertation with regards to the input. Data resources of various kinds were basically divided into generic knowledge about organisms and data that is associated with a certain individual, experiment or animal model.

Life science research is dedicated to gaining a better understanding of complex natural systems. The long term objective is to find a cure for various diseases. However, the current body of knowledge is erroneous and incomplete. In order to find the missing links and correct the current state of knowledge, a combined consideration of these manifold data resources is essential. This thesis aims to provide the means to answer concrete domain-specific research problems, but at the same time addresses common Visual Analytics challenges, as the heterogeneous set of complex data comprises major challenges confronting the field of Visual Analytics: different levels of scale, different formats, different types and different sources.

The domain-specific research problems on the one hand motivated the solutions which were created in the course of this thesis, and on the other hand, serve to demonstrate their practical value. However, great care was taken to translate the domain-specific research problems to more general visual analysis problems – thus providing the broader context of the solutions and their possible applicability to other domains.

[6] `http://www.bbmri.eu`

Chapter 3

Problem Analysis and Related Work

Contents

The challenges of analyzing multiple heterogeneous data sets are widely recognized in the field of Visual Analytics. Thomas and Cook recommend the creation of "methods to synthesize information of different types and from different sources into a unified data representation" [Thomas and Cook, 2005, p.11]. Such a unified representation can be envisioned as an heterogeneous *information landscape*, in which information foraging and sense-making take place. Like a person exploring unknown territory in the real world, a user navigating the high-dimensional, multi-faceted and overwhelmingly large, combined data space of multiple heterogeneous data sets must be provided with some means of orientation. This challenge has been described in the context of information retrieval as early as 1993 [O'Day and Jeffries, 1993]. This publication also coined the notion of *information landscapes* and identified different strategies commonly used to gather information within them – *e.g.*, exploring the data in an undirected fashion or following a concrete plan for finding the desired information. While an effective visual analysis system has to support both types of analysis, it first needs to be clarified what the different kinds of support are which the user needs during an analysis.

3.1 Levels of Analyst Support

Two distinct levels of *analyst support* can be identified:

S1 **Orientation** that communicates

- *S1.1* the current position within the information landscape,
- *S1.2* the path of analysis steps that led there (history), and
- *S1.3* possible directions for further investigation (*e.g.*, related data sets).

S2 **Guidance** that suggests concrete analysis steps to be taken in order to get from an analysis hypothesis to an analysis result.

These support levels serve as reference points throughout this thesis. On the one hand, existing related work is allocated to the respective level and on the other hand, for the proposed concepts and techniques from this thesis it will be stated which of the levels is being addressed in particular.

3.2 State-of-the-Art in Supported Visual Data Analysis

Orienting a user in an analysis is a challenge on its own right. The support can be given in the **data space**, the **view space** as well as in **time**. Examples of techniques that provide support in the data space are semantic zooming [Perlin and Fox, 1993] or edge-based traveling for the exploration of graphs [Tominski et al., 2009]. Support examples in the view space are navigation techniques such as zoom/pan/rotate as well as general paradigms like focus+context and overview+detail. Many of these orientation techniques in space and also in time are already an integral part of visualization systems. However, depending on the size of the information landscape and the complexity of the analysis setup (see Section 3.4), a combination of temporal as well as data and view space orientation techniques is required. The novel visualization techniques introduced in Chapter 6 and 7 suggest such a combination for different domain problems with various degrees of complexity. For instance, the analysis of heterogeneous data visualized in many synchronized views demands another set of orientation techniques than the analysis of a single multi-dimensional data set. Thus, although many individual techniques have been proposed over the years, it remains difficult to build a system that successfully realizes orientation support.

In contrast, the related work that addresses guidance in the sense of recommending future analysis steps is rather confined. The following section discusses possible strategies in terms of input information that can then serve as a basis for realizing guidance support. On a conceptual level, the literature offers two prevalent strategies for guidance support: bottom-up, provenance-driven vs. top-down, model-driven. While the provenance-driven strategy bases the user recommendations on events and information collected during past analysis sessions, the model-based strategy builds upon authored workflows and further systematically associated information that can then be dynamically utilized for assisting the user.

3.2.1 Provenance-Driven Guidance Approaches

The most common strategy is the provenance-driven, bottom-up strategy, which gathers data and distills it into navigational cues. Based on the gathered information from past sessions, analyst support can in principle be realized for orientation purposes on the level *S1.3* as well as for recommending concrete next steps (*S2*). In the first case, a system can present to the user which next steps are possible by looking up what other users did at a certain stage during the analysis. The second approach that provides concrete suggestions for next steps is also well known from recommendation systems which are applied in various domains [Scheidegger, 2009].

An example for a provenance-based system is VisSheet which generates a number of previews for a range of possible visualization parameter changes and presents them to the user to choose from [Jankun-Kelly and Ma, 2001]. Another example is the provenance based approach of HARVEST's behavior-driven visualization recommendation which analyzes the user's analytic activity [Gotz and Zhou, 2009, Gotz and Wen, 2009]. The follow up work also extends the recommendations by presenting possibly interesting annotations from past analysis sessions [Shrinivasan and Gotz, 2009]. TIBCO Spotfire® Guided Analysis[1], a commercial product, follows a similar approach that captures "best practices", which then can be shared between analysts.

When employing the concept of Social Navigation, one also gathers user data, but crowdsources it from multiple users and displays it as usage statistics indicating popular or neglected user choices [Willett et al., 2007]. Similar provenance-driven techniques can be employed on a higher level as well, as it is done by the VisComplete system (part of VisTrails[2]) which mines a database of existing visualization pipelines to aid the user in constructing new ones by suggesting possible completions [Koop et al., 2008].

[Garg et al., 2008] utilize machine learning algorithms for aiding a user in the discovery of patterns in the data. Users provide the system with analysis patterns, which the system tries to learn. On the basis of the resulting model, the system then makes suggestions to the analyst.

[1] `http://spotfire.tibco.com/about/guided-analytics.aspx`

[2] `http://www.vistrails.org`

3.2.2 Model-Driven Guidance Approaches

An approach used more rarely than the provenance-driven strategy is the model-driven, top-down strategy, which derives navigational assistance by instantiating a predefined, abstract best-practice solution with concrete visual and computational techniques. One example is the Systematic Yet Flexible system, which gives a step-by-step guidance along a high-level workflow, while leaving the choice of concrete techniques that achieve the higher-level objectives to the user [Perer and Shneiderman, 2008].

The model-driven approach, presented in detail in the next chapter, falls into this second category, but has a much larger scope. In contrast to the state-of-the-art approaches that only capture the actual workflow, it also utilizes information about the analysis setup, which includes the available data sets and their interrelations, the available algorithmic and visual methods and packages, as well as their applicability to achieve individual steps of the workflow. As a result of making this additional information available, it is possible to automatically determine suitable analytical techniques and subsequently use this information to provide navigational cues on a more specific, lower level along a given analysis path. This proves especially useful in interactive systems that exhibit a large number of possible continuations at any given point – so that the decision which functionality to use on which part of the data and in which order is particularly challenging.

Having discussed the basic ways of capturing and externalizing the input information that lay the groundwork for the user support, the next section introduces the visual means that a system can employ to incorporate this information in the visual representation with which the user interacts during an analysis.

3.3 Means of Visual Communication for Analyst Support

In order to visually express orientation and guidance a series of communication strategies exist:

- Utilizing **visual cues** in existing visualizations
- **Meta-visualizations** as auxiliary views
- **Textual descriptions**

Each strategy for communicating support information implies a certain additional cognitive effort for the user. To keep the cognitive load to a minimum, the most effective and most efficient strategy needs to be chosen for each support operation. In the following, individual communication strategies are briefly discussed while also stating the suitability of each strategy for the levels of support defined in Section 3.1.

3.3.1 Visual Cues

Altering the property of a visual variable in existing visualizations (*e.g.*, increasing the size of an object) which is not encoding information on its own is a common method for drawing the user's attention to a specific region or object on the screen. Well suited for this purpose are variables that are selective according to [Bertin, 1983] – such as color, size and shape.

However, to orient a user in an information landscape *(S1.1)*, it is necessary to indicate relationships between elements. Synchronized highlighting, *i.e.*, concurrently changing the visual representation of the related elements, is a common way of achieving this. In combination with linking & brushing [Becker, 1987], this simple variant of visual linking is a powerful and well-established technique in information visualization.

Visual variables that are associative ([Bertin, 1983]), like color, size, orientation and hue, are suited for this purpose. Synchronized blinking is another strong visual attractor that could be used for showing relations, but is often perceived as disturbing [Seo and Shneiderman, 2002]. The final and most expressive alternative is to draw lines between the related data items – referred to as *visual links* in the remainder of this thesis.

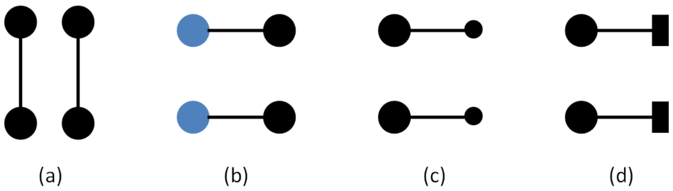


Figure 3.1: The connectedness overrules the Gestalt law principles: (a) proximity, (b) color, (c) size and (d) shape [Ware, 2004].

According to [Palmer and Rock, 1994], connectedness between elements is even a stronger grouping principle than proximity, color, size and shape, see Figure 3.1. Palmer and Rock suggest even extending the organizing principles described in the Gestalt laws [Ware, 2004] by the *connectedness* of objects. Furthermore, the trend to large, high resolution displays in combination with the small active visual field [Ware, 2004] is another argument for using visual links. For relating spatially distributed pieces of information, it is easier for users to follow visual links with the eyes than matching the color, shape or animation of two elements over a longer distance. In addition, color and shape do not scale well, as users cannot distinguish more than five to seven [Healey, 1996], making these a bad choice for simultaneously showing relations between multiple sets of elements. In contrast, visual links are only limited by the additional visual clutter they introduce on top of existing visualizations. However, this problem can be alleviated by bundling the lines according to additional structural information, like Hierarchical Edge Bundles do [Holten, 2006]. The utilization of visual links is also particularly useful for situations in which color already encodes an attribute of the data set. Summing up, visual connection lines are strong in supporting the perception of related objects and therefore well suited for orientation purposes.

In a node-link graph visualization, data dependencies are represented by edges (*e.g.*, drawn as lines) connecting the nodes in a network. Hence, they are part of a self-contained, single visualization technique that represents a homogeneous data set – the graph itself. Therefore, graph edges can be considered as the most simple example of visual links. Examples are cone trees [Robertson et al., 1991] for the visualization of hierarchies as well as the early work on SemNet [Fairchild et al., 1988], a 3D representation of a network.

While visual links are an inherent component of network/graph visualization, in the last two decades they have also become relevant for extended purposes in information visualization:

- **Adding extra relational information** on top of an existing visualization. Examples are Hierarchical Edge Bundles [Holten, 2006] as well as many other more specialized solutions, like Fekete's tree map overlays with graph links [Fekete et al., 2003] and also the NFlowVis system [Fischer et al., 2008].
- **Bridging multiple visualizations** resulting in a new compound visualization. An early example for the usage of visual links in this context is the work by [Risch et al., 1996] on data intelligence analysis in a virtual environment (see Figure 3.4). Semantic substrates [Shneiderman and Aris, 2006] add cross-edges between 2D views arranged side by side (*cf.*, Figure 3.2). Collins and Carpendale [Collins and Carpendale, 2007] later generalized visual links for interconnecting multiple visualization from different types and furthermore allowed users to arrange and navigate the linked views in an interactive 3D setup (see Figure 3.5). [Hoellerer et al., 2007] patented a concept that also makes use of visual links for depicting inter-view relations in 3D (see Figure 3.3). Recently, visual linking was used in [Viau et al., 2010] for showing interactive dependencies between their FlowVizMenu interface which dynamically parametrizes a connected scatterplot matrix.

While straight connection lines are the most common shape of links, curves, ribbons and surfaces are possible as well. The decision which style to use depends on the content of the visualization. For instance, for visualizations with mainly symmetric content, curved connection lines stand out and are therefore easier to discriminate [Hoffmann et al., 2008]. In addition, elements connected by smooth (curved) lines are easier to perceive as being related [Ware, 2004, p.193].

As visual links are rendered on top of existing visualizations, they carry the risk of visual clutter and thus should not be overused. Clever routing of links that takes into account the information underneath is one countermeasure for minimizing this visual clutter – as identified in [Shneiderman and Aris, 2006] as potential direction for further research.

Some of the novel visualization techniques proposed in this thesis employ visual links between views for providing orientation. For details see Chapter 6, 7 and 8.

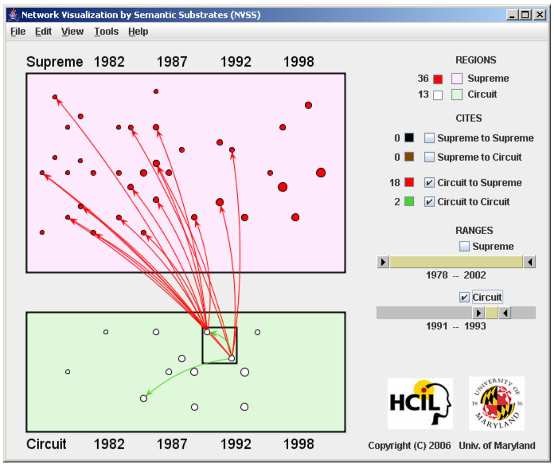

Network Visualization by Semantic Substrates (NVSS)
File Edit View Tools Help
Supreme 1982 1987 1992 1998
Circuit 1982 1987 1992 1998
REGIONS
36 🟥 □ Supreme
13 □ □ Circuit
CITES
0 ■ □ Supreme to Supreme
0 ■ □ Supreme to Circuit
18 🟥 ☑ Circuit to Supreme
2 🟩 ☑ Circuit to Circuit
RANGES
□ Supreme
1978 -- 2002
☑ Circuit
1991 -- 1993
HCIL
Copyright (C) 2006 Univ. of Maryland

Figure 3.2: Network visualization by semantic substrates [Shneiderman and Aris, 2006]. Semantic substrates are non-overlapping regions where nodes are placed according to node attributes. The user can specify regions, for which all edges connected to the enclosed nodes become visible.

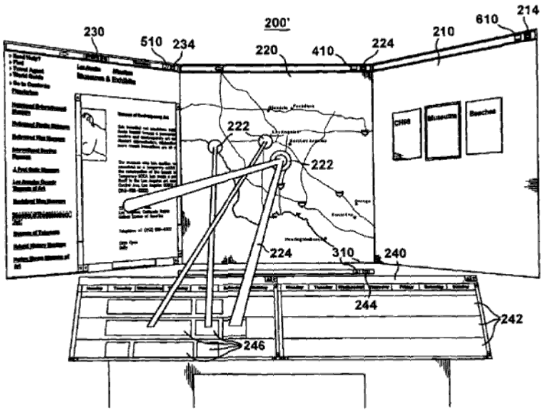


Figure 3.3: A setup patented by Microsoft that interconnects visual representations in a statically arranged information workspace [Hoellerer et al., 2007]. An additional data attribute is encoded in the thickness of the lines.

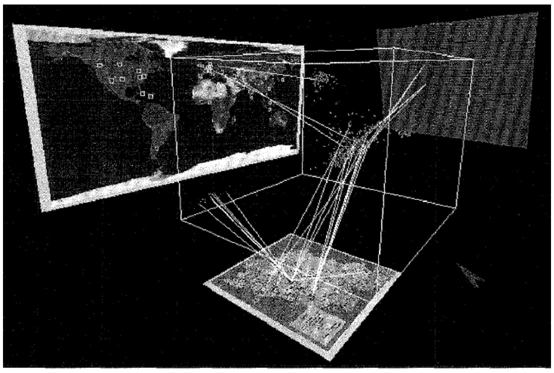


Figure 3.4: Cross referencing of information in a virtual environment [Risch et al., 1996]

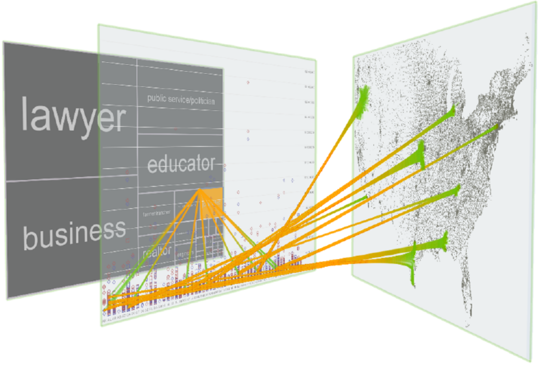


Figure 3.5: VisLink method propagating inter-plane edges between existing visualizations [Collins and Carpendale, 2007]

3.3.2 Meta-Visualizations

In contrast to standard visualization techniques that render the actual data set, meta-visualizations aim at providing additional information about the data, the setup or the current analysis session. Two kinds of meta-visualizations can help to establish orientation:

- a history graph (*e.g.*, [Kreuseler et al., 2004, Shrinivasan and van Wijk, 2008, Heer et al., 2008]), showing previous analysis steps (orientation level *S1.2*), and
- an explicit, abstract representation of, for instance, the data sets and their interde-

pendencies. This meta-visualization can serve as the basis for orientation on level *S1.1*, *S1.3* and *S2*.

For both cases it makes sense to show only the part of the graph that is of current interest for the analyst. A semantic reduction of the graph information is also a useful mechanism to present minimal yet sufficient pieces of information for the current analysis situation.

3.3.3 Textual descriptions

In principle, it would be possible to describe every support action textually. As textual descriptions demand the full attention from users for a short period of time, this is not optimal for simple cases of orientation, where the highlighting of a button or specific region of the visualization is sufficient. However, for guidance in the sense of *S2*, text is essential for communicating step-by-step instructions which could also be formulated in domain-specific words tailored to the analyst's language and background.

Presenting text is a necessary but not sufficient tool for providing guidance. A well thought-out combination of all three support communication strategies is therefore required. The Stack'n'flip system presented in Chapter 8 uses one possible combination that is specifically targeted at realizing guidance – while also providing orientation at any time during the analysis.

3.4 Analysis Setup Characteristics

The complexity of creating an analysis system that offers the support under discussion is influenced by a series of setup characteristics. It is relevant whether an analysis comprises one or multiple: **data sets**, **views**, **applications**, **displays** and **users**.

Figure 3.6 depicts a sample single-user analysis setup where multiple instances of each individual characteristic are involved. The higher the degree of heterogeneity is in terms of these variables, the more important support becomes. At the same time, support also becomes harder. The following section discusses the influence of each of these aspects on supported visual analysis.

3.4.1 Within / Across Data Sets and Views

When discussing orientation in an information landscape, it is necessary to consider the data sets and their visual representations at the same time. In this respect, a series of combinations are possible – each of them imposing different requirements regarding the difficulties for maintaining a user's mental map. The easiest case is a single data set, represented by a single view. Here, the relations within the data set are communicated by the visualization technique itself, *e.g.*, a tree map [Johnson and Shneiderman, 1991], transporting the hierarchical relations between elements. If the technique is well designed

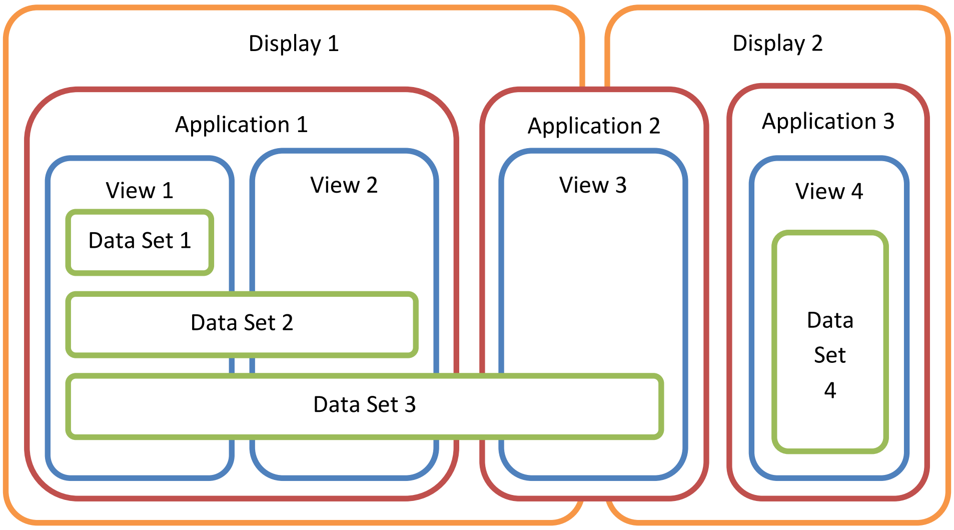


Figure 3.6: A sample single-user analysis setup covering multiple data sets visualized in multiple views. The analysis runs in a series of applications spanning multiple displays (monitors).

and suited for visualizing the data set, no additional measures need to be taken for keeping a user oriented. Multiple views showing different aspects of the same data set by using various visualization techniques are also easily manageable for a user in terms of cognition.

However, when multiple heterogeneous data sets are the subject of an analysis, this is not as straightforward. The state-of-the-art solution is to employ multiple coordinated views [Roberts, 2007], where the user can interactively explore how the data sets are linked. Nevertheless, it remains difficult for a user to understand the association between the data sets and the respective views as well as the interrelations between the data sets at any time [Baldonado et al., 2000].

Chapter 6 introduces visualization techniques which take the first step towards orientation in such a heterogeneous data analysis scenario. These techniques rely on visual links as an additional visual cue – making the relations within and between the data sets on the level of individual items more explicit. However, although these techniques have proven their value for investigating a low number of distinct data sets, they do not scale well to a complex, interwoven information landscape. This requirement is addressed by the Stack'n'flip system presented in Chapter 8. This technique incorporates a meta-visualization that presents the relations between the data sets in an explicit manner, the history (*i.e.*, the data sets the user has already visited during the analysis), and also the possible next data sets that are connected to the currently investigated one. In addition, the visualization makes it clear which data set is currently being visualized by which view(s). Thus, Stack'n'flip enables orientation in all of its facets (*S1.1*, *S1.2* and *S1.3*).

3.4.2 Within / Across Applications

State-of-the-art visual analytics systems allow users to load multiple data sets which can then be analyzed in a multiple coordinated view fashion. However, creating one single super-application which supports users in all their analysis needs is unrealistic. In real world situations, a profound visual analysis often comprises multiple, highly specialized and expensive tools. An example from clinical data analysis is to use a database query tool for finding patients with common characteristics with respect to a disease in the first place, and then load the MR data of these patients in a volume viewing tool for an in-depth comparison.

However, like in this example, in most real world analysis scenarios intercommunication between separate tools, if existent, is limited. For instance, an import/export feature cannot provide the same level of support as a well integrated seamless analysis workflow carried out in a single application. Consequently, there is a strong need for bridging the gaps between these independent tools. The three-stage model provides the foundations for doing so, as it is irrelevant which set of applications offer the visual and computational interfaces.

What remains is to solve the technical problems of such a multi-tool scenario. In the spirit of the Snap-Together Visualization [North and Shneiderman, 2000], possibilities to orient a user (support level *S1*) by visually linking information across applications [Waldner et al., 2010] have been explored, as presented in detail in Section 7.1. In contrast to Snap-Together, the proposed approach works without a common database on the basis of ID-Strings that are collected from minimally-modified applications (*e.g.*, via plug-ins) and matched by a light-weight management application. The related entities thereby identified are connected by visual links. While this approach is a first step towards orienting a user in an application-spanning analysis, model-driven guidance along a workflow in such a scenario is a promising next step, as described in Section 9.2 on future work.

Note that distributed Visual Analytics systems (*e.g.*, [Natarajan and Ganz, 2009]) do not span independent tools and therefore do not fall into the across-application category.

3.4.3 Within / Across Analysts

Collaboration is another important building block for solving complex domain problems. For a comprehensive analysis, experts from multiple domains with different background knowledge are beneficial. Initial results in the direction of collaborative Visual Analytics are already available. For example, [Heer and Agrawala, 2007] discuss possible design considerations in this context, [Brennan et al., 2006] present a multi-analyst Visual Analytics framework and [Isenberg and Fisher, 2009] realized collaborative linking and brushing in such environments. Although numerous research activities address various aspects of the problem area, to our knowledge no work exists that aims for a systematic externalization of such multi-analyst scenarios. Thus, the proposed three-stage model-driven concept has the potential to serve as a basis for supporting these collaborative setups.

Collaborative data analysis is not in the main scope of this work. However, aspects of this work can be and have been applied to such scenarios [Waldner and Schmalstieg, 2011]. Section 7.2 introduces a prototype implementation [Waldner et al., 2009], where experts from different domains can jointly perform a co-located analysis. The proposed solution is again targeted at the first stage of support – orientation.

3.4.4 Within / Across Displays

It lies in the nature of information visualization that the available number of pixels is a restricting factor. Even for medium-scale data analysis scenarios, the number of data items exceeds the available pixels for visualizing them. Abstraction techniques are one way to handle the problem – another way is to extend the available screen real estate. The spectrum ranges from a high resolution multi-projector system to off-the-shelf desktop setups. Two large monitors arranged side by side are already regarded as quasi standard for information workers.

The normal work environment of expert users that deal with biomedical data are clinical facilities and laboratories where technical resources are limited. Only a few of them will have access to expensive multi-projector facilities. Nevertheless, visual analysis in such an environment can add value – in particular for cases where multiple experts need to collaborate to reach a desired analysis goal, as discussed in the previous section.

Although the vast amount of pixels available for visualization is tempting for data analysis purposes, the challenges in terms of orientation as well as guidance become even more complicated (*e.g.*, related elements are further away from each other). In addition, human computer interaction (HCI) aspects need to be considered in these multi-display scenarios. Besides the issues concerning HCI and information visualization, the technical realization of multi-display environments is a field of research on its own. Section 7.2 introduces Caleydoplex, a prototype multiple-display environment developed at the Graz University of Technology. However, the consideration of the setup in this thesis does not focus on the technical aspects, but on the implications for collaborative visual analysis and how the model-driven approach can help in this respect.

3.5 Categorization of Related and Own Work

Up to this point, this chapter has considered analysis support in all its aspects: the various kinds of support, what it takes to be able to aid a user in terms of input and how this support can finally be conveyed during an analysis. Taking this as a fundament, this section classifies the state-of-the-art in supported visual data analysis as well as the techniques proposed in this thesis, which are introduced in detail in the following chapters. This systematic classification not only informs the proposed techniques, but also makes it possible to identify remaining empty spots that indicate future research directions in our field. Table 3.1 lists various systems, either published scientific work or commercial systems, that cover at least one of the support levels from Section 3.1. In addition, the table classifies whether a system covers across-application, across-display or across-analyst support.

Looking at the table, it becomes obvious that basic support on the orientation level, be it within the information landscape and/or in the analysis history, is already a part of many systems. Also, numerous systems cover data analysis including multiple data sets and multiple coordinated views. The Matchmaker approach (introduced in Section 6.1.2), the Jukebox (Section 6.2.2) and the Bucket (Section 6.3.3) are all targeted at orienting users in an analysis by visual means. While the Matchmaker is a visualization technique for the comparison of multi-dimensional, quantitative data, the Jukebox as well as the Bucket are techniques to analyze multiple related data sets in a multiple coordinated view fashion.

The Snap-Together Visualization [North and Shneiderman, 2000] as well as the Visual Links Across Applications system, introduced in Section 7.1.1, are the only representatives in the category across-applications. Systems which allow across-user analysis are for example LARK [Tobiasz et al., 2009], Cambiera [Isenberg and Fisher, 2009] and also the Caleydoplex setup (*cf.*, Section 7.2.3). Collaborative Information Linking [Waldner and Schmalstieg, 2011] (see Section 7.3) combines the strengths of multi-user analysis with the preliminary work on visual links across applications.

However, all these systems address orientation in an information landscape, but not guidance where a system assists users by suggesting future steps. Only the lowest quarter of the categorization table lists systems which fall into this category. Guidance is either based on information captured during past analysis sessions or on an authored model. While Vistrails [Bavoil et al., 2005], HARVEST [Gotz and Wen, 2009] and TIBCO Spotfire use history information, the Systematic Yet Flexible [Perer and Shneiderman, 2008] and the proposed Stack'n'flip system (Chapter 8) utilize a previously defined model. As mentioned before, the Systematic Yet Flexible system suggests future steps along predefined workflows. However, due to the three-stage model, Stack'n'flip can even go a step further by also suggesting which interface to apply on which data set for each of the domain-specific tasks.

Summing up, analyst assistance for complex data analysis scenarios is only sparsely addressed in the literature. In particular, guidance across multiple applications, across multiple displays and across multiple analysts are gaps that need to be filled by future research.

3.6 Summary

This chapter started by pointing out the users' need for closer support during the analysis of complex, heterogeneous data sets in order to reduce the risk of disorientation. Providing a

Table 3.1: Classification of some Visual Analytics frameworks and techniques according to the level of support they provide and the setup characteristics for Visual Analytics applications identified in Section 3.4. In addition to the systems known from literature and Visual Analytics products already on the market, the methods that are part of this thesis (in bold) are listed as well.

	Orientation in Info Landscape	**Orientation** via History	**Guidance** Provenance-based	**Guidance** Model-based	**Across Data Sets**	**Across Views**	**Across Applications**	**Across Displays**	**Across Analysts**
Matchmaker (*cf.*, Section 6.1.2)	■								
VisLink [Collins and Carpendale, 2007]	■				■	■			
Jukebox (*cf.*, Section 6.2.2)	■				■	■			
Bucket (*cf.*, Section 6.3.3)	■				■	■			
Image Graphs [Ma, 1999]		■				■			
History View [Kreuseler et al., 2004]	■	■				■			
Tableau (former Polaris [Stolte et al., 2002])	■	■			■	■			
Snap-Together [North and Shneiderman, 2000]	■				■	■	■		
Visual Links Across Apps (*cf.*, Section 7.1.1)	■				■	■	■		
LARK [Tobiasz et al., 2009]	■					■			■
Cambiera [Isenberg and Fisher, 2009]	■				■	■			■
Caleydoplex (*cf.*, Section 7.2.3)	■				■	■		■	■
Collaborative Information Linking [Waldner and Schmalstieg, 2011]	■				■	■	■	■	■
HARVEST [Gotz and Zhou, 2009]	■		■			■			
TIBCO Spotfire® DecisionSite Guided Analyt.	■		■		■	■			
VisTrails [Bavoil et al., 2005] and VisComplete [Koop et al., 2008]	■	■	■			■			
Systematic Yet Flexible Discovery [Perer and Shneiderman, 2008]	■			■	■	■			
Stack'n'flip (*cf.*, Chapter 8)	■	■		■	■	■			

feature-rich user interface for accessing computational and visual methods is not sufficient. A system that assists users during an analysis can support them on several levels. First, it is essential to keep a user **oriented** within the information landscape and the features of an analysis system. With orientation as a precondition, a system can dynamically suggest concrete future steps – realizing **guidance**, as the second and more comprehensive level of support.

After elaborating on the possible kinds of support, the chapter continued with discussing existing work that aims at orientation support and guidance. For orientation support, the state-of-the-art offers several techniques that assist users in terms of data and view space as well as time. The combination of the techniques required depends on the complexity of the information landscape and the analysis setup. For realizing guidance support, which offers concrete suggestions of future analysis steps, the approaches known from literature can be split into two basic approaches. First, systems that provide assistance based on provenance information and secondly, approaches that rely on a model that comprises predefined tasks and workflows. While the first strategy is limited to matching gathered information from previous analysis sessions to suggest future analysis steps, the second strategy has more potential, as it can guide a user towards a concrete analysis goal. However, the current state-of-the-art only uses workflows but without telling a user which tools to use and actions to perform in order to get there. Thus, a model is needed that comprises not only the workflow and tasks, but also contains the associations to the data on which the tasks should be performed and also which visual or computational tools can be applied. The next chapter proposes such a unified representation that captures the interplay between data, view, and task which lays the theoretical foundation for realizing the support in a concrete visualization system.

Irrespective of these basic strategies on which the support can be based, various ways exist to convey the modeled or captured information to an analyst. The spectrum ranges from using visual cues on top of existing visualizations, to meta-visualizations, to descriptive text. However, the choice of the appropriate means of communication is influenced by the characteristics of the analysis setup: whether the analysis comprises multiple data sets and views, multiple applications, multiple displays or also multiple analysts. With these distinctive features in mind, the chapter is concluded by contrasting the current state-of-the-art in supported visual analysis with the novel visualization techniques that are presented throughout the thesis.

Chapter 4

Three-Stage Model-Driven Design Process

Contents

Large information landscapes with multiple heterogeneous data sets and numerous visual and computational interfaces for accessing them require means of support to ensure their timely and accurate analysis. Providing such user support is not a trivial task, as the degree of support required by the user may vary during the analysis session. The user may need concrete guidance during one part of the analysis session and only orientation support during other parts. To realize such a smooth back and forth between these two levels of support, a visual analysis system must have considerable knowledge about the available data sets and their relations, the goals of its user, as well as the analytical capabilities. Figure 4.1 illustrates this transition from input to output. However, finding a structured way of creating the supportive visual representations is the central research question that is addressed in this chapter. Various kinds of input can be potentially employed: the input data and its corresponding relations, contextual information about the available analysis framework, as well as domain-specific knowledge about a concrete analysis goal and high-level tasks for reaching this goal. Note that in a feedback loop, the user himself can feed additional information or also dynamically alter existing input information.

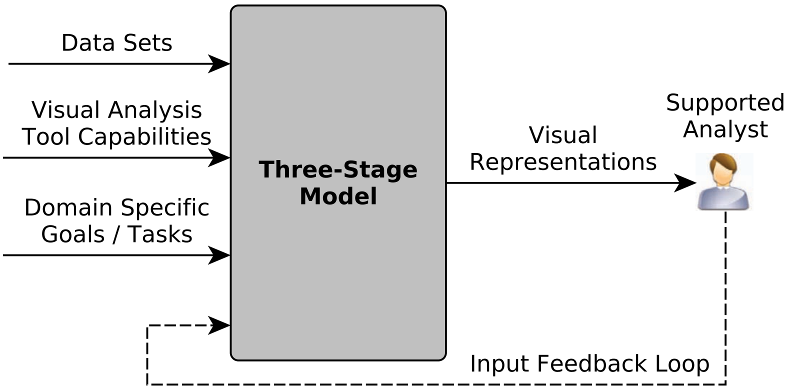


Figure 4.1: Input/output transformation for supported visual analysis. The proposed model-driven design process aims to fill the black box that converts the input information to a visual representation which realizes the analyst support.

The presented model-driven design presented here aims at realizing this transition from input to output. The concept draws upon first ideas from the position paper [Streit et al., 2009c]. It encapsulates the input information in three models:

- A domain-independent **model of the setup** in which the interactive visual analysis takes place – describing the data sets, the visual and computational interfaces to the data, and the different analytical operations that can be performed with them.
- A **model of the domain** that captures what can be done with a given setup in the context of a specific domain – describing the numerous domain-specific tasks and relating them to the data sets and analytical operations of a given setup model.
- A **model of the analysis session** that lists what has to be done to pursue a given analysis goal – describing the analysis workflow as a sequence of domain-specific tasks from a given domain model.

The knowledge specified by these models requires an authoring phase in which the models are put together. It is obvious that the overhead of such an elaborate modeling phase is not justified for straightforward setups with a manageable complexity. However, with increasingly complex analysis scenarios, the benefits soon outweigh the initial modeling costs. This is especially true for highly repetitive analysis sequences, which have to be modeled only once and can be reused over and over again. For such routine tasks, the guidance ensures that every repetition is done with the same care as the very first analysis and without forgetting a crucial intermediate step. A guided analysis thus provides a high degree of reproducibility and traceability, which makes most sense for application fields in which a faulty analysis may lead to dire situations, such as the diagnosis of patients or the analysis of safety hazards in airplane inspections. Nevertheless, if the user wants to deviate from the workflow of a guided analysis to freely roam the information landscape in a more explorative, unplanned fashion, he can do so at any point, resulting in a fall-back from guidance to orientation support. Transitioning back from such an exploratory side step onto the planned analysis path allows the analyst to continue with step-by-step instructions again.

The setup model is authored once and needs to be adapted or extended only when new data sets or tools become available. With this underlying, domain-independent model, different domain models can be associated, as different application domains may use the same setup to carry out the analysis. This can be frequently observed, *e.g.*, in the field of life sciences, where a geneticist and a biochemist may use the same data sources and interfaces, but perform completely different tasks. In the last step, a concrete analysis workflow is formulated, which is then tailored to the availability of data and analysis methods for a given case, by pruning tasks that cannot be performed. This yields a streamlined analysis workflow, which contains only those analysis paths that can be realized with the given data and tools. The next subsection outlines the overall authoring process together with the different roles involved in each individual authoring step.

4.1 Overall Authoring Process and Involved Roles

The description of complex, possibly cross-domain analyses requires a good deal of expertise in all fields involved. As the assumption of an omniscient expert is unrealistic, different roles for the authoring of the different models were elicited. Table 4.1 lists the three roles involved in the authoring, as well as two possible roles for analysts using the models.

The process of authoring the different models is best described as a step-wise procedure that sequentially adds to the complexity of the models until they are fully specified. This is shown schematically in Figure 4.2. The authoring process consists of the following sequence of steps, each being the responsibility of one of the three expert roles from Table 4.1:

I Developing the data model: this is the responsibility of the **data manager**, who describes the data sets and their interrelations. Especially in larger organizations, dedicated data managers are often employed – *e.g.*, clinical data managers in hospitals.

Role	Description	Category
Data Manager	responsible for building and maintaining the data model	author
Visual Analysis Expert	responsible for compiling interfaces and their operators	
Domain Expert	responsible for compiling tasks and analysis workflows	
Informed Analyst	works on open research questions with no predefined analysis workflow	user
Guided Analyst	works on answering a routine question along a predefined analysis workflow	

Table 4.1: Roles in authoring and using models of setup, domain and analysis session.

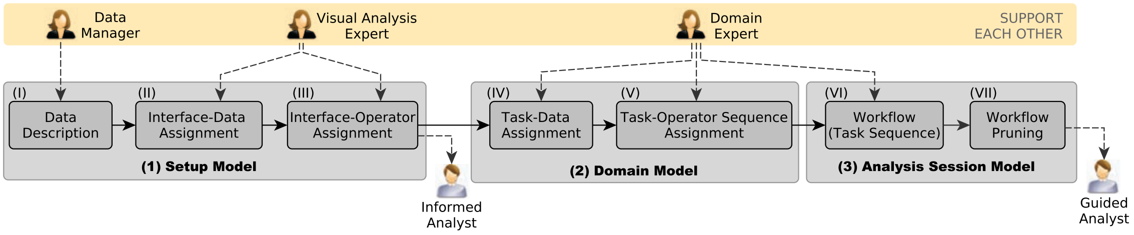


Figure 4.2: The authoring process shown as a sequence of authoring steps (I-VII) carried out by data, visual analysis and domain experts. Depending on the complexity of the use case and the analysis goals, experts can assist each other. Also, multiple roles can be fulfilled by one person. The authored models can be taken as aids for providing analyst support on two levels: the setup model (1) for informing (providing orientation to) a user within the sets of data, computational procedures and visualizations, and the domain model (2) on which the analysis session model (3) bases for guiding (providing step-by-step directions) the user.

II Enriching the data model with interfaces: this is done by the **visual analysis expert**, who annotates the data sets with information about what infrastructure is available to access each of them – via graphical interfaces (visualizations) or through computational interfaces (query languages, statistics packages).

III Compiling a list of operators for each interface: this lies within the responsibility of the **visual analysis expert**, who denotes which interface is suitable to perform which operations, as some interfaces may be more fitting than others. For instance the operation "clustering" is best done with a statistics package and not with a plain SQL interface.

IV Connecting tasks to the data model: for this, the **domain expert** identifies the required data sets for each of the high-level analysis tasks that are commonly performed in a given scenario and relates them to the task.

V Associating operators with the tasks: this is specified by the **domain expert** who links concrete operators to carry out the given tasks on the associated data. As the operators are domain independent, the translation from domain-specific tasks to operators should be supported by the **visual analysis expert** who contributes knowledge about suitable analysis methods.

VI Specifying a workflow of analysis tasks: in this step, the **domain expert** details concrete analysis sessions for pursuing a given goal by defining an analysis workflow using the tasks defined.

VII Pruning the workflow according to the actually available data sets and tools: as a final step, it is **automatically** determined, which paths within the workflow cannot be performed for a concrete instance of data and analysis tools. These are then pruned from the workflow.

The first three steps of this process describe the rather static setup of the analysis: data sources, ways to access these data sources and analytical operators to run on them. Steps IV and V concern the domain model, as they add the domain-specific tasks on top of the setup model. The last two steps connect these tasks to meaningful analysis sessions and prune these sessions to use only the data and tools available at analysis time.

Two different roles of users can benefit from the explicitly modeled setup and analysis session. The first is the **informed analyst** who analyzes the data freely, without following a predefined analysis path. For the informed analyst, the key benefit is the provision of orientation on level *S1.1, S1.2* and *S1.3*, which allows him, throughout the entire exploration process, to pinpoint exactly which part of the information landscape is currently under investigation, which methods are available to analyze this particular information and which other parts of the information landscape may be related and thus be of interest. For informing an analyst, only the model of the setup with all its data sets and different visual and computational operators is needed.

The second role is the one of a **guided analyst**, who follows a given analysis path and possibly conducts similar analyses routinely. The guided analyst benefits from the formal model of the analysis session, as it provides exactly the step-by-step guidance in the sense of support level *S2* on how to pursue an analysis path to achieve a given analysis goal with the data at hand. If necessary, the guided analyst may also deviate from the proposed workflow, in which case the user's role switches to informed analyst.

It should be noted that a one-to-one mapping of a specific person to a role is not required. Depending on the use case and its complexity, the responsibilities of one role can be performed by multiple individuals. Also, one person can fulfill multiple roles – for instance, the domain expert may fulfill one or both user roles. It is also possible to further extend or subdivide the suggested roles, for example with more concrete user profiles for specific applications.

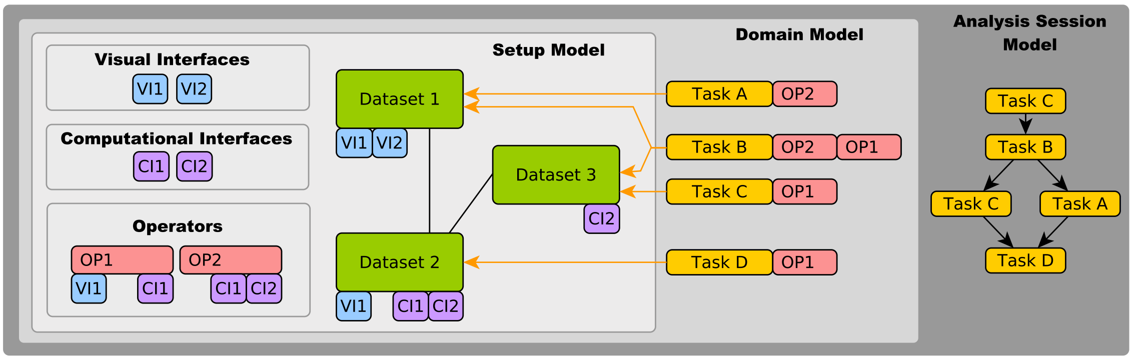


Figure 4.3: The different parts of the setup, the domain and the analysis session: interfaces (blue and purple), operators (red), data sets (green) and tasks (yellow). These parts are described and interrelated during the authoring process. An example of a fully authored model of analysis setup, analysis domain, and analysis session is shown in Figure 4.8.

Having sketched the overall authoring process, it remains to detail the individual authoring steps and how they build upon one another.

The models are not targeted towards orientation and guidance per se, but can potentially be used to optimize all kinds of processes, such as the treatment of missing data or collaborative analysis, as envisioned in [Streit et al., 2009c]. This generality is a strong argument for such a comprehensive authoring approach. To reflect the clear distinction between the general models and their specific application, the following explanation of the authoring steps is general as well. Nevertheless, the domain of biomedicine is used to give examples.

4.2 Authoring the Setup Model

The setup model is the first of three stages in the overall authoring process (see Figure 4.4). It captures the basic infrastructure in which the analysis takes place. Besides all the different data sources being available (Step I), this includes the software infrastructure for accessing the data (Step II), as well as the available software tools, such as visualization frameworks or statistics libraries, for analyzing the data (Step III).

4.2.1 Step I: Developing the Data Model

The data model captures all data sets (shown green in Figure 4.3) available in an analysis setup. This can include local data sets (*i.e.*, an electronic patient file), data sets available from online databases (*i.e.*, pharmaceutical lists or digital anatomical atlases), streamed data (*i.e.*, a patient's vital signs coming from intensive care), *etc.* In addition, the different data sets contain different types of data, such as imaging data from body scans, gene expression data from micro-array analyses, text data from electronic documents, *etc.*

When addressing complex problems, it is often essential to consider multiple

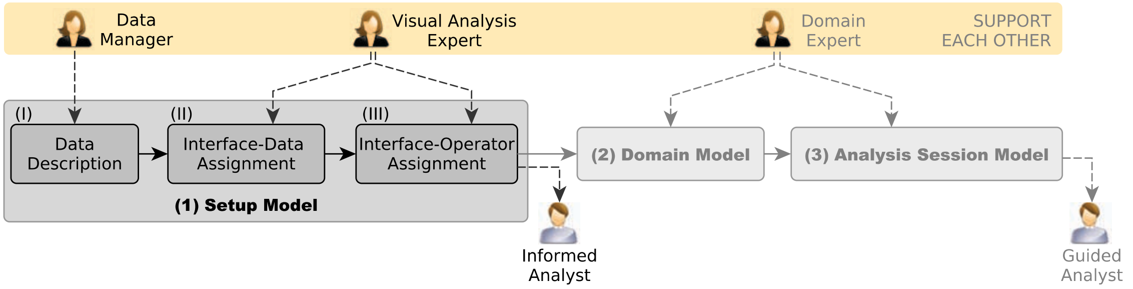


Figure 4.4: First stage of the model-driven design process: the analysis setup model, comprising step I-III of the overall process and building the basis for orientation support.

different levels simultaneously. The semantic dependency between the levels could be based on the scale of the data, their organization, explanation and also on observation [Ahl and Allen, 1996]. This means that the individual data sets can be assigned to these levels, which as a whole form a natural hierarchy. This inherent multi-level aspect is not a special case, but can frequently be observed in all kinds of domains. The hierarchy example from biology, already mentioned earlier in Section 2.1, is only one; others are the assembly hierarchy of a whole network of electronic devices down to the individual logic gate in the field of electrical engineering [Andrianantoandro et al., 2006] or the refinement process in software engineering from the specification documents down to the actual code. Figure 4.5 shows example hierarchies from the field of biology and electrical engineering. Note that one concrete hierarchy for a special domain is not generally valid, as it always represents only a certain perspective on the data, often driven by a particular research question in mind. Hence, a different research question can result in a slightly adapted or even radically changed hierarchy.

These multi-level dependencies are of great value for data analysis, as the transitions between adjacent levels indicate seamless analysis paths with a reduced mental effort for the user. However, the natural hierarchy is often not flexible enough to serve as a data model itself. In many cases additional cross-references between data sets, associated to non-adjacent levels, are needed. Therefore, a data model structured as a general graph is better suited for capturing the relations between the data sets. However, to a certain extent an underlying natural hierarchy will always be reflected in the full data model.

The data sets are related via common keys or identifiers where this is possible. In the biomedical use case, this can be for example the patient's name or social security number, thus identifying a patient's records across different data sets. In the case of different conventions being used for identifiers among multiple data sets, an ontology can often be used to map them. An example for this case is the mapping of gene and protein names using the Gene Ontology [Ashburner et al., 2000].

A data model of this sort is commonly used to plan and implement the combination of large database collections [Marrs et al., 1993]. Large organizations, such as hospitals,

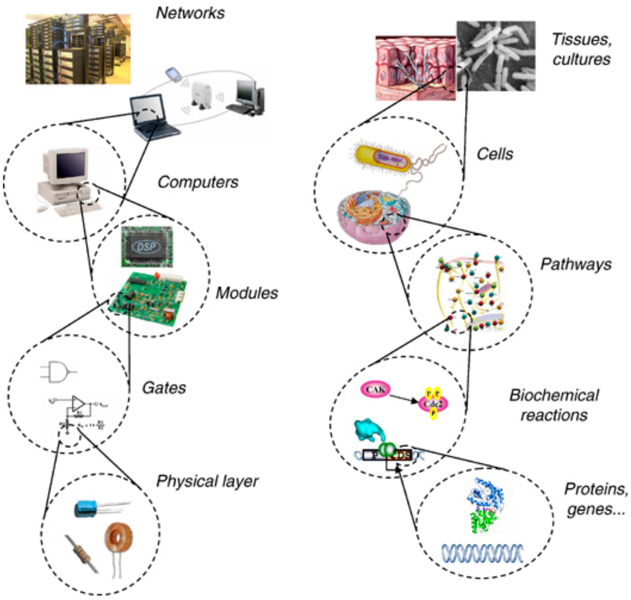


Figure 4.5: Natural multi-level hierarchy examples for the field of electrical engineering on the left and biology on the right, as presented in [Andrianantoandro et al., 2006]

usually have employees dedicated to define and refine such models, to validate and cross-reference entered data, and to supply necessary meta-data. Hence, many larger setups and even many freely available data collections, such as `linkeddata.org` or `data.gov`, do already have a data model of some sort. Yet beyond the pure organization of data sets, such data models are rarely used. A first approach utilizing a data model for visual analysis was only recently given in [Lieberman et al., 2010]. They use well-established, standard data models (*e.g.*, ERM [Chen, 1976]), which the proposed approach relies on too. This makes it easy to reuse or adapt existing data models for the setup model.

4.2.2 Step II: Enriching the Data Model with Interfaces

A first step to enhance the data model beyond what is stored is to add information about what is available in terms of infrastructure to access each data set. The access is conceptually performed through interfaces, which can be

- **Computational Interfaces** (purple in Figure 4.3) that fetch the data either directly from the source (low-level, query interfaces – *e.g.*, SQL or MapReduce [Dean and Ghemawat, 2008]) or calculate derived data, such as clusterings or correlations (high-level, algorithmic interfaces – *e.g.*, R statistics toolkit or WEKA)
- **Visual Interfaces** (shown blue in Figure 4.3) allowing for access using interactive, graphical methods, such as visual queries or query by example. Examples are scatterplots or parallel coordinates.

These interfaces are provided by the software infrastructure of the analysis setup – database front-ends, statistical libraries, visualization frameworks, *etc.* As different types of data require or permit different interfaces, the information about which method of access is available for each data set is added to the data model. This is done through one-to-many assignments, as a data set may require a combination of multiple visual interfaces to be properly displayed, or as an algorithmic interface may need several data sources to derive the desired information.

4.2.3 Step III: Compiling a List of Operators for the Interfaces

After having defined what to access by means of different interfaces in Step II, Step III focuses on how to access it. Operators (shown red in Figure 4.3) are domain-independent analysis actions that describe in general terms what each available interface can be used for. For example, an SQL interface is perfect for querying individual data items, a statistics library is well suited for correlation analyses and clustering, and a parallel coordinates view is ideal for interactive filtering.

The list of operators for each interface is usually based on the experience of the visual analysis expert, as well as on domain-specific conventions and recommendations from the literature. Hence, this step encodes common knowledge and the current state of research in the field of Visual Analytics in general.

This completes the modeling of the setup. It effectively describes the information landscape, and computational as well as visual access methods, as they are needed for guiding the informed analyst.

4.3 Authoring the Domain Model

The domain model adds a layer of domain-dependent knowledge on top of the setup model, see Figure 4.6. It does so by associating tasks being formulated in terms of the domain with the appropriate data (Step IV) and operators (Step V).

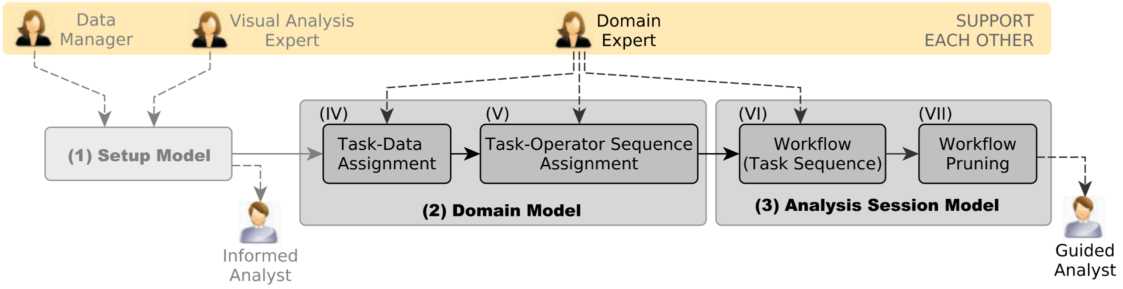


Figure 4.6: Stage two and three of the model-driven design process: the domain and analysis session model are added on top of the setup model.

4.3.1 Step IV: Connecting Tasks to the Data Model

As Munzner points out, the term “task” is overloaded in the visualization literature [Munzner, 2009]. Hence, it should be made clear that the term “task” is being used here for domain-dependent, textual descriptions of what an analysis step should achieve on which data set. An example of a domain-specific task is “Find all patients with a common characteristic”. At this stage, tasks (yellow in Figure 4.3) are described and linked to the data sets they are performed on. In the example given, patient characteristics may be scattered across multiple data sets. As no concrete characteristic is specified, the task would be connected to all of these data sets.

Tasks are closest to the actual analytical process and describe, in the words of the domain expert, what is being analyzed with which goal. They are used later as the building blocks of analysis sessions.

4.3.2 Step V: Associating Operators with the Tasks

While Step IV models what to do with which data set, Step V finally defines how to do it, in order to actually be able to carry out a task. This is achieved by mapping the tasks to the domain-independent operators. The mapping can either assign a single operator or a few operators to be carried out subsequently. Otherwise, in the case of tasks getting too complex, they can always be broken down into multiple, more fundamental tasks. In the case of the example task “Find all patients with a common characteristic”, this would be a single *filter* operator that filters the data set of patients by the given characteristic. If a data set provides multiple interfaces to perform the filter operator with, *e.g.*, an SQL interface and a parallel coordinates visualization, then the task is connected to all of these operators provided by the different interfaces. Which one to choose is for the user to decide.

This completes the modeling of the domain. It effectively yields a graph that connects data sets with tasks via domain-independent operators, *cf.*, the sample graph in Figure 4.3. The domain model bridges the domain-dependent analysis steps and the domain-independent analysis setup. The last authoring steps define the missing workflows on top of the domain model.

4.4 Authoring the Analysis Session Model

Often, an analysis session is seen as being equivalent to performing a sequence of analytical tasks. Yet in the proposed concept, analysis sessions are more abstractly defined, also capturing different analytical possibilities, in order to ensure their re-usability for other instances of data (*e.g.*, other patients). Specifically, an analysis session model consists of two parts: the actual analysis workflow (Step VI) and the constraints imposed on the workflow due to unavailability of data sources or analysis tools (Step VII).

4.4.1 Step VI: Specifying Workflows of Tasks

This authoring step assembles analysis workflows using the available tasks as building blocks in whichever order they are needed. In addition to simply appending tasks in a purely sequential order, Step VI also makes it possible to model more complex analysis patterns than a linear, step-by-step composition of tasks. In order to capture the involved and convoluted nature of analysis, branching, looping and forward jumping is possible as well – in the very same spirit, as task models [Stary, 2000] or user-task models [Puerta, 1997] are authored in the field of interface design.

The analysis workflows are modeled as directed graphs with tasks as nodes and edges as transitions from one task to the next. Alternative analysis paths leading to the same analysis goal are rather common, so the branching of a workflow is an important property that makes it possible to capture and combine multiple analysis paths in one analysis session model. Likewise, the incorporation of forward jumps as shortcuts allow the same session model to be used for novice and professional users, alike. The guided analysis can switch between a detailed step-by-step walkthrough for the former and a less elaborate, shorter "todo-list" for the latter – even in the middle of the analysis. On top of that, loops make it possible to encode any number of task repetitions by revisiting a task (sequence) until its result is refined enough to be taken as an input for the next task. Moreover, it is possible to define preconditions per task to specify certain requirements to be met, *e.g.*, a hierarchical clustering or aggregation to be performed before visually analyzing the results of the processed data. Likewise, postconditions can be formulated that impose requirements on the result of an analysis task, *e.g.*, with regard to accuracy.

The creation of tasks and their composition to workflows is a demanding activity that needs to be done by domain experts. However, instead of creating the workflows from scratch, it would be desirable to reuse existing workflows from public sources. The online project *myExperiment*[1], for instance, allows users to define and share scientific workflows. Figure 4.7 depicts a sample workflow that a user put in the public domain. It represents the task sequence that is needed to analyze micro-array data of human material. Such workflows capture domain specific tasks but without coupling them to concrete tools. However, in principle, it would be possible to take these existing workflows and associate the individual tasks with the operators, as described in Step V. This would allow for an easy and fast integration of existing workflows into the proposed model-driven design process.

While the definition of the analysis workflows is usually done by a domain expert, it is also possible to leave this to an informed user, who can define paths for the guided, routine users on-the-fly.

[1] `http://www.myExperiment.org`

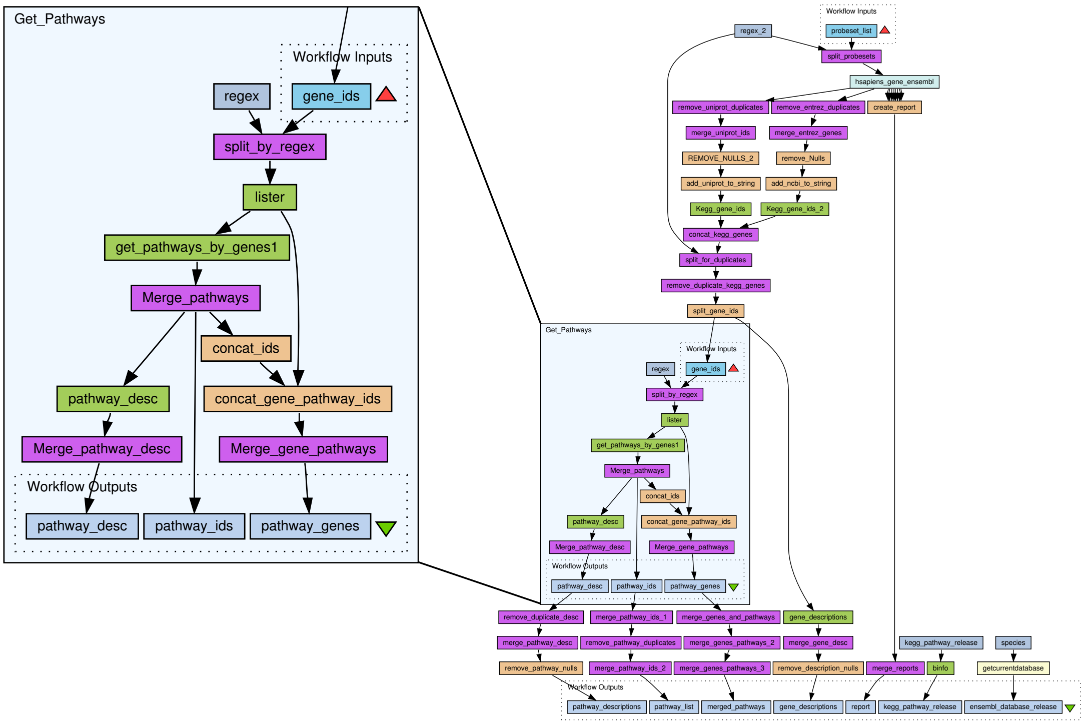


Figure 4.7: Workflow of human micro-array analysis authored by a domain expert and shared on *myExperiment*. Source: `http://www.myexperiment.org/workflows/143.html`.

4.4.2 Step VII: Pruning the Workflows according to the Available Data Sets and Tools

As a final step, the analysis session model is adapted to the constraints imposed by the unavailability of data (*e.g.*, as not all theoretically possible data may have been collected for a given patient or the analyst may not have the clearance to view them) and of the analysis tools (*e.g.*, licensing issues may prevent their use or an analyst may not be properly trained to use them). This adaptation is done by automatically pruning all tasks that rely on unavailable data or interfaces from the workflows. As a result, the remaining workflows cover all currently possible analysis paths which can be chosen as the analysis progresses.

This completes the overall authoring process. It may seem quite elaborate at first, but the modularity of the three models ensures a high level of reusability. The same setup model can be used to build different domain models on top of it, and the same domain model can in turn be used to author numerous workflows utilizing it. This makes sense, as the definition of workflows is usually more short-lived and prone to be changed and optimized more often than the basic setup model or the domain model. The following section briefly explores the final use of the models for providing analyst support, which motivated the externalization of the experts' knowledge about infrastructure, domain and workflows in the first place.

4.5 Utilizing the Models for Analyst Support

The use of the setup model for orientation support is rather straightforward, as the model itself already provides a map in which to pinpoint the current analysis step and determine possible next steps. Using all three levels of the model for the guidance support requires some extra computation.

What needs to be determined first is whether any continuous analysis paths are left after pruning. This makes it possible to check whether or not an analysis goal can be pursued at all by the specific analyst on the given data within the current setup. If not, one could for example request the collection of additional data in order to obtain enough information to be able to complete an analysis path. In our use case, this can be additional tests or screenings for a patient. The session model makes it possible to determine the smallest gap among the analysis paths which can then be bridged at minimal cost – financially or in terms of the stress a patient has to go through. It thus realizes the opposite direction of the pruning: the pruning ensures that nothing is (intended to be) used that is actually unavailable by removing these parts from the model, whereas the reachability check makes sure that everything is available which is needed at the bare minimum to pursue the intended analysis goal.

Secondly, it must be determined which analysis path should be actually used for guiding the analyst through all the possible analysis paths contained in the analysis session model. To reach this aim, it is important to observe that the paths differ in terms of their *seamlessness* and *effectiveness*. A path is considered to be *effective* when it is short compared to other possible analysis paths. A path is called *seamless* if for each transition from one task to the next, there exists a relation (edge) between the data sets that the tasks are connected with as well. A seamless analysis path would allow the analyst to proceed from one task to the next without destroying the mental map, as the data sets used by both tasks are related via a common identifier. The more discontinuities between data sets an analysis path has to bridge, the less seamless it is. For a traceable and swift analysis, paths that are more seamless and effective are generally preferred and thus chosen for suggesting concrete next steps, realizing a guided analysis.

To bring this whole process to life, the following section gives an example for authoring the three models and using both forms of analyst support.

4.6 Applying the Design Process to the Biomedical Use Case

Based on the theoretical foundation laid in the previous section, what follows is a demonstration of how to apply the concept to a real use case. The use case covers a comprehensive analysis of patient-related data. The long-term collaboration partners from the Institute of Pathology at the Medical University of Graz had a need for visual analysis: they try to base their decision of how to treat a newly diagnosed cancer patient on a wider array of available data. In such a scenario, they would like to analyze the patient's basic data, anamnesis, tissue data, gene expression data, *etc.* and relate it to other reference patients. Moreover, they want to be able to explore information about genes, proteins or pathways, which they encounter during an analysis. Hence, it is a prime example of visual analysis across multiple heterogeneous data sets.

4.6.1 Creating the Setup Model

The starting point for creating the setup model (*cf.*, Section 4.2) is a well defined data model, which, in an optimal case, can be based on an existing hospital data management system. In this scenario, many of the data sets are directly linked to the patient. This is reflected in Figure 4.8 by the high degree of connectivity from the patients' basic information to other data sets.

To create the data model, the clinical data manager collects those data sets (green in Figure 4.8) and defines their relations (*cf.*, authoring Step I). Having the data model at hand, the design responsibility is handed over to the visual analysis expert, who chooses or develops suitable visual as well as computational interfaces and assigns those interfaces to the data sets (Step II). This step requires in-depth knowledge about the tools available for conducting the analysis. In this scenario, Caleydo's computational and visual interfaces, as introduced in Chapter 5, are used for analyzing biomolecular-, tissue-, patient- and meta-data. A commercial volume visualization tool is used for MR/CT and X-ray data. The visual analysis expert starts by compiling a list of the available (visual as well as computational) interfaces, as shown in Figure 4.8 at the bottom (visual interfaces are shown in blue and computational interfaces in purple). The available visualization techniques are suitable for depicting data with specific properties. For example, parallel coordinates are capable of visualizing multi-dimensional data. Therefore, this visual interface can be assigned to expression data as well as patient information. Other visual interfaces are the document viewer, heat map, web browser, pathway viewer, *etc.* Caleydo's computational interfaces, including the R statistics toolkit, WEKA and SQL, are assigned to the data sets using the same procedure as before.

The visual analysis expert then compiles a list of operators and assigns interfaces from the compiled list (*cf.*, Step III). In Figure 4.8, the operator pool is presented as a series of red blocks. Operators in our use case are for instance query, similarity analysis of images as well as partitional and hierarchical clustering, where partitional clustering is realized through the R interface, and hierarchical clustering through WEKA. Note that the operators provided in Figure 4.8 are only a sample compilation for the workflow of patient treatment planning. This completes the setup model for the use case.

4.6.2 Creating the Domain Model

In Step IV, the domain expert, in this case a colleague from the Medical University, defines a set of tasks (yellow in Figure 4.8) and assigns the tasks to the data on which they operate. A sequence of operators which enable the fulfillment of the task is associated to each task

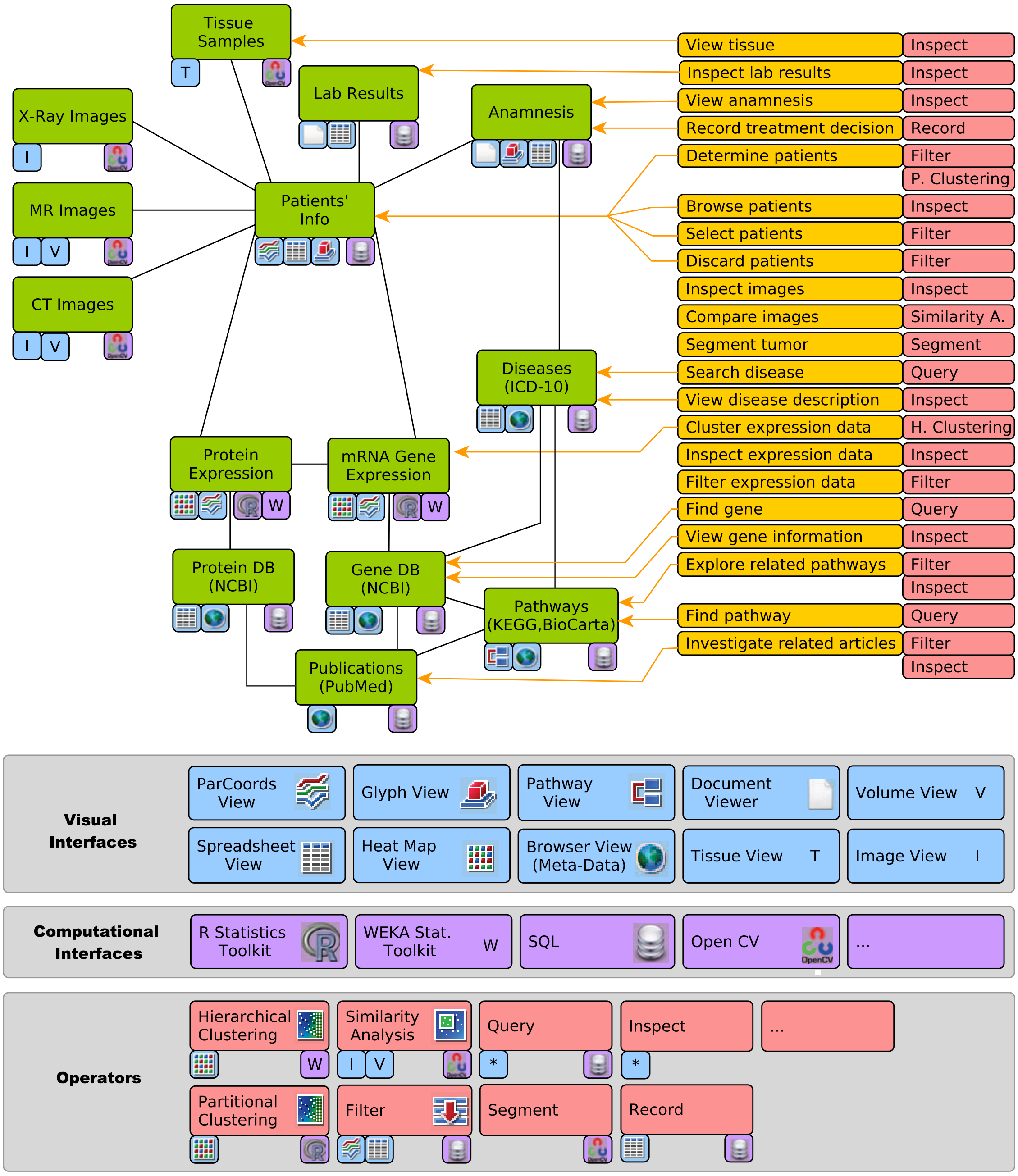


Figure 4.8: Setup and domain model of the biomedical use case. The data sets (green) – either from local or online sources – are connected when they share a common identifier. The interfaces (blue for visual interfaces, purple for computational interfaces) are compiled from several tools and assigned to the data sets. For the analysis session description, tasks (yellow) and operators (red) are added and connected to the data sets.

(Step V). One example for our use case is the “Find gene”-task, which is assigned to the gene database and can be accomplished using the Query operator. Note that this step does not include ordering or connecting the tasks.

4.6.3 Creating the Analysis Session Model

In Step VI, the domain expert defines the workflow as a sequence of tasks, which is the basis for guidance. The following workflow, depicted in Figure 4.9, is an example aimed at the goal described before: determining a treatment plan for a patient diagnosed with cancer. Patients are known to respond differently both to therapy and the disease itself based on several factors, including their genetic traits. Therefore, it is crucial to identify the likely course of the disease for a patient under different treatments.

1. **Determine similar patients**
   First, the guided analyst filters patients based on their anamnesis (for example in terms of age, gender, blood values) using a computational approach.
2. **Browse patients**
   The analyst explores the patients that remain in the sample and tries to find differences in their conditions.
3. **View tissue**
   For the remaining patients, he explores the tissue images, on which the initial diagnosis was based. This is done to make sure that the patients actually present similar manifestations.
4. **Discard patients**
   Remove patients with different manifestations in terms of the tissue samples.
5. **Cluster expression data**
   To be able to identify patients with similar gene expression patterns, which might indicate common traits and therefore a similar course of the disease, the data is clustered.
6. **Inspect expression data**
   The analyst inspects the clustering results to find patterns where the patient under investigation is similar to one group of patients, while different to others. He then selects a group of genes that clearly distinguishes the patient group from others. If the genes’ functions are clear to the analyst (*e.g.*, a well-known proto-onco or tumor suppressor gene) he can directly jump to Task 9. If this is not the case, he can proceed with the next task to find out more about their function.
7. **Explore related pathways**
   To understand the found genes’ function, the analyst explores the pathways containing the genes.

8. **View gene information**
   Further information about a particular gene is gathered by inspecting its entry in an online database.

9. **Select patients**
   With the knowledge that the genes are in fact relevant for the condition, the analyst goes back to the gene expression view, where he selects those patients that are in the same group as the patient under investigation.

10. **View Anamnesis**
    The analyst then views the anamnesis to judge whether previous courses of actions were successful for similar cases and bases his treatment decision on the findings.

11. **Record Treatment Decision**
    He records the treatment decision in the patient's anamnesis.

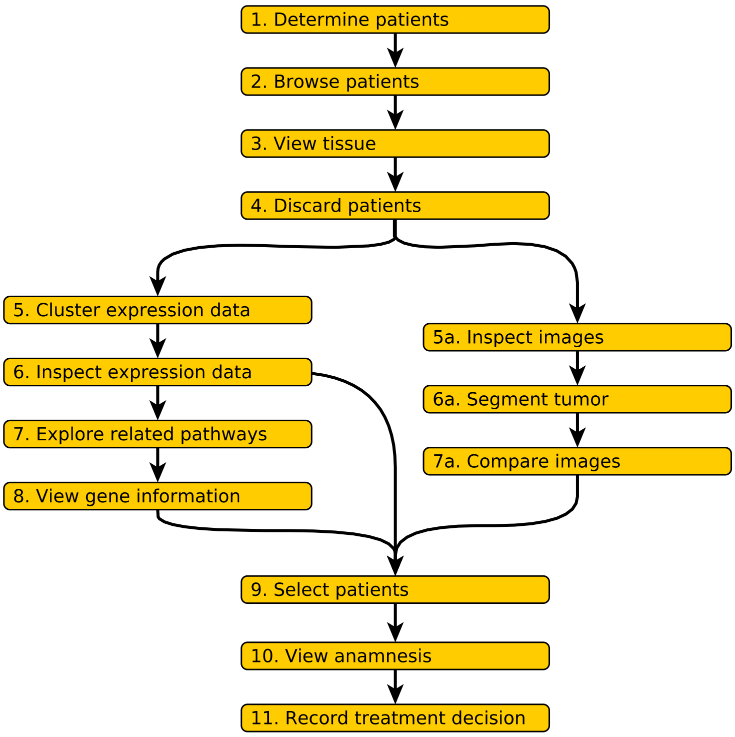


Figure 4.9: The workflow of finding a treatment plan for a newly diagnosed cancer patient.

Alternatively, instead of conducting an analysis based on gene expression data (Tasks 5 to 8 in the left branch of Figure 4.9), the guided analyst can choose to conduct the selection of patients in Task 9 based on an exploration, segmentation and comparison of tumor images (*cf.*, Tasks 5a to 7a). However, the right branch is only feasible if the disease under investigation causes tumors, visible in imaging data.

Preconditions are defined optionally for each task. For instance, before viewing the tissue slices in Task 3, the analyst needs to filter less than 20 patients.

Before the models can be utilized by an analysis system they need to be tailored to the given constraints (*cf.*, Step VII). In our scenario, the patients' protein expression profiles are not available which makes the protein database obsolete. Furthermore, due to access restrictions at the hospital, lab results cannot be a part of the setup. Based on the remaining available setup resources, the automatic pruning of paths is performed. As the exemplary defined workflow samples are rather small, all tasks of the workflow are possible and consequently remain in the analysis session model.

Before a real system can employ the described model, ways to create such a model (*i.e.*, authoring interfaces) need to be discussed first.

4.7 Authoring Realization

To be of use in actual systems, the models described must be available in machine-readable form: either by explicitly creating the model offline, or by capturing interface actions and associating them with tasks at runtime. The interactive method is only suitable for the domain and the analysis session model, since it requires the setup model to be performed.

Tools for offline creation of the model range from dedicated authoring solutions[2] to simple XML editors. While these external tools can be used out-of-the-box, an integrated solution is potentially more powerful: on-the-fly editing and refinement can be tightly bound to the visual data analysis. It enables users to create and refine models – making a live role switch possible – *i.e.*, the analyst becomes the author.

The choice between these two variants is a trade-off between flexibility and costs. This tight integration of data analysis and authoring requires high initial costs in terms of software engineering. As authoring interfaces are not the focus of this paper, the models covering the biomedical use case have been directly created in XML.

In addition to the explicit way of creating the model via an authoring interface, this information can also be hard-wired in an analysis system where visualization experts design and implement a system that contains the knowledge implicitly. This is often the case for special purpose software that is particularly tailored to a certain use case and only addresses a small class of domain-specific problems. Especially knowledge covered by the analysis setup model is a precondition of every functioning visual analysis system. Creating the system requires the same effort from the visualization designer, but an explicit model enables them to employ this knowledge for orientation and guidance support.

[2] *e.g.*, Altova Authentic®, `http://www.altova.com/authentic.html`

4.8 Summary

This chapter introduced a model-driven design approach, one of the major contributions of this thesis. The model not only captures multiple interconnected data sets and the interfaces to operate on the data, but it also allows an author to define workflows on top of this information. The model lays the theoretical foundation of this thesis that allows an analysis system to realize both levels of support, orientation as well as guidance.

Chapter 6 and 7 will introduce concrete visualization techniques that provide orientation support in the information landscape (*S1.1*) based on the first stage of the model, the analysis setup. While Chapter 6 realizes the orientation support in one or multiple data sets visualized in multiple views, Chapter 7 discusses the multi-application as well as the multi-user aspect, both of which impose new requirements in terms of orientation. Finally, in Chapter 8 the Stack'n'flip analysis system will be presented as one possible way of realizing comprehensive analyst support that covers all levels (*S1.1, S1.2, S1.3* as well as *S2*). Stack'n'flip employs the full three-stage model and gives an impression what the model is able to achieve in terms of analysis support.

Before this thesis continues with the concrete ways to utilize the model by means of interactive visualization, however, the next chapter introduces Caleydo, the visualization framework that was created over the last several years and served as major infrastructure for realizing the visual analysis techniques presented in the remainder of this thesis.

Chapter 5

Caleydo - Visual Analysis Framework

Contents

Caleydo is a visual analysis framework designed and implemented in close collaboration with Alexander Lex in the course of this thesis project. The framework contains the concrete set of visual and computational interfaces that are part of the analysis setup model from the complex biomedical use case introduced in the previous chapter. As this framework is the basic infrastructure of all research prototypes presented in this dissertation, a separate chapter is dedicated to it.

In 2005 the framework's development was initiated by Michael Kalkusch who left the project team in 2007. Until 2009 the focus was on the interactive visualization and exploration of expression data in the context of cellular processes. After that, the integration of computational data processing capabilities and the comprehension of a wider spectrum of input data, as described in the Chapter 2, broadened its range of application. The Caleydo software is available for download free of charge from `caleydo.org`. The framework is implemented in Java and runs on Windows and Linux machines.

The Caleydo system has not only been published in the InfoVis community [Streit et al., 2008, Lex et al., 2010a] but also as an application note in the Bioinformatics journal [Streit et al., 2009a]. This dissemination activity in the Bioinformatics community helped in gaining access to new users from the focus group. The system is currently being used as a research tool by various biomedical research institutions. First results acquired with the support of the software have already been published (see [Schmidt-Gann et al., 2009]).

Knowledge Gap

Van Wijk identified user-centered design as the royal road for a successful interdisciplinary visualization research project [van Wijk, 2006] . Knowing the users and their tasks has been regarded as essential for developing high-quality user interfaces for more than a decade now [Hackos and Redish, 1998]. Conducting a requirements and task analysis leads to better quality software and therefore lowers costs, for both the developers and the users. This view of software development has long been adopted by the fields of visualization and visual analytics (*e.g.*, [Kang et al., 2009, Saraiya et al., 2005]). In the case of Caleydo, the framework's development was, right from the beginning, driven by input from domain experts. However, interdisciplinary research projects require both sides, researchers from the problem domain as well as visualization experts, to learn a lot about the other domain. This discrepancy between expertise, vocabulary and background between the ones who provide visualization solutions and those using them is referred to as a *knowledge gap* [van Wijk, 2006]. On the one hand, visualization experts have to learn the target domain's vocabulary in order to understand the problems and in turn to be able to come up with solutions. On the other hand, domain experts need to invest time and have to be open-minded for innovative and novel applications. Although this process consumes a considerable amount of time, it can be very fruitful and lead to innovative solutions.

In our community the requirement elicitation process is usually repeated for every project. This has led [Scholtz, 2009] suggest to create a handbook of user profiles to alleviate the difficulties in accessing real end users and to reduce the time that needs to be invested by both parties. As each project is driven by a very specialized set of requirements and user demands, such profile collections cannot replace the whole procedure. However, a good portion could be shared among the projects. Based on experience gained in the Caleydo project, the author also advocates such an initiative.

Interest Gap

In contrast to the knowledge gap, an *interest gap* between the domain expert and the visualization researcher can be observed [van Wijk, 2006]. While the visualization expert wants to realize innovative visualizations and visual analysis techniques, the domain expert (user) focuses on solving open domain problems. In order to keep the knowledge gap as narrow as possible, actually usable software – not a pure proof of concept research prototype

– is important. Thus, scientifically irrelevant features need to be integrated. Examples are a flexible data importer or a standard scatterplot implementation. However, the effort made in the Caleydo project to provide a stable and ready-to-use software for end-users laid the groundwork for further challenging research questions. The framework served as a platform that makes a rapid realization of novel visualization research ideas possible. In particular, the related question of whether or not software engineering pays off for research in the long run is addressed in [Streit et al., 2010].

5.1 Framework Design

Modern software engineering is heavily affected by best practices for solving recurring abstract problems, so-called design patterns. Following these patterns helps to reduce the costs of creating re-usable, high-quality software. While collections of design patterns, applicable to general software architectural problems have been used for more than twenty years now (*e.g.*, [Gamma et al., 1995]), specific patterns tailored to the needs of information visualization software were not formulated until recently. Based on existing visualization frameworks and the experiences gained with *prefuse* [Heer et al., 2005], Heer and Agrawala identified valuable design patterns valid for information visualization [Heer and Agrawala, 2006]. They focus on the interplay of data representation, graphics and interaction. The design of Caleydo was naturally influenced by some of these patterns which will be indicated and named at appropriate points in the remainder of this chapter.

The Caleydo framework is conceptually divided into four main building blocks:

- The **Core System** contains basic functionality such as data management, data parsing, ID management and the event system for the propagation of information and updates within the framework.
- **Data Domain Plug-ins** reflect a certain kind of domain-specific data – for example genetic data, pathways or tissue data. Each data domain plug-in determines the compatible data types, a central ID (*i.e.*, primary key) of the data set, how to parse raw data of this type, *etc.*
- **View Plug-ins** contain the implementation of a concrete visualization technique. Each view is associated with the data domains it is capable of visualizing.
- **External Libraries and Tools** add functionality to the framework, *e.g.*, for statistics computations. They are accessed by Caleydo via an API.

Figure 5.1 illustrates how Caleydo is structured in terms of these modules. Each module is subdivided into separate plug-ins. The strict separation between view and data follows the Reference Model design pattern [Heer and Agrawala, 2006]. It enables a free composition of views that render data from multiple sources.

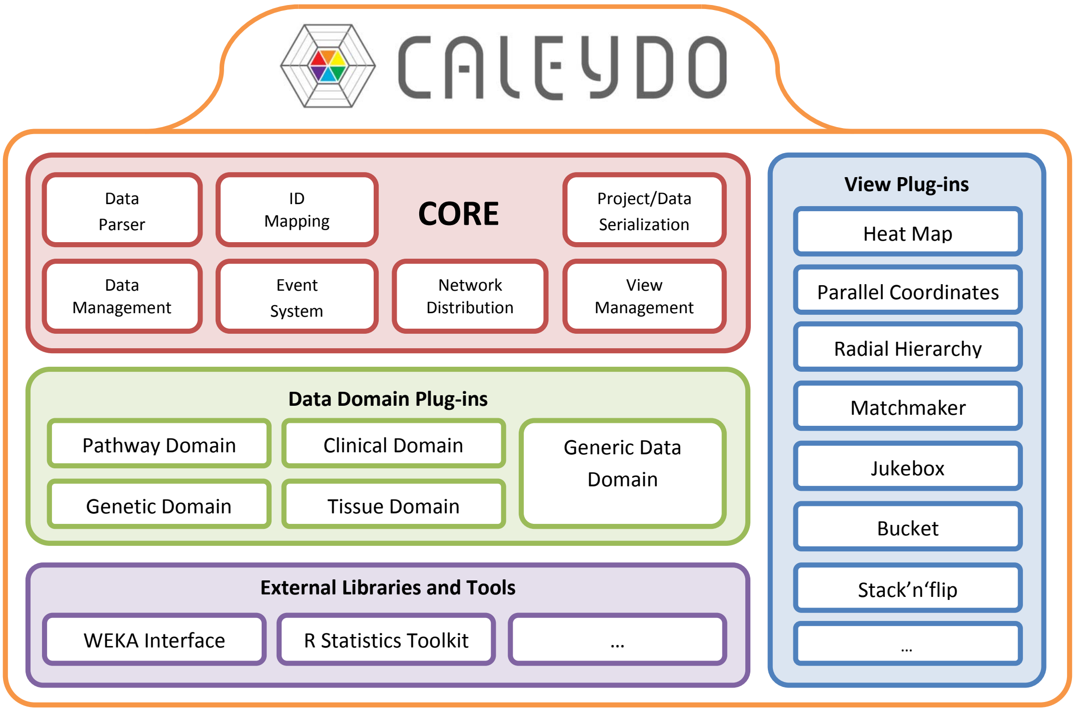


Figure 5.1: Main building blocks of the Caleydo framework.

View and data domain plug-ins have full access to the core's base functionality. In contrast, the core itself has no knowledge about or references to other plug-ins. This modular software design makes it possible to extend the Caleydo framework without altering the core system. In addition, by maintaining a stable core, new visualization prototypes, student projects, and also work with external collaborations can be developed in a sandbox environment without influencing other modules of the framework.

5.1.1 Data Management

One precondition of a framework targeted at heterogeneous and/or comparative data analysis is its ability to concurrently operate on multiple data sets. In Caleydo, data domain plug-ins realize the conceptual separation of different kinds of data. They hold rules how to load, store and process the contained data. Instances of these data domain plug-ins store the data itself. The framework provides the infrastructure for graph, image as well as multi-dimensional tabular data. In order to guarantee immediate access to the data – a requirement for interactive analysis – everything is kept in the main memory during the the application's runtime.

Session Store/Restore

In order to make software ready for use in real world data analysis, one of the most essential features is its ability to store (and restore) a certain state. By employing the Memento paradigm [Gamma et al., 1995] Caleydo can save the currently loaded data set, the arrangement of views as well as additional information, like the chosen color coding, current selections, filters and associated processing results, *e.g.*, cluster information. This session-specific data is stored in a project file. This way, analysts can not only quickly recover when temporarily interrupted, but also manage and share their findings.

Handling of Multi-Dimensional, Numerical Data

When dealing with huge, multi-dimensional data sets, a visual analysis system's data management has to consider any special needs in terms of availability and flexibility. The way n-dimensional data is stored and accessed is different from the conservative way used in relational database systems. While databases use a row major order in which all data fields are stored per record, information visualization systems often store data on a per column basis (*cf.*, *Data Column* design pattern in [Heer and Agrawala, 2006]). This solution makes the grouping of records (in row direction) more difficult, but has the advantage of a unique data type per column, enabling a compact storage in arrays.

Storage arrays in Caleydo are not accessed directly but via *virtual arrays* – a list of indices pointing to the actual data entry. This way operations like brushing, grouping and sorting can be performed efficiently without changing the original data storage.

5.1.2 Annotations and ID-Mapping

When working with multiple heterogeneous data sets, shared identifiers within and among data sets are the fundament on which sense-making is based. In this case, the relations between multiple data sets are implicitly contained, for example, if a row in a tabular data set is identified by the same ID as a node in a graph. In addition to these implicit data mappings, external mapping tables have to be utilized to resolve indirect mappings between data sets. Especially in the biological domain, analysts are confronted with a broad set of different annotation systems. Established annotations for the identification of genes are for instance *RefSeq*, *Entrez Gene ID*, *Genebank Accession* and many more. However, because of duplicates, semantic variations or even false identifiers in out-dated annotation systems, a 1:1 mapping between these systems is impossible. Life science experts are usually aware of these issues, but want the analysis software to handle it. Caleydo relies on the *DAVID* annotation database[1] [Huang et al., 2008], a unique point of mapping for all kinds of genetic entities. This allows Caleydo to offer a flexible data importer (*e.g.*, for loading gene expression data) which understands any of the existing standard annotations that

[1] http://david.abcc.ncifcrf.gov

DAVID covers. The same DAVID-based mapping mechanism is beneficial for Caleydo's search feature, where users can find entities by using any ID as a search query.

At start-up, Caleydo builds up a mapping graph by loading a series of mapping tables. In order to create a flexible network of relations, the graph structure supports uni- as well as bi-directional mappings of identifiers. Furthermore, the implementation copes with 1:n as well as n:m mappings, which are, for instance, relevant for handling multiple spotted genes on a micro-array chip. When resolving identifiers, *i.e.*, transforming from one annotation system to another, the shortest path in the mapping graph is determined by applying the Dijkstra algorithm [Dijkstra, 1959].

5.1.3 Event System

The propagation of events is a key mechanism of any visual analysis systems. It is the fundamental building block of multiple coordinated view systems [Roberts, 2007] which supports linking & brushing [Becker, 1987, Martin and Ward, 1995, Ward, 1994], but also the basis for any kind of communication between a framework's modules. In Caleydo, the event mechanism is a variation of the Observer design pattern (*cf.*, [Gamma et al., 1995]) where objects declare interest in certain types of events and are notified whenever the event occurs.

The synchronization of data brushes among views is realized by exchanging *[unique_ID,state]* tuples. Examples of data item states are mouse-over, selected, deselected and removed. In a decentralized fashion, views maintain the state of all their rendered data items. The framework supports either sending the full selection state of a data set, which is naturally an expensive operation, or alternatively sending only the altered portions. These small incremental updates are of course favored for brushing operations in terms of performance and scalability. However, in case a view is newly opened, the framework needs to initialize its state with a full one-time update, so that data which is already filtered in existing views does not re-appear in the new view.

Brushing of data can either be performed by computational means or visually by the user. Irrespective of how the brushing is triggered, all data operations are collected by the data domain instance that is associated with the data set. The collected operations are fed into a pipeline, also called "Brushing sequence" [Chen, 2003]. By default, brushing operations are connected as a series combined by logical AND-operations. Logically combining data filters (AND, OR, XOR, *etc.*) is a common concept in visual analysis frameworks [Martin and Ward, 1995, Doleisch et al., 2003]. Upon every newly added or altered brush, the pipeline is re-evaluated and the result is published to the views and other interested modules. In addition, storing the filter pipeline enables an arbitrary removal of parts – realizing an UNDO on the level of data operations. By explicitly presenting the filter pipeline to the user in a designated meta-visualization, its full flexibility can be accessed: interactive resorting, refining, logical combining and removing of filter steps. In Caleydo, the filter pipeline is part of the data meta view, described in Section 5.3.4.

5.2 Computational Interfaces

Computational interfaces refer to all algorithmic means a framework provides to calculate derived data, such as cluster results or a filtered subset resulting from an applied statistical test. It was a design decision to rely on standard libraries for these kind of tasks, as they offer a pool of validated, state-of-the-art statistical methods. For that purpose, Caleydo integrates the R statistics toolkit[2] [R Development Core Team, 2010] as well as the data mining workbench WEKA[3] [Hall et al., 2009].

The standard workflow of using such libraries is to access them as a black box: input data is pushed in, processed and the result is then presented to the user. For this final step, most computational tools support rather static plots that are very limited in terms of interaction. Given the common requirement that a user needs to alter a query continuously during an analysis, for instance changing a parameter of a clustering algorithm, the only way to do this is to adapt the calculation rules by hand and re-trigger the processing pipeline. This inherent black box paradigm cannot be circumvented when using the standard statistics packages (as Caleydo does), but the integration in a Visual Analytics software can improve usability significantly. By providing easy access via a graphical user interface, the user does not have to deal with the libraries' complexity. This enrichment of the computational capabilities with the power of interaction makes the integration interesting for Visual Analytics applications. Other frameworks that employ external libraries for this purpose are Mayday [Dietzsch et al., 2006], SEURAT [Gribov et al., 2010] and SpRay [Dietzsch et al., 2009]).

When input consists of big, multi-dimensional data sets, a common first analysis step is to reduce the data by applying dimension reduction procedures (*e.g.*, PCA) or other basic statistical tests like t-testing, variance-based filtering within groups or fold-change reduction. For the analysis of patterns and trends in the data, Caleydo offers several clustering algorithms, such as hierarchical clustering [Eisen et al., 1998], k-means and affinity propagation [Frey and Dueck, 2007]. The latter in particular has proven valuable for the analysis of expression data. However, the set of supported clustering algorithms and statistical methods can be easily extended to Caleydo by triggering the appropriate commands in the external libraries.

5.3 Visual Interfaces

A visual interface is a concrete implementation of a visualization technique – also referred as a view. Due to the modular plug-in mechanism, the framework facilitates rapid prototyping for quick integration of novel visualization techniques.

Caleydo is implemented as a multiple coordinated view system. For each view, the framework provides base functionality like the coupling with the event mechanism for

[2] `http://www.r-project.org`

[3] `http://www.cs.waikato.ac.nz/ml/weka`

syncing filter and selection operations.

The framework employs the Rich Client Platform (RCP) [McAffer and Lemieux, 2005], the back-end of the popular Eclipse software development environment. RCP provides the base functionality that allows users to freely arrange views in tabs and/or in a side-by-side fashion by using drag-and-drop. Due to the fact that in complex analysis sessions users invest considerable time in customizing their view arrangement, Caleydo provides a memento-based store and restore mechanism for remembering the view layout. Views can either be realized with the Standard Widget Toolkit (SWT) or as embedded OpenGL content. While SWT is employed for standard GUI views like tables or lists, all implementations of novel, interactive 2D and 3D visualization techniques are rendered in OpenGL using the Java OpenGL Toolkit (JOGL)[4].

In the following section, Caleydo's default set of visual interfaces are introduced. Although they are based on well-established techniques, a series of novel features deserve special mentioning. This is particularly true for the hierarchical heat map, the parallel coordinates and the pathway graph view. Auxiliary views are also listed and briefly described. Further variations of established visualization techniques that are part of Caleydo, like for instance a scatterplot matrix (SPLOM) [Carr et al., 1986, Becker, 1987] or a Sunburst view [Stasko and Zhang, 2000], need no further discussion and have therefore been omitted for the sake of brevity. Novel visualization techniques developed in the course of this thesis are presented in greater detail in Chapter 6 and 8.

The choice of views that were implemented in Caleydo was naturally driven by the biomedical use cases and the requirements connected to them. However, the general applicability of the techniques to multi-dimensional as well as graph data allows Caleydo to serve as a general framework for visual data analysis.

5.3.1 Hierarchical Heat Map

A heat map is a visualization technique for representing multi-dimensional data as a color-shaded matrix and the quasi standard for visualizing expression data. Each row of the data matrix corresponds to a data record, each column to one dimension and each cell is mapped to a color. An unsorted heat map can be used as a lookup table for small data matrices ($<$ 50x50). However, by permuting the rows and columns, structure and coherent patterns in the data become apparent (*cf.*, Bertin's reorderable matrix [Bertin, 1983]). A color-shaded matrix, where similar elements are positioned close to each other, is called clustered heat map [Wilkinson, 2009].

Clustering a group of patients with known features is a common procedure to analyze expression data. The goal is to find similarities and/or differences between their profiles which in turn allows one to draw conclusions in the analysis. In addition to a comparison between different patients, time-series experiments of the same patient are frequently available. Consequently, a combined correlation analysis on the basis of different patients

[4] `http://jogamp.org/jogl`

measured at multiple points in time is often subject of an analysis.

While heat maps are used for a wide variety of application fields, they are very common for visualizing gene expression data [Eisen et al., 1998]. Clustered heat maps have appeared in over 4000 life science-related articles and are therefore probably the most widely used visualization technique in this field [Weinstein, 2008]. Clustered heat maps are a particularly valuable visual analysis tool in this context, due to the tendency of genes involved in the same functional process to be co-regulated. Consequently, genes which are assigned to the same cluster potentially perform similar cellular functions [Eisen et al., 1998].

Traditionally, bioinformaticians apply statistical methods to the expression data in order to search for trends as well as differences or similarities between subsets in the data. In such a scenario, visualization is used for pure presentation purposes in the form of static plots generated in a post-processing step. However, an interactive heat map combined with features such as linking & brushing is much more valuable for analyzing the data. An early example of a framework for dynamic querying of expression data is the Hierarchical Cluster Explorer (HCE) [Seo and Shneiderman, 2002]. Nowadays, clustered heat maps can be considered as state-of-the-art as they are an integral part of many general purpose visualization frameworks (*e.g.*, TIBCO Spotfire [5]), as well as special purpose software for the biomedical domain (*e.g.*, GeneSpring (Agilent Technologies, Inc., USA)[6], Java TreeView [Saldanha, 2004] and Mayday [Dietzsch et al., 2006]). A comprehensive review of the current tools available can be found in [Gehlenborg et al., 2010].

A typical expression data set is degenerated in size with respect to the ratio of dimensions (*i.e.*, columns) and items per dimension (rows) – *e.g.*, several dozen experiments with 30,000+ regulation values each. In general, naive heat map implementations do not scale well, as the number of simultaneously visualized elements has a strict upper limit: the number of available pixels on the screen. Established tools like Mayday [Dietzsch et al., 2006] handle that problem with a scroll and pan interface, which can only visualize a small subset of the full data set at a time. However, the main purpose of clustered heat maps is to show the complete data set in order to reveal patterns within the data in an intuitive way. Consequently, simultaneous investigation of overall trends and patterns is as important as easy access to individual elements. Caleydo addresses this issue by realizing the heat map in a multi-level fashion that employs a focus+context concept reminiscent of *e.g.*, [Ball and Eick, 1996, Seo and Shneiderman, 2002, Saldanha, 2004]. Figure 5.2 (c) shows a hierarchically clustered heat map of about 800 elements as the center view of the Caleydo workbench. The overview level on the left presents the full heat map rendered to a texture. Due to interpolation performed by the graphics hardware, the scaled texture retains the major trends in the data. In a sliding window fashion the user can interactively determine the subset that is enlarged in the subsequent level. Up to three levels of detail can be shown simultaneously. The number of required levels is automatically determined according to the number of items. In the last level on the right, the subset of elements is small enough

[5] `http://spotfire.tibco.com`

[6] `http://www.genespring.com`

to present the individual gene names. The initial version of the hierarchical heat map was implemented by Bernhard Schlegl in the course of his master's thesis [Schlegl, 2009].

On-site Dendrogram

If the data set is clustered hierarchically, additional dendrograms are shown on all levels. Cluster borders are visualized with overlaid lines. The desired level of granularity of groupings (*i.e.*, clusters) can be adjusted interactively by dragging a cut-off line of the dendrogram (similar to the minimum similarity bar in HCE [Seo and Shneiderman, 2002]).

Color Coding

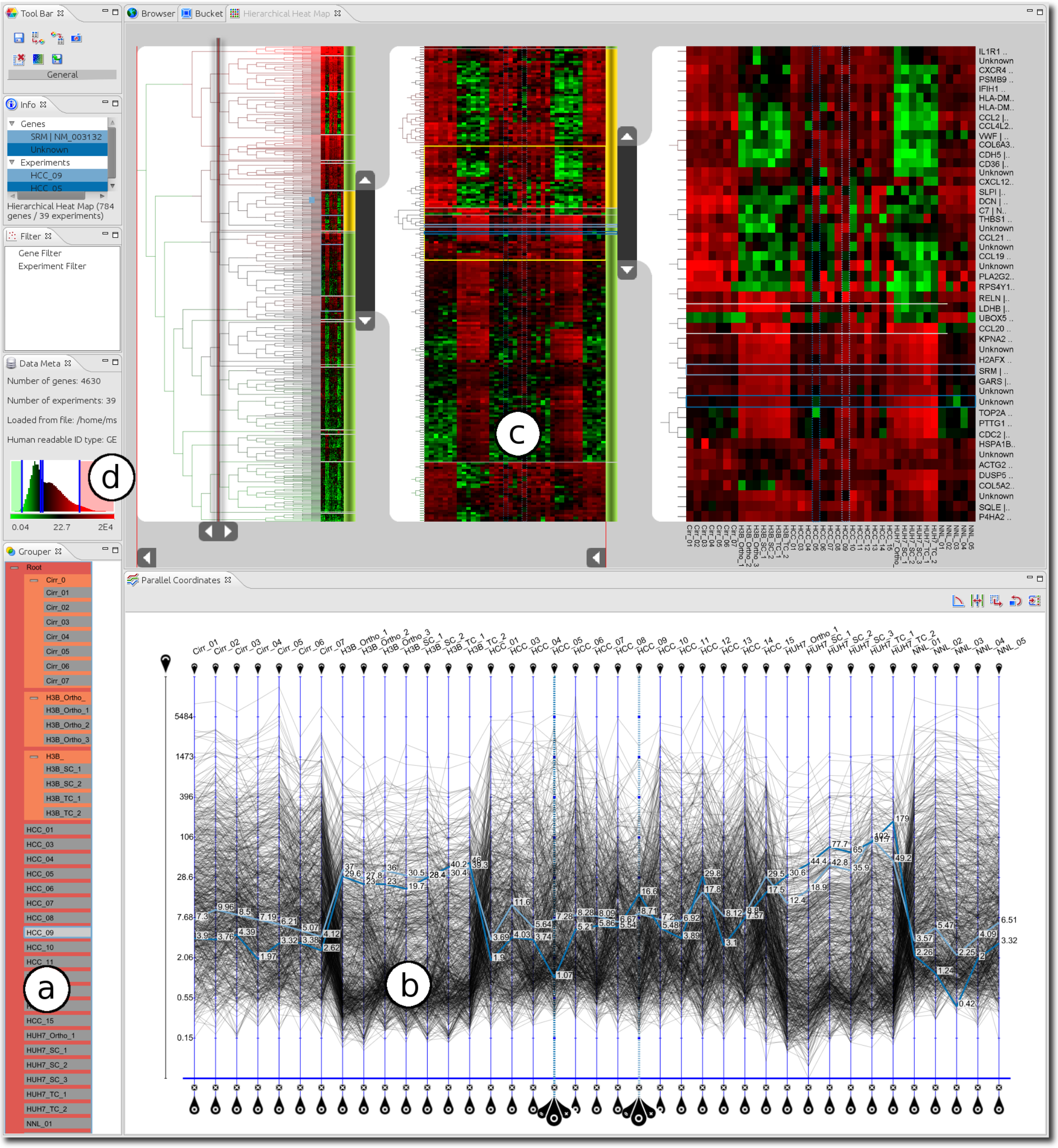
Choosing a suitable transfer function that determines the encoding of numeric values to colors is an important aspect when working with heat maps. One widely-used color map for visualizing gene expression values ranges from red (indicating up-regulated genes) to black (meaning a similar regulation to a reference experiment) to green (down-regulated genes). It is a quasi domain-specific convention and therefore also the default in Caleydo. However, in order to satisfy the needs of color-blind users and also to enable flexible adaptation to various use cases, the system offers a set of predefined color palettes to choose from as well as the possibility to load arbitrary, user-defined color schema. Via an on-site histogram view, users can interactively change the transfer function, the effects of which are immediately reflected in the heat map. Due to the sensitivity of heat maps to the transfer function, this live adaptation of the color mapping is a valuable feature when it comes to the discovery of patterns and trends.

While clustered heat maps work well for visually inspecting the data and getting a feeling for overall trends, they are not suited for performing visual filter operations. For that purpose Caleydo offers a parallel coordinates view.

5.3.2 Parallel Coordinates

The state-of-the-art parallel coordinates implementation in Caleydo (*cf.*, (b) in Figure 5.2) allows users to freely arrange axes in a drag-and-drop fashion. Various brushing strategies, such as brushing gates per axis (dimension) or angular brushing [Hauser et al., 2002], are realized.

In order to be able to deal with large data sets that would naturally result in visual clutter, the implementation renders a reduced random sample [Ellis and Dix, 2006], which adapts automatically once data items are filtered out – always displaying a predefined number of polylines. When the number of remaining items is smaller than this number, sampling is turned off. An initial version of the Caleydo's parallel coordinates view was implemented by Alexander Lex in the course of his master's thesis [Lex, 2008].



Hierarchical Heat Map (784 genes / 39 experiments)

Number of genes: 4630

Number of experiments: 39

Loaded from file: /home/ms

Human readable ID type: GE

Figure 5.2: Caleydo workbench sample analysis setup. The loaded data set is published in [Panzitt et al., 2007] and comprises 4630 gene expression values for 39 patients. The grouper view (a) allows a user to hierarchically structure the patients according to their semantic commonalities. By using visual brushes in the parallel coordinates view (b) the analyst has filtered the data to 784 genes. This reduced set of genes was clustered hierarchically. The hierarchical heat map (c) depicts the hierarchy as an in-place dendrogram. An auxiliary view (d) contains meta-data about the loaded data set and an interactive histogram for adapting the color transfer function.

5.3.3 Composition of Visual Interfaces

Separate windows as containers for views are the usual way to present data in information visualization systems. This also applies to Caleydo which supports classic GUI views, such as a spreadsheet view for presenting raw tabular data or a web browser view, or alternatively, views that use OpenGL for rendering information, like parallel coordinates or heat maps.

In contrast to these standalone views for the visualization of interconnected, heterogeneous data sets, a composition of various views in one window is a powerful concept. Therefore, Caleydo provides a mechanism for rendering views remotely at an arbitrary position in a 2D or 3D scene. Each view is thus a fully functional standalone view. The layout of the combined views is based on “style-sheets”. Consequently, it is possible for a view to accommodate various arrangement strategies.

Having combinations of views embedded in one window and therefore in one OpenGL scene is the prerequisite for depicting relations across views. This remote view mechanism is the fundament in terms of infrastructure for the Matchmaker (*cf.*, Section 6.1.2), the Jukebox (Section 6.2.2), the Bucket (Section 6.3.3), and the Stack’n’flip technique (Chapter 8).

5.3.4 Auxiliary Views

This section briefly introduces support views that are not necessarily interesting from a scientific point of view, but are relevant in order to understand the analysis scenarios throughout the thesis.

Data Meta View

As Caleydo supports concurrent loading of multiple data sets, one instance of a data meta view is available per data set (*cf.*, (d) in Figure 5.2). Depending on the data set, the view presents basic information like size, source location, data type, primary identifier and further properties. An embedded histogram not only shows the distribution of the data set, but also facilitates an interactive alteration of the color transfer function.

Grouper

In a multi-dimensional data set the dimensions can either be grouped by computational means or manually by the user, according to known semantics. An example of the algorithmic approach is the application of a clustering algorithm that assigns dimensions to clusters according to a defined distance measure. However, in many cases a grouping of dimensions is desired by means of meta-knowledge about the data set. A flat or hierarchical grouping structure can reflect the semantic relation between the dimensions. On the left side of the Caleydo workbench (see (a) in Figure 5.2) the grouper view shows a pre-grouped, multi-dimensional expression data set. In this example, replicate experiments have been run in order to reduce the uncertainty of micro-array measurements. These replicate experiments are then aggregated, constituting the lowest level of the hierarchical grouping. Subsequently, these replicate groups are again assigned to groups of experiments that contain multiple patients with the same clinical condition. By taking into account the (not explicit) knowledge about the data set, this kind of hierarchical structuring can be arbitrarily defined by the user.

Group labels are automatically determined from common string parts of the contained dimension labels that can be changed by the user.

In addition to structuring purposes, the grouper supports removal, sorting (by drag-and-drop), collapsing and duplication of dimensions, where the latter is especially relevant for comparative data analysis. Statistical operations on single dimensions as well as on groups of dimensions can be triggered via the context menu, the result of which is immediately reflected by a reduction of the data items.

Tabular Data View

When analyzing tabular data in sophisticated, multi-dimensional visualizations, like parallel coordinates, scatter plots or heat maps, a basic user requirement remains to access and browse the raw data in a table, since this is usually most familiar to the users.

Selection View

A permanently visible mini-view in the side-bar of Caleydo shows a list of the currently selected entities.

Bookmark View

During analysis sessions domain experts often want to remember entities either to return to them at a later point in the analysis or to export them at the end of the analysis. Bookmarking of entities is particularly relevant for exploratory analysis where hypothesis verification and falsification play an important role.

External Information Browser

A significant amount of time spent exploring and analyzing complex, interconnected data is consumed by the investigation of meta-data. In order to understand complex entities, relations and processes, the consideration of additional knowledge is essential. Especially for hypothesis generation tasks, external databases are of great importance. While the sheer amount and complexity of the data are challenging aspects, the uncertainty of the loaded data and the contained ID-mapping is problematic as well. In this regard, the information from external databases can help users to do quick plausibility checks.

Using a details-on-demand technique, selected data items in Caleydo are dynamically queried in major online databases, which then return detailed information about genes, enzymes, protein structures and other entities. One already mentioned example of such important external databases is *PubMed*. Given the fact that the most comprehensive body of knowledge about a biological entity or process is the pool of scientific publications on that topic, dynamic queries on *PubMed* are a powerful way of achieving profound sense-making.

In addition to the well-established linked meta-browser as a passive data output facility, Caleydo actively parses selected text strings of the loaded web page and looks for matching entities in the internal mapping tables. Positive matches are highlighted system-wide in all open views. Currently unloaded but known entities, like pathways, are opened in the designated views.

5.4 Summary

This chapter introduced the Caleydo Visualization Framework. It is a multiple coordinated view system focused on the analysis of biomedical data. The framework serves as a basic infrastructure for realizing the novel visualization techniques presented throughout this thesis.

Although Caleydo in its current state scales well to data sets up to a magnitude of tens of thousands of (interrelated) items, handling bigger data sets will require adaptations in the framework. While the currently used application workflow requires the data to be analyzed to be loaded to the main memory, a combination with database systems, serving as data warehouse, seems to be inevitable. In this scenario, the visual analysis system would dynamically query these databases, resulting in chunks of data which are then transferred to the main memory for immediate access.

The limitation in terms of data size not only necessitates new designs in visualization techniques, but also requires novel ways to ensure their interactive handling. In this regard, multi-threaded render passes per view combined with early thread termination [Piringer et al., 2009] could be one building block of a visual analysis system that scales up to the next generation of analysis problems.

Chapter 6

Orientation Support in Classic Multi-View Applications

Contents

By utilizing the information from a well-defined analysis setup model, a visual analysis system can aid analysts and therefore alleviate the risk of disorientation that would result in an undirected and longer lasting analysis. This chapter introduces a series of novel visualization techniques targeted at the informed user which focus on orientation support in the data space (support level *S1.1*), each of them addressing a concrete, open domain problem. Before discussing the individual techniques in detail, the respective domain problem is introduced, and a domain-independent, general research question is formulated.

The visualization techniques presented are designed to facilitate an exploratory analysis of either a single multi-dimensional data set (gene expression), or the combined analysis of two heterogeneous data sets (expression in the context of pathways). The low number of data sets involved and the connected visual/computational interfaces make the setup easily comprehensible and manageable. Thus, the analysis setup models that cover these scenarios are only a very reduced subset of the full version introduced in Chapter 4. An explicit meta-visualization, showing the dependencies between the data sets and the assigned interfaces, useful for keeping a user oriented in the information landscape is not necessary here. Nevertheless, these small setups are challenging in terms of orientation, as the analyst needs to be informed about the relations within the information landscape, on the level of individual data items. This is a precondition for orientation in a more interwoven, complex setup, comprising a variety of heterogeneous data sets. In order to make users aware of the data interdependencies, all of the techniques presented rely heavily on visual links as a visual aid.

6.1 Support Within Data Set, Within View

The first technique to be discussed, Matchmaker, addresses the comparative analysis of a multi-dimensional data set in a single visual representation. Even in such a straightforward scenario (one data set, one view), the challenge for achieving orientation on the level *S1.1* is to effectively convey the relations between the compared data items. This technique is based on the ideas of Alexander Lex and has been published at the *IEEE Information Visualization* conference [Lex et al., 2010b].

6.1.1 Domain Problem: Comparative Analysis of Expression Data

Heat maps and their employment as standard visualization technique for the representation of and interaction with expression data were introduced in Section 2.3. In order to make coherent patterns in the multivariate data visible, the ordering of both rows and columns of the data matrix is essential [Weinstein, 2008]. Each column represents one dimension (*e.g.*, one experiment) and each row one data record for all dimensions. The order can be determined either automatically by a clustering algorithm, or be extracted from data properties and therefore be based on some kind of semantics. In the latter case, for instance, expression data from patients who have the same disease constitute one group of dimensions and the healthy control patients a second. Further common grouping criteria in this context could be time-series dependencies or replicates of the same experiment. These semantic groupings can then also be combined and organized hierarchically [Wilson and Bergeron, 1999].

Traditional clustered heat maps depend on “biclustering” where a clustering algorithm is run on the dimensions as well as the records. However, in many cases a hybrid variant is needed, where the dimensions are grouped according to semantics for instance, while the records are clustered by means of their similarity – or vice versa. Comparing these multiple groups of dimensions is a common analysis requirement. An interesting case occurs, for instance, when data records increase over time in one group and one wants to know if this trend is also present in the other group.

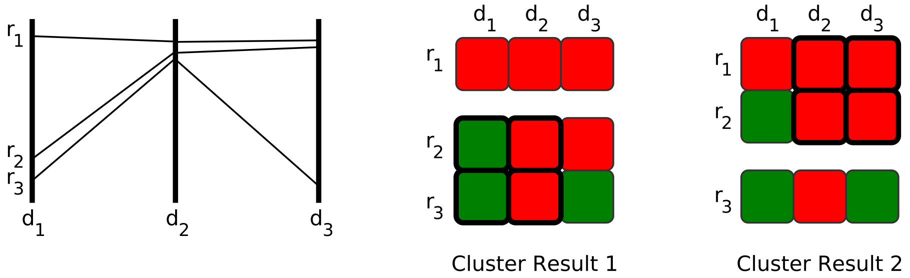


Figure 6.1: A small example demonstrating the issue of lost correlations of records among multiple groups of dimensions. Two cluster runs, with different algorithms or parameters applied, could divide the data into different groups. In this case, each result emphasizes only one particular correlation (framed), while the other one is obscured.

The usual procedure is to cluster the whole data set according to the similarity of the data records at once and show the result in a single heat map. However, this potentially obscures relations between homogeneous groups. Figure 6.1 shows a tiny sample data set with three dimensions and three records per dimension. When clustering the data set, two different clustering results are possible, depending on the algorithm chosen, the distance measure and other parameters – each of which focus on different correlations between records (bold framed in Figure 6.1). However, either the correlation between record 2 and 3 or between 1 and 2 is obscured. The more dimensions and records are involved, the worse the fragmentation of patterns gets.

Research Question 1:
How to enable a user to discover homogeneous trends within semantic groups of dimensions and also to discover trends between them?

A related problem, which additionally complicates the cluster analysis of multivariate, quantitative data, is the fact that the type and the meaning of the patterns in the heat map are highly sensitive to a series of factors [Weinstein, 2008]:

- the applied **preprocessing** (*e.g.*, normalization, clipping, *etc.*)
- the **color scheme**
- the chosen **clustering algorithm**
- the **distance metric** used

A profound knowledge of the analyst about all of these influencing factors is a precondition for extracting meaningful insights. However, bad choices in terms of these factors can lead to misinterpretations. An in-depth consideration of data preprocessing is beyond the scope of this thesis, and the color coding aspect has already been discussed in Section 5.3.1. The latter two aspects are probably the most complicated to address, as hardly any quality metrics exist that support analysts in their decision which algorithm to choose with which parameters. The standard way of judgment is to run the algorithms (or the same algorithm with different parameters) multiple times on the same data set and visually assess the alterations in the results. Although this approach can help to some extent, it is neither intuitive nor effective.

Research Question 2:
How to aid a user in visually evaluating the influences of choices regarding the clustering algorithm and its parameters on the clustering result?

6.1.2 Solution: Matchmaker

The two predominant approaches used for analyzing patterns and trends in multi-dimensional, quantitative data either rely on dimensionality reduction or else visualize the data set as a whole using techniques like parallel coordinates or heat maps. However, neither of these approaches consider semantic relations between dimensions. In the following, the Matchmaker visualization technique is introduced. It enables a comparative analysis between arbitrarily semantically grouped dimensions. The solution is based on the divide & conquer paradigm: split data into batches, process them separately and compare the batches.

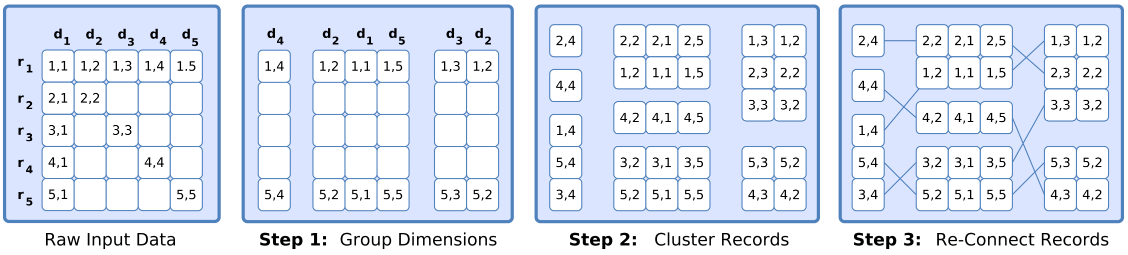
The technique's basic process is outlined in Figure 6.2. The starting point for an analysis is a multi-dimensional, quantitative data set. The process is composed of the following steps:

1. **Step 1: Group dimensions** With the help of Caleydo's grouper view (see Section 5.3.4), the user can arbitrarily join dimensions to sub-groups which are then subject to comparison analysis. Dimensions can also be duplicated, assigned to multiple groups and organized hierarchically, thus allowing a flexible composition of sub-groups reflecting the semantic relations.

2. **Step 2: Cluster sub-groups** The records in each group are clustered separately, resulting in homogeneous clusters for each of the semantic groups.

3. **Step 3: Reconnect records** The independent clustering of the sub-groups results in a loss of context between them. Therefore, in a last step, the records are connected among the sub-groups, which re-introduces the overall context.

This three-step process outlines the processing steps with regard to the data. The following sections describe how Matchmaker realizes the visual comparison of clustered groups in terms of visualization (*cf.*, step 3).



Raw Input Data **Step 1:** Group Dimensions **Step 2:** Cluster Records **Step 3:** Re-Connect Records

Figure 6.2: The three-step process of the Matchmaker technique: first, the user groups the dimensions according to meta-data (step 1), then each sub-group is clustered individually (step 2), and finally the lost connections among the sub-groups' records are re-introduced again (step 3).

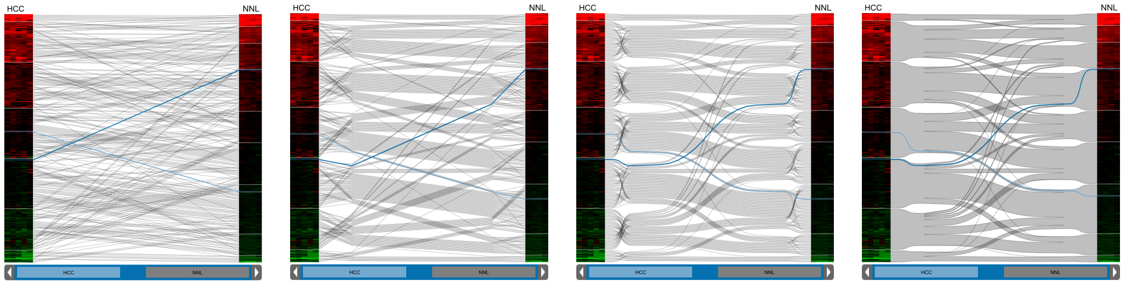
Visualization Technique

A straightforward and common way to visually compare data is to arrange multiple views side by side. However, humans are not very adept at comparison tasks. Alternatively, the comparison can be encoded in a single view where the relations between elements can be presented explicitly. The data to be compared can be arranged using various layouts: in parallel (*e.g.*, TreeJuxtaposer [Munzner et al., 2003]), circular (MizBee [Meyer et al., 2009], Circos [Krzywinski et al., 2009]), *etc.* Matchmaker falls into the first category and arranges the clustered sub-groups parallel to each other, reminiscent of parallel coordinates (*i.e.*, one group corresponds to an axis) or Parallel Sets [Kosara et al., 2006]. However, the parallel sets technique is designed to visualize categorical data, not individual data records and thus cannot be applied to this problem.

In contrast, the Hierarchical Cluster Explorer [Seo and Shneiderman, 2002] addresses a similar problem of how to compare two clustering algorithms applied to one data set. Seo and Shneiderman suggest a side-by-side arrangement of the two clustered heat maps with straight connection lines, depicting the position of each record in the opposite representation. However, as also stated by the authors, this approach works only for very small data sets ($<$ 6 dimensions, 50 records), as the line crossings soon result in visual clutter.

Matchmaker arranges the clustered heat maps – each representing one sub-group of dimensions – side by side. The clusters themselves, as well as the records inside the clusters, are sorted according to their mean value. The individual records among the heat maps are connected by lines, *cf.*, Figure 6.3(a).

In order to alleviate the problem of line crossings between opposing clusters, Matchmaker applies a specifically designed edge bundling mechanism. The strategy for minimizing inter-tree edge crossing presented for the Hierarchical Edge Bundles [Holten, 2006] cannot be applied, because it depends on a resorting of records. In our case, due to the records' assignment to the pre-sorted clusters, this is not possible. In addition, the clustering is not necessarily hierarchical, which is a precondition for the Hierarchical Edge Bundles strategy. Matchmaker reduces the number of crossings by introducing support points, resulting in wide bands (*i.e.*, parallel lines) which effectively show main trends between the compared sub-groups. Outliers are immediately visible as they are shown as thin bands or single lines with steep angles. This comes at the cost of many crossings between the heat map and its support points, but optimizes the inter-cluster crossings (see Figure 6.3(b)).



(a) (b) (c) (d)

Figure 6.3: The different styles of connecting records among heat maps: in (a) the records are simply connected by straight lines. (b) shows the result with the bundling strategy applied, reducing inter-cluster crossings. In (c) the straight lines are replaced by curves. (d) shows an optional mode where parallel lines are abstracted to ribbons.

In a next step, Matchmaker renders lines as spline curves, *cf.*, Figure 6.3(c). As an additional mode, the individual lines can be abstracted to ribbons. This not only solves the problem of Moiré patterns, but also improves the render performance. However, a details-on-demand approach, triggered by hovering over the ribbon, still allows one to see individual connections. Interactive brushing of records works by selecting whole clusters, individual ribbons and on the level of single records (lines).

The Matchmaker technique supports two modes, targeted at different analysis goals:

- the **Overview mode** presents clustered heat maps in a parallel coordinates fashion, as shown in Figure 6.4. This representation gives the user a feeling for overall trends. Groups of dimensions can be interactively rearranged. The user can focus on the comparison of two sub-groups by scrolling the mouse wheel. An animated transition helps to understand the switch to the detail mode.
- the **Detail mode** (see Figure 6.5) enables a drill-down analysis to the level of individual records. Two clustered heat maps are represented on the left and the right side of the view. The user can select individual clusters on both sides, for which a detailed heat map is opened, presenting the label of each single record. Orthogonal stretching [Sarkar et al., 1993] is applied to deal with space constraints when focusing on certain parts of the representation.

For a detailed discussion of the technique's scalability and the bundling algorithm, refer to the full paper [Lex et al., 2010b]. Furthermore, the paper elaborates on two case studies that demonstrate Matchmaker's applicability for solving the two posed research questions:

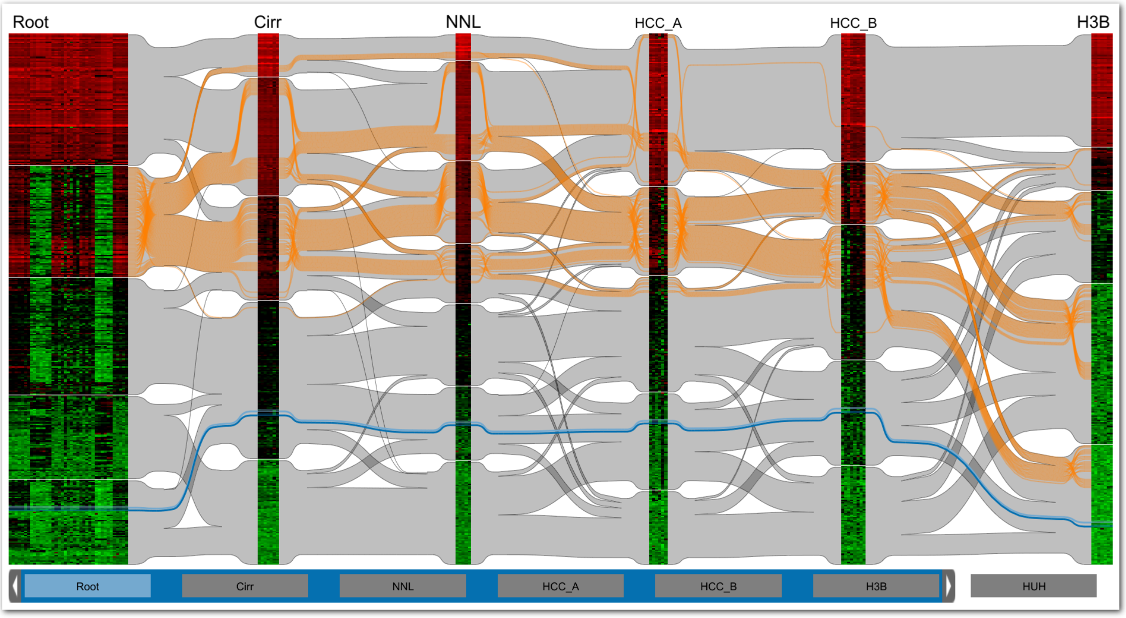


Figure 6.4: Overview mode of the Matchmaker technique. In this example, sub-groups comprise the gene expression data from patients who have the same disease [Kashofer et al., 2009]. Every dimension is shown twice: once as a part of the undivided data set on the very left, and once as part of a semantic sub-group. The relations between the clustered groups are shown by ribbons. Here, the user selects a cluster on the left, which interactively brushes its records in all other groups (orange).

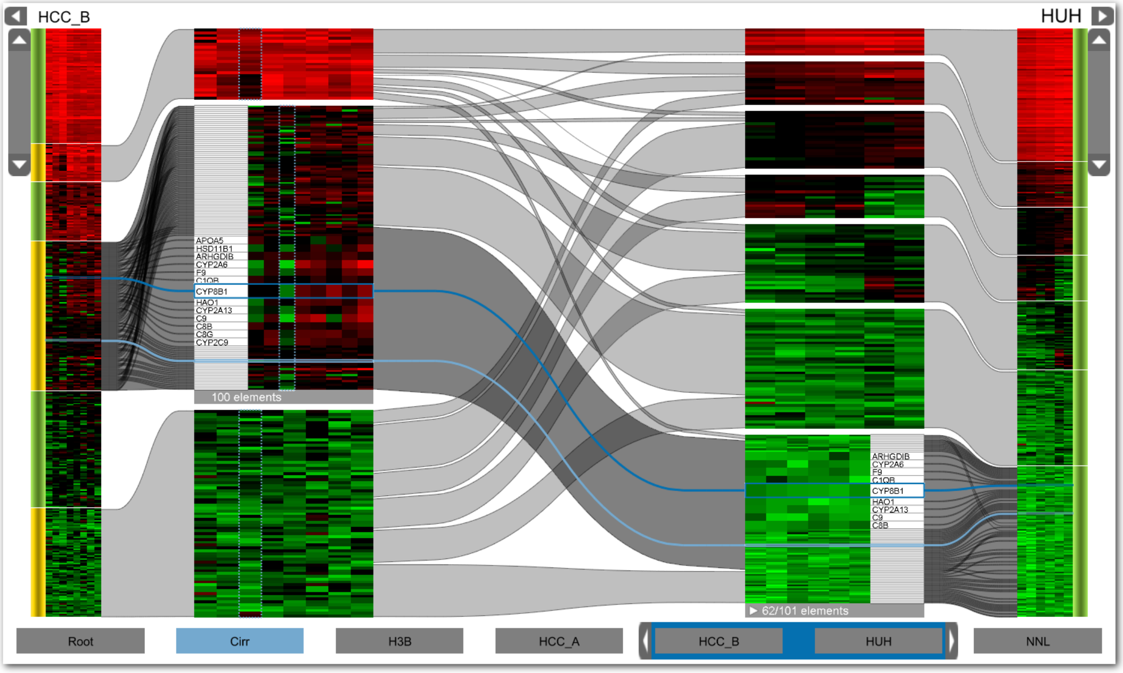


Figure 6.5: Detail mode of the Matchmaker technique. The user can select individual clusters and inspect the relation of each record to the comparison group.

the discovery of trends (*cf.*, research question 1) and the evaluation of clustering algorithms and the parameters to choose (research question 2).

Summary

Matchmaker is a technique for discovering patterns and trends in multi-variate, numerical data. This technique splits the data set into semantically connected groups of dimensions, clusters them individually per group and allows the user to interactively inspect the results in order to get insights about interesting differences or correlations among the groups of dimensions. By relating the individual data items between the compared dimension groups, it effectively addresses the domain problem stated previously while also realizing orientation support on the level *S1.1*.

6.2 Support Within Data Set, Across Views

Like in the case of Matchmaker, this section also introduces a visual analysis concept that operates on a single data set. However, due to the different type and structure of the data to analyze – small, interconnected graphs – an analyst needs to consider a multitude of views concurrently in order to draw conclusions about their interrelations. This analysis across the boundaries of individual visual representations poses new problems in terms of orientation, as the system needs to convey the dependencies of the data items among the views.

6.2.1 Domain Problem: Pathway Exploration

Pathways are graphs representing biological processes in living cells, as already introduced in Section 2.2. Traditionally, these complex, biological interaction networks were printed on large posters. Figure 6.6 shows the popular metabolic network published by Roche Applied Science [Michal, 1999]. Metabolic networks describe chemical reactions that occur in living organisms. Although designing such posters is complicated and the handling for end-users is obviously tedious, they were and are still used as an inexpensive, static way of presenting cellular networks. However, due to the progress in life science research, the overall network is now far too big to be represented on a single poster.

Large Graph Visualization

The obvious alternative to the hand-crafted, static network representation is to lay out the network using graph visualization. One option is to determine the layout automatically by means of graph routing algorithms. However, the rapidly increasing number of nodes and edges naturally leads to uninformative, giant "hairballs" [Suderman and Hallett, 2007] for naive approaches. Interactive graph exploration techniques can support the user when browsing the network and therefore alleviate the problem to a certain extent. Nevertheless,

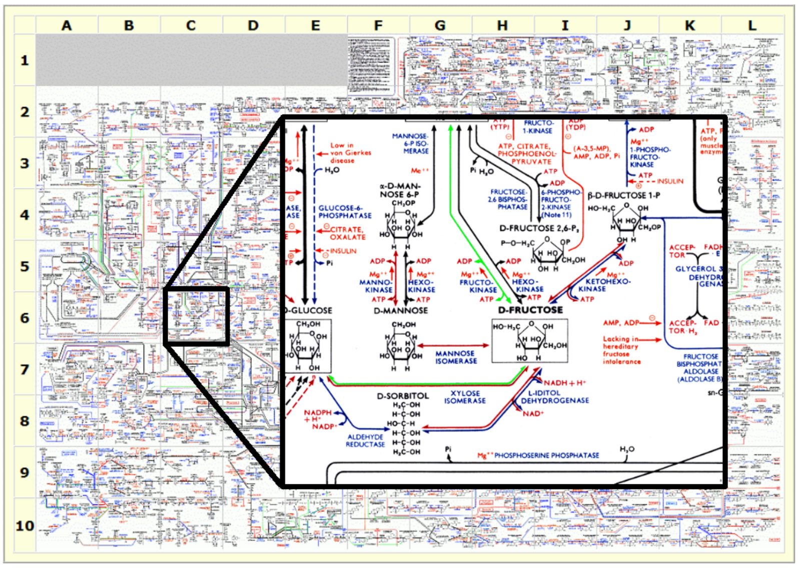


Figure 6.6: Section of the Roche Metabolic Network [Michal, 1999]. The graph is hand-routed. The poster comes with a booklet which contains an index that provides the mapping from entity to a sector on the poster.

even with an optimal layout and with the aid of interaction, handling huge graph visualizations is challenging. Furthermore, a large and dense representation is overwhelming and hinders quick and easy perception of graph relations, which is particularly problematic when trying to make sense of complex biomedical processes.

Due to the high degree of interconnectedness, [Rojdestvenski, 2003] suggests visualizing the pathway network as a 3D graph. MetNetVR [Yang et al., 2006] goes a step further by presenting a hierarchical visualization in a 3D CAVE environment. These 3D approaches suffer from the common problem that the user can easily get confused in a purely abstract, but complex 3D environment. In particular, there is no obvious three-dimensional subspace in the data that can be used for natural organization of the domain. Moreover, Virtual Reality (VR) environments such as a CAVE are expensive. However, for a visualization technique to be widely adopted, it is vital that the system runs in a standard office environment – making a VR or CAVE solution not applicable in this context. It remains to be seen whether this will change when affordable, off-the-shelf 3D-capable displays become available in the near future.

Subdivision into Small Functional Graphs

The predominant strategy over the last few years has been to artificially decompose the network according to functional context into smaller sub-graphs, with in most cases up to 200 nodes relating to one individual pathway. An example of a fundamental pathway is the *Citric Acid Cycle* where a series of chemical reactions perform a conversion of fats, proteins and carbohydrates to energy. Another fundamental one is the *Apoptosis* pathway, representing the programmed cell death. As a side-effect, the subdivision of the large network to individual pathways led to reduced complexity and made exploration easier.

However, there are advocates of both approaches, the large single network and the multiple small graphs. Automatic layout of large graphs is beneficial when the goal is to modify the graph interactively, or when the subdivision of the network into predefined pathways is unwanted [Barsky et al., 2008, Chung et al., 2005]. However, as pathway editing is not relevant for pathway exploration and sense-making tasks, the subdivision to small, functional pathway graphs is the preferred option.

Automated Pathway Layout

Considering the special pathway graph characteristics, an automatic layout determination is, even for the relatively small networks, a challenging task. One of the first attempts to dynamically model metabolic pathways was proposed in [Karp and Paley, 1994]. As pathway graphs became more and more diversified and complex over time, graph drawing approaches needed to be employed and enhanced. [Becker and Rojas, 2001] propose an algorithm that builds on the ideas of Karp and enhances them by including topological structure like cyclic or partially cyclic structures. Also hierarchy-based force-directed approaches were proposed [Tsay et al., 2010].

The fact that life scientists manage the complexity by memorizing patterns of well known pathways, strongly speaks against the application of automatic layout algorithms to pathway graphs. Frequently, pathways are immediately recognized by users because of a particular layout, such as circularly arranged nodes. The MetaViz approach [Bourqui et al., 2007] takes this fact into account and creates a metabolic network using multiple pathways. Moreover, this work also addresses the duplication problem of nodes (*i.e.*, entities performing multiple functions) by clever clustering and overlapped drawing of the graphs. However, the necessity to incorporate additional meta-information, not part of the graph's topology or attributes, requires a new solution.

Additional Meta-Information

In the small-world pathway graphs, the consideration of contextual information such as the localization of the compound in the cell or cellular structures is becoming more and more important and therefore needs to be additionally encoded in the pathway layouts. For example, the pathway in Figure 6.7 denotes the cell border as two bold horizontal lines. This trend towards the incorporation of meta-information is another strong argument against the application of automatic drawing and layout to pathways. Illustrators use grouping and annotations to convey this meta-information that is essential for understanding and memorizing the pathways. Consequently, this information is only available in hand-routed and carefully designed images.

This additional information implicitly encoded in the hand-drawn small-graph layout turns out to be essential for understanding the processes in detail. In discussions with our biomedical focus group, it turned out that the superiority of the hand-crafted layout and familiarity with the existing pathways cannot be matched by automated layouts.

Pathway Databases

Examples of widely-used public online-databases are KEGG (Kyoto Encyclopedia of Genes and Genomes) [Kanehisa et al., 2006] and BioCarta[1]. A comprehensive list of pathway resources of all kinds, including protein-protein interaction networks, metabolic pathways and signaling pathways can be found on `pathguide.org`. In the case of KEGG and BioCarta, each pathway is hand-crafted and contains only curated information backed up by peer-reviewed publications. While KEGG uses a factual node-link diagram style, see Figure 6.7, BioCarta draws pathways in a visually appealing, cartoon style, as the sample pathway in Figure 6.8 shows.

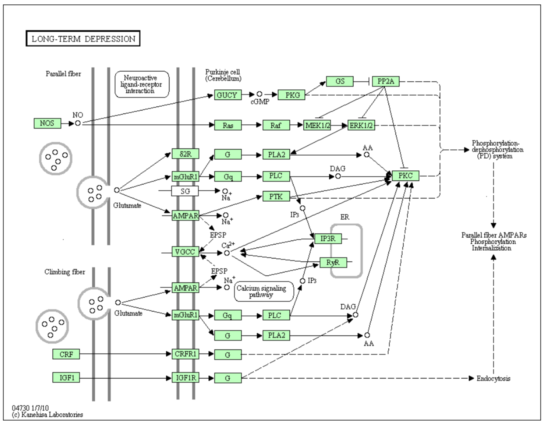


Figure 6.7: KEGG pathway map modeling *Long Term Depression* using a simple node-link drawing style. Note the structures in the background that convey meta-information.

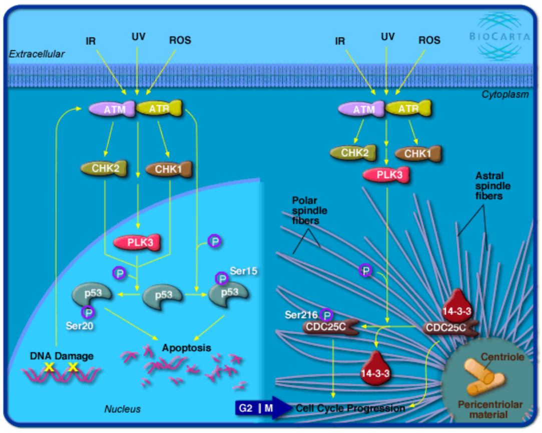


Figure 6.8: BioCarta pathway *Regulation of cell cycle progression by the Plk3 gene* drawn in a cartoon style. Additional information is encoded in the shape of the nodes as well as in the different types of links.

As of December 2010, the KEGG pathway database contains 381 reference pathways from which variants are available for all 1,482 KEGG supported organisms[2]. Based on these reference pathways over 120,000 organism specific pathways can be generated using a KEGG service. BioCarta contains 354 pathways, valid for homo sapiens (human) and mus musculus (mouse).

Rather than discarding the pre-determined pathway layout in favor of a computer-generated layout, Caleydo uses the pathway textures directly as an image-based background and enriches the static representations with interactive content rendered on top of the texture, as proposed in [Streit, 2007, Jianu et al., 2010]. This augmentation approach

[1] `http://www.biocarta.com`

[2] Current statistics on KEGG are available on: `http://www.genome.jp/kegg/docs/statistics.html`

results in an interactive, layout-preserving graph representation which allows seamless scaling as well as arbitrary 2D and 3D transformations.

Loss of Context

Genes can catalyze multiple reactions in living organisms and therefore perform different functions in the cell. However, due to the subdivision of the overall cellular network to small, functional graphs, genes can be contained several times within the same or in multiple pathways. This results in a loss of context, as the occurrence of a gene in a single pathway does not provide a comprehensive picture of the full set of cellular processes in which a gene is known to be involved. Thus, when exploring the pathways researchers must typically consider a working set of pathways at once, including the interconnections between them. A study carried out by [Saraiya et al., 2005] identifies the visualization of interconnections as a significant requirement for pathway research. Our collaboration partners confirmed that they had difficulties understanding inter-pathway dependencies when exploring collections of many small pathways using state-of-the-art tools.

Summing up, in order to understand pathways it is necessary to:

- view the pathway itself,
- view related pathways,
- view meta-information, and
- identify and explore interconnections between pathways.

State-of-the-Art in Pathway Visualization Tools

KEGG and BioCarta both provide traditional web interfaces based on lists and hyperlinks. In addition to the web content, which suffers from the mentioned loss of context between the pathways, KEGG developed a browser-based application for exploring the overall metabolic pathway network in a zoom and pan interface (see Figure 6.9). This web service tries to reintroduce the context by showing an abstraction of the whole network. However, they cannot handle the 1:n connections between pathway entities.

In addition to these limited web-based solutions, dedicated pathway visualization tools have been developed. [Klukas and Schreiber, 2006] describe an approach that arranges KEGG pathways and adds inter-pathway edges, resulting in a mixture of static layouts of hand-routed pathways and automatically drawn layouts. The latter approach works well for a small number of pathways (2-3), but as the number increases, nodes become small and too many links between pathways result in visual clutter.

Since KEGG and BioCarta use nested pathways (*i.e.*, pathways represented as single nodes inside pathways), the problem of visualizing multiple pathways can also be interpreted as a problem of browsing hierarchical graphs. [Klukas and Schreiber, 2006] addresses this issue by combining multiple pathways in a single network that supports

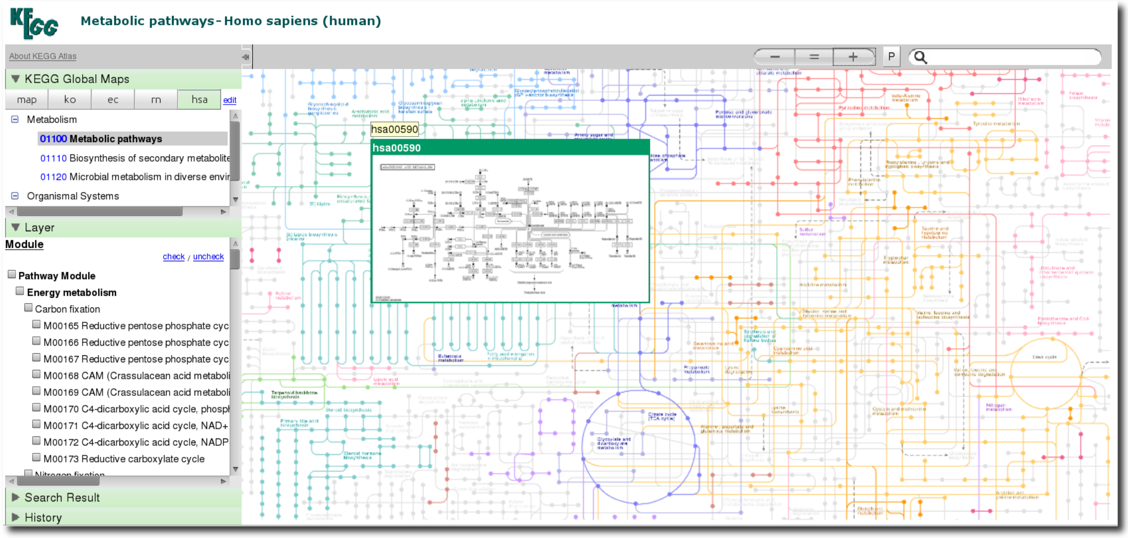

KEGG
Metabolic pathways - Homo sapiens (human)
About KEGG Atlas
▼ KEGG Global Maps
map
ko
ec
rn
hsa
edit
⊟ Metabolism
01100 Metabolic pathways
01110 Biosynthesis of secondary metabolite
01120 Microbial metabolism in diverse envir
⊟ Organismal Systems
▼ Layer
Module
check / uncheck
□ Pathway Module
□ Energy metabolism
□ Carbon fixation
□ M00165 Reductive pentose phosphate cyc
□ M00166 Reductive pentose phosphate cyc
□ M00167 Reductive pentose phosphate cyc
□ M00168 CAM (Crassulacean acid metaboli
□ M00169 CAM (Crassulacean acid metaboli
□ M00170 C4-dicarboxylic acid cycle, phosph
□ M00171 C4-dicarboxylic acid cycle, NAD+
□ M00172 C4-dicarboxylic acid cycle, NADP
□ M00173 Reductive carboxylate cycle
□ Nitrogen fixation
▶ Search Result
▶ History
− = + P
hsa00590
hsa00590

Figure 6.9: The KEGG Atlas browser application allows users to explore the global metabolic pathway network with a zoom and pan interface. On mouse-over an individual pathway is opened as a pop-up overlay.

interactive level of detail change by expanding and collapsing the pathways from/to single nodes. While the system partly solves the users' needs in terms of navigation, the relations between the graphs, depicted by connection lines, get cluttered easily.

Many further visualization tools addressing pathway visualization have been proposed over the last years. While Pathway Studio[3] is a prominent example of a commercial software product, the Cytoscape Network Analysis and Visualization software[4] is available as open-source. Only recently, [Gehlenborg et al., 2010] list and categorize the state-of-the-art in pathway visualization tools regarding their features. However, these solutions do not satisfactory address the following requirement:

Research Question:
How to re-introduce the lost context between the visualizations without introducing visual clutter while keeping the user oriented?

6.2.2 Solution: Jukebox - Stacking of Interconnected Graphs

This section introduces a visualization concept called Jukebox which allows for an efficient and interactive navigation inside the network of connected graphs. It respects the meta-information available in the hand-crafted pathways but turns the static pathway layouts into fully interactive representations. The setup allows the analyst to manage a working set of pathways in a 2.5D stacked representation which makes inter-pathway connections

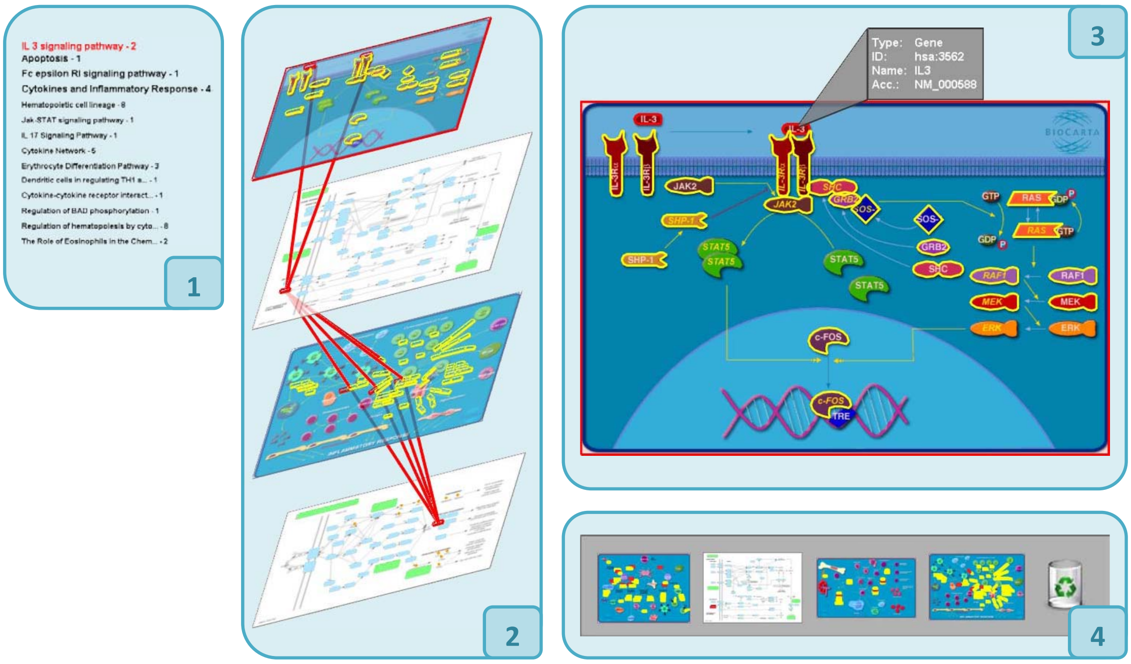
[3] Ariadne Genomics, Inc., USA, `http://www.ariadnegenomics.com/products/pathway-studio`
[4] `http://www.cytoscape.org`

evident and therefore maximizes the use of screen real estate while avoiding clutter. The stacking and interlinking of pathways was initially introduced as a poster at the *IEEE Information Visualization* conference [Streit et al., 2007]. Subsequently, the Jukebox approach was published as a full paper at the *EuroVis* conference [Streit et al., 2008].

Jukebox Setup: Graph Stacking in 2.5D

The four levels of the Jukebox concept facilitate the management of and navigation within the set of interconnected pathways:

- Level 1: **Pathway pool list**
- Level 2: **2.5D stacked layer view**
- Level 3: **Graph under interaction view**
- Level 4: **Memo pad**



IL 3 signaling pathway - 2
Apoptosis - 1
Fc epsilon RI signaling pathway - 1
Cytokines and Inflammatory Response - 4
Hematopoietic cell lineage - 8
Jak-STAT signaling pathway - 1
IL 17 Signaling Pathway - 1
Cytokine Network - 5
Erythrocyte Differentiation Pathway - 3
Dendritic cells in regulating TH1 a... - 1
Cytokine-cytokine receptor interact... - 1
Regulation of BAD phosphorylation - 1
Regulation of hematopoiesis by cyto... - 8
The Role of Eosinophils in the Chem... - 2

Type: Gene
ID: hsa:3562
Name: IL3
Acc.: NM_000588

Figure 6.10: The Jukebox view combines (1) a textual list menu for browsing related pathways by name, (2) with an interconnected pathway stack, (3) an area designated for a detailed examination of a graph, and (4) a memo pad.

Figure 6.10 shows the four levels of the Jukebox setup annotated in a screenshot. The setup works analogous to a jukebox where audio records can be selected from a larger collection and loaded to the turntable. A list of pathways is presented as a compacted, textual list containing the pathway names (Level 1). By selecting an entry in the list, the graph is loaded to the intermediate Level 2, where a predefined number of graphs is shown in a 2.5D stacked layer representation. Level 3 shows a large version of the pathway for interactive inspection.

With the Tecate system [Kochevar and Wanger, 1995] suggest projecting 2D views in a 3D scene (see Figure 6.11). The individual views are arranged in a pyramid-shaped stack that allows users to browse the history of a hyperlink system by flying through the scene. The Jukebox builds upon this idea and adopts a 2.5D layout where the additional ordinal dimension is used to manage and arrange the view representations. The Jukebox concept employs a stack of graphs arranged in a 2.5D layout to densely pack pathway information in the available screen space, but also relate multiple planar graphs to one another. Individual pathways are scaled down and also compressed due to the orthographic (tilted) view. However, signature features and proportions familiar to the expert are retained, which makes the pathways still highly recognizable despite the perspective foreshortening.

The Jukebox setup was inspired by the work presented in [Brandes et al., 2004] where similar pathways are stacked on top of each other to visually differentiate them (see Figure 6.12). However, their methods do not provide solutions for showing relations between the layers. In contrast, our approach uses the additional relation afforded by 2.5D to link multiple ordered pathway layers. The relations between elements in different pathways are visualized using straight lines between the layers [Streit, 2007]. In comparison to the VisLink approach [Collins and Carpendale, 2007] that also facilitates inter-plane edges, the Jukebox setup provides a solution for managing related 2D visualizations in a hierarchical way. The concept of inter-layer connections was also adopted later on by the Arena3D system [Pavlopoulos et al., 2008] for relating content between various biological levels, each represented as a plane in a 3D scene.

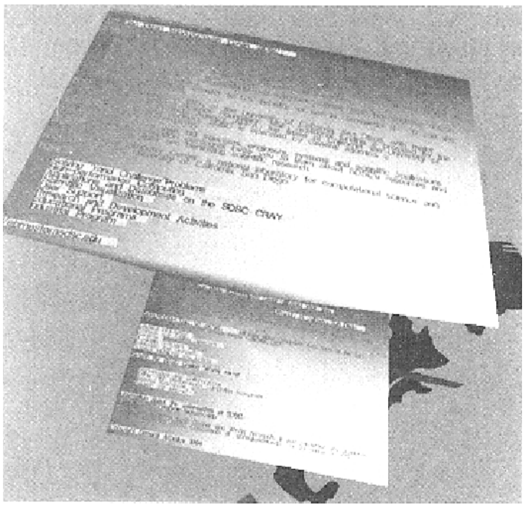


Figure 6.11: The Tecate system stacks hypertext documents [Kochevar and Wanger, 1995].

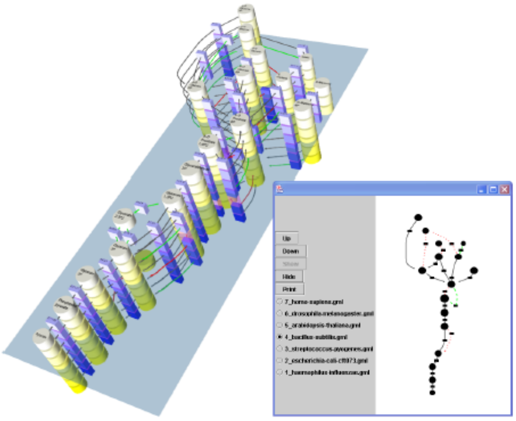

Up
Down
Show
Hide
Print
○ 7_homo-sapiens.gml
○ 6_drosophila-melanogaster.gml
○ 5_arabidopsis-thaliana.gml
● 4_bacillus-subtilis.gml
○ 3_streptococcus-pyogenes.gml
○ 2_escherichia-coli-cft073.gml
○ 1_haemophilus-influenzae.gml

Figure 6.12: Stacked metabolic pathways allow the user to identify changes between the graphs [Brandes et al., 2004].

In the context of pathways, the connection lines between layers enable a fast identification of identical nodes in the whole network. Level 3 in Figure 6.10 shows the gene *IL3* that is selected by the user in the *IL 3* signaling pathway. Consequently, all representations of that specific gene are highlighted and interactively connected. Selecting a particular gene allows the user to quickly determine its global relevance to various aspects of the working set. Circumstances under which a gene appears in multiple pathways as part of an identical chain of reactions can also be discovered. The user can then choose a different pathway from the stack, which is exchanged with the pathway in the main interaction view. Moving a pathway up or down the hierarchy to another Jukebox level is visually supported by animated transformations [Shoemake, 1985]. It was found that users value the continuous transitions when the complex networks, which are scattered over multiple pathway views, demand their full attention.

There are multiple reasons why an entity can be represented multiple times in a set of pathways. One reason has to do with layout considerations. Another possibility is that a particular gene is catalyzing a specific reaction in a large variety of biological processes in the cell. As a consequence, genes appear several times in various pathways. Without global selection it is hard to identify such situations. We therefore provide a mechanism that automatically searches the whole pathway pool for the selected entity. The resulting pathways are shown in the pathway list (Level 1) by displaying their names plus a score which is based on how often selected genes occur in that particular pathway. According to this score the most relevant pathways (*e.g.*, highest score) are moved to the stacked pathway view, where the user can continue to explore inter-pathway relationships.

For larger-scale problems, the automatic management of the stack based on a least frequently used policy can create the undesirable situation that a pathway vanishes from the stack, but may be needed again later and must be manually retrieved. We therefore provide the memo pad (see Level 4 in Figure 6.10), an area of the screen where the user can place important pathways for semi-permanent safekeeping. Storing and retrieving pathways works by simple drag-and-drop. The memo pad not only stores the graph, but also the current selection of nodes, so that a particular working state can be completely restored instantaneously. The memo pad and the stacked pathway view are complementary: While the memo pad is designated to hold pathways for an entire analysis session, the 2.5D layered view is a volatile stack that may be changed during the dynamic loading of dependent pathways.

Neighborhood Visualization

Algorithms for the calculation of adjacencies in graphs and their visualization are well researched. As mentioned before, nodes in pathways can be part of the same graph multiple times as well as be incorporated in other pathways. Caleydo combines the dynamic loading of dependent graphs with the highlighting of adjacencies. After the user selects a node, the system presents all pathways that contain this entity. Dijkstra's graph search algorithm [Dijkstra, 1959] is then applied to all instances of the entity in the selected working set of pathways. As a result, the neighborhood is propagated throughout all involved graphs (in configurable depth). This extended adjacency visualization makes a comprehensive exploration across the boundaries of multiple pathways possible, as shown in Figure 6.13. Searching the neighborhood in all pathways simultaneously is extremely valuable, since it enables the user to reveal hidden biological dependencies and to detect reaction cascades in several pathways without being forced to go through all pathways manually.

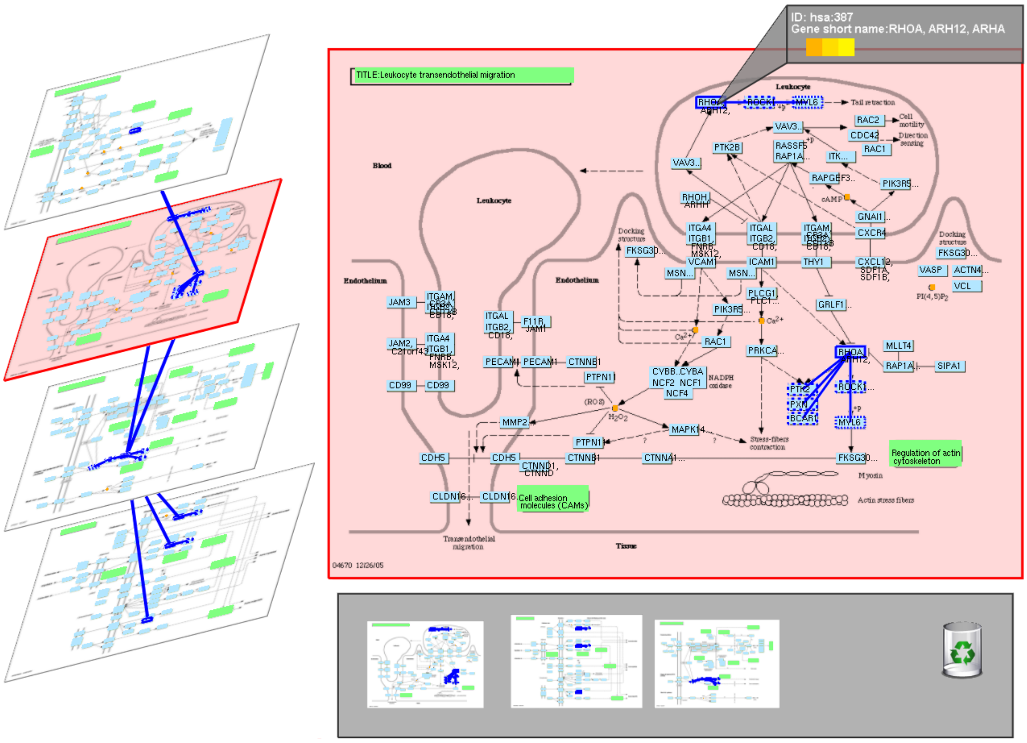

ID: hsa:387
Gene short name:RHOA, ARH12, ARHA
TITLE:Leukocyte transendothelial migration
Leukocyte
Blood
Endothelium
Tissue
Regulation of actin cytoskeleton
Cell adhesion molecules (CAMs)
Transendothelial migration
04670 12/26/05

Figure 6.13: Propagation of a signal in the pathway network. Upon selection of a gene, the system highlights the flow of the signal – not only within the local pathway where the selection was triggered, but through all of the loaded pathways in the Jukebox stack.

Discussion

Although the methods presented have the potential to facilitate knowledge acquisition in huge relational networks, the system is subject to restrictions. A good portion of the pathways has a size and complexity which is suitable for our design. However, some of the graphs are degenerated in size, which is problematic for the stacked visualization. The mixing of extraordinarily small graphs with big graphs can be aesthetically displeasing, and even disturb the user's ability to interpret the visualization. These effects can be mitigated by adaptive scaling, but only to a limited degree.

We tested the Jukebox setup in multiple configurations, varying the degree of scene customization permitted to the user. It turned out that the most restrictive setup was perceived to be the best. While a rotation of the pathway stack can give a better perception of the line connections between the layers, most users became disorientated. Consequently, the stack planes are tilted at a fixed angle of 60 degrees and the camera cannot be altered by the user. During tests with expert users we turned off the connection lines between the graph planes in the stacked visualization while only using linking & brushing for highlighting the selected nodes. Many users complained about the missing edges between the layers. Further user tests showed that the maximum stack size should not exceed five planes. Otherwise users began to feel overwhelmed by too many graphs at the same time. Nevertheless, this restriction of the stack is inherently absorbed by the Jukebox's hierarchical concept.

6.3 Support Across Data Sets, Across Views

Up to now, this chapter has introduced two visualization concepts, each of them facilitating the analysis of a single data set. The next section progresses regarding the degree of heterogeneity, since the scenario includes two data sets that differ in terms of both type and structure. In order to keep a user oriented in the information landscape (*cf.*, support level *S1.1*), an analysis system needs to not only communicate the interdependencies between the individual views, but at the same time make the user aware of which visualization contains data from which data set. In the following, two techniques will be discussed which analyze multi-dimensional, numerical data (gene expressions) in the context of a network of small, interconnected graphs (pathways).

After laying out the domain problem that is subject of the analysis, the applicability of the Jukebox for this purpose will be evaluated and the identified shortcomings will be addressed by a novel visualization concept – the Bucket.

6.3.1 Domain Problem: Expression Data in the Context of Pathways

In order to understand the function of genes and their roles in diseases, a simultaneous consideration of gene expression data and pathways is crucial. Typical data analysis questions are:

- In which biological processes is a gene involved that is significantly up- or down-regulated between two groups of patients?
- Is this conspicuously regulated gene involved in multiple functionally related processes?
- How does the gene regulation influence the chain of reactions in a pathway? For instance, a significantly down-regulated gene at the beginning of a pathway can in fact make all the following nodes irrelevant.

Typical Workflow

After being approached by our partners from the Medical University of Graz, we analyzed the goals they were trying to achieve, and discovered two distinct workflows: The first is a pathway-centric approach, the second concerns the analysis of gene expression data with hypothesis generation and quick plausibility checks.

- **Pathway-centric Analysis Workflow**
  In the pathway-centric approach, the expert is interested in a specific biological process, like the development of colorectal cancer. The starting point for an analysis could be the KEGG *Colorectal Cancer* pathway. The user explores the interdependencies of this pathway with other pathways. When simultaneously exploring the pathway and gene expressions from multiple samples of cancerous tissue, the expert can detect differences in the gene expressions of groups of samples. Such a variation can indicate different sub-types of the disease or response to treatment in a time-series analysis.

  Since not the whole set of gene expression values (about 30,000) has a mapping to a pathway, the initial expression data set can be reduced by the genes that are not represented in at least one pathway. In the case of KEGG and BioCarta, these are about 5,000 genes.

- **Expression-centric Analysis Workflow**
  In a gene expression-centric approach, the expert analyzes the expression data first. In this case, knowledge about the clinical factors that distinguish the different experimental conditions or patients is essential. For example, an expert could arrange the data in such a way that patients with short disease-free survival are grouped. He then looks for differentially expressed genes, supported by filters and analytical tools such as clustering. Such evidence may lead to a hypothesis which can be checked for plausibility by analyzing the biological context (*i.e.*, pathways or literature) of the differentially expressed genes. Only plausible hypotheses are subjected to expensive clinical studies.

Existing Approaches

The visual analysis of ~omics expression data in the context of pathways has emerged as a hot research topic over the last couple of years [Gehlenborg et al., 2010]. [Saraiya et al., 2005] discuss the design space for relating expression data to pathway graphs, allowing the related work to be classified as follows:

- **Option 1: In-place mapping of one or more data values onto the nodes**
  The node acts as a glyph for conveying the linked information. Color coding the nodes according to one or more experiment is the most commonly used strategy. Both research (*e.g.*, [Lindroos and Andersson, 2002], GScope [Toyoda et al., 2003], PathwayExplorer [Mlecnik et al., 2005], SpotXPlore [Westenberg et al., 2010]) and commercial tools (*e.g.*, GeneSpring[5], Pathway Studio) use this direct mapping approach.

[5] Agilent Technologies, Inc., USA, `http://www.agilent.com/chem/genespring`

However, this approach only works under certain conditions for a small number of experiments ([Lindroos and Andersson, 2002] claim eight). Due to the tiny size of the node, a large number of experiments become indistinguishable. Moreover, the text on the node itself is completely occluded, and the method is only usable for rectangular nodes, such as those used in KEGG, but not compatible with free-form shapes, such as those common in BioCarta. Another strong argument against the in-place mapping is that KEGG contains many nodes which are encoded by multiple genes. This multi-mapping is impossible to visualize by color coding the node. Methods like in-place tool-tips [Streit et al., 2008], complex glyphs (*e.g.*, [Unger and Schumann, 2009]) or even in-place embedded views (*e.g.*, GeneSpring draw miniature heat maps on the position of the nodes in the graph) fall into the same category and therefore all suffer from the limited space to some extent.

- **Option 2: Small multiples** [Tufte, 1983, p.170]
  A series of small views representing the graph is rendered side by side, whereby each view encodes one experimental condition. Tools that facilitate small multiples for relating expression to pathways are for instance Cerebral [Barsky et al., 2008] and Pathline [Meyer et al., 2010]. While this approach does require a lot of screen space and therefore does not scale to larger numbers of experiments, it works well for a limited (less than 20) number of small graphs that do not have multiple genes encoding one node. Comparability between the different experiments suffers, however, since each small view encodes only one experiment.

- **Option 3: Multiple coordinated views**
  The most flexible way of relating expression data to pathway graphs is the utilization of multiple coordinated views, where the pathways are related to a separate, full heat map view. In this case, the depiction of multiple genes encoding one node as well as a simultaneous mapping of multiple expression values (as needed for a time-series experiment) is unproblematic. However, the drawback of classic multiple coordinated view solutions is the additional effort required on the part of the user to relate the information between the views, even with the visual linking techniques applied – as discussed in Section 3.3.1.

It is possible to combine these three alternatives. Recently, Gehlenborg *et al.* [Gehlenborg et al., 2010] as well as Unger [Unger, 2010] thoroughly reviewed the utilization of these alternatives in the current state-of-the-art. However, since all of the mentioned alternatives have their shortcomings, they cannot fully address the analysis needs to a full extent – formulated as follows:

Research Question:
How to simultaneously present multi-dimensional, quantitative data in the context of a collection of interconnected graphs in a simple, yet effective way?

6.3.2 Solution: Jukebox Revisited

The basic idea and functioning of the Jukebox was introduced in Section 6.2.2. However, at that point the Jukebox was only introduced as a concept for the exploration of pathways – a homogeneous data set. The technique needs to be re-evaluated to see, whether or not it can address the additional requirement of relating gene expression data. In a first approach, the direct mapping approach (Option 1) was implemented, as explained in the following sample analysis session.

Jukebox Sample Analysis Session

In this section the interaction with the Jukebox was briefly outlined by means of a sample session. Figure 6.14 provides an exemplary workflow documented by a series of screenshots. First the user triggers a search action for the *PTK2* gene that is, according to a statistical pre-processing, suspected to be relevant for a disease from which the experimental tissue samples are taken. The system then loads all pathways that contain the selected gene to the Jukebox setup (*cf.*, (a) in Figure 6.14). Next, the user starts to investigate the *Erb* signaling pathway in detail. Obviously the *PTK2* gene is located at the end of a signal cascade. By performing the in-depth adjacency visualization, previously unknown relationships emerge (Figure 6.14). By switching to the *Focal Adhesion* pathway from the top of the stack (c), the user can further investigate the neighboring genes. In (d), the user selects an adjacent node of *PTK2* on which multiple genes are mapped. Nodes with multi-mappings are depicted in a predefined color that is not contained in the color map used for encoding the expression values (cyan). When hovering over the node, a star-shaped, in-place tool-tip containing the requested information about the genes is opened as an overlay. When selecting one of these genes, the system again performs the dynamic pathway search and loads a new set of pathways into the Jukebox setup.

Discussion

Although the interaction with the Jukebox was well received by our life science partners, the concept suffers from two major drawbacks:

- Relations between views that are not adjacent in the stack are not directly connected. [Collins and Carpendale, 2007] also identify this issue as a yet unsolved problem for visually linked views.
- There are no visual links between the view in focus and the views in the stack. One solution is to show a duplicate of the view in focus on the top of the stack. This approach makes it possible to show connections to adjacent views, however, the duplication of the view wastes screen space and is not intuitive. Also, additional visual cross-links between the view in focus and the stack is not a real alternative, as it would cause additional visual clutter.

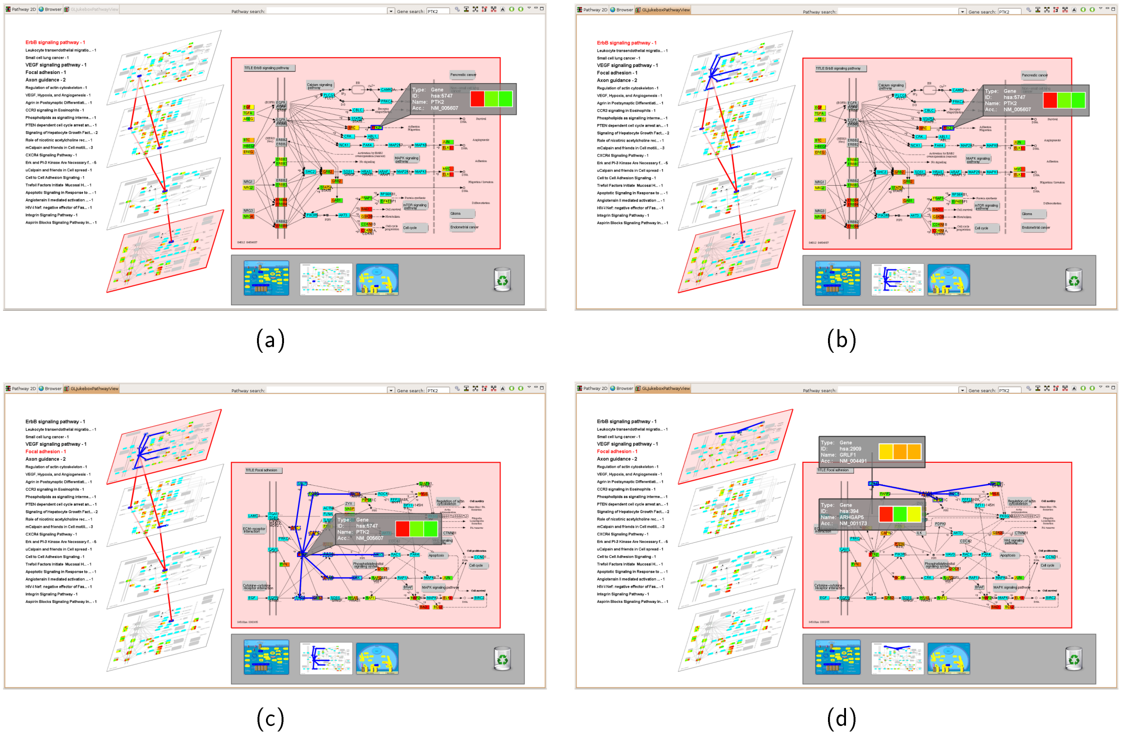

(a)
(b)
(c)
(d)

Figure 6.14: This series of screenshots depicts an exemplary part of the visual exploration process with the Jukebox. In (a) the system presents the pathways for the PTK2 search query. By investigation of the adjacencies in the graph stack the user can identify other genes connected to PTK2 which are not present in the local pathway context (b). In (c) the user switches to the topmost graph in the stack. (d) shows multiple genes mapped on a neighboring enzyme.

In addition to these weak points, the applied direct expression mapping was rather rudimentary and does not scale well for the reasons already stated (see Section 6.3.1). In principle, it would be possible to add an additional linked heat map view to the Jukebox stack. However, also in this case the two drawbacks would not be overcome.

In the following section, a novel concept called the Bucket will be presented which aims to resolve the shortcomings of the Jukebox.

6.3.3 Solution: Bucket

Multiple linked views have proven valuable for thoroughly comprehending complex data sets. By interactively updating corresponding data in all views simultaneously, the investigation of interrelated aspects of a problem becomes feasible. However, the presentation of views side by side is restricted by the available screen space. High-resolution displays and multi-monitor configurations can increase the number of available pixels, but are ultimately limited by the maximum angle conveniently observed by a human. Clearly, novel compact viewing arrangements are required. Therefore, a spatial setup of multiple 2D visualizations embedded in a 3D scene – called the Bucket (see Figure 6.15) – was developed.

The Bucket was briefly introduced as part of the application note paper on the Caleydo system [Streit et al., 2009a]. The visualization concept along with an evaluation was published at the *IEEE PacificVis* conference in [Lex et al., 2010a]. [Mueller et al., 2009] discuss a workflow for using Biobanks, where the Bucket is an integral part of the analysis setup.

The Bucket is used to show pathways and contextual gene expression information in a heat map. The heat map in the Bucket contains only those genes that occur in at least one of the pathways. By clustering and sorting the genes every time a pathway is added or removed, the genes with the highest value (*i.e.*, the highest average of a cluster over all experiments) are always on top (*cf.*, Figure 6.15). There are several ways to load pathways into the Bucket, for example by keyword search for a specific pathway, or by loading a pathway containing a particular gene. The new pathways are placed in the Bucket, where the relations can be explored.

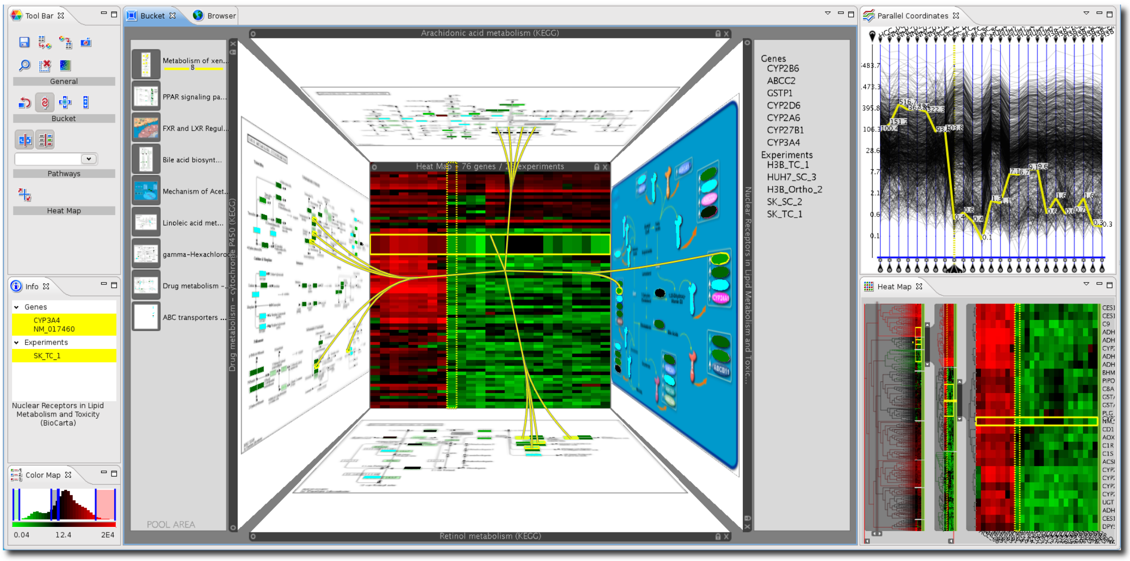
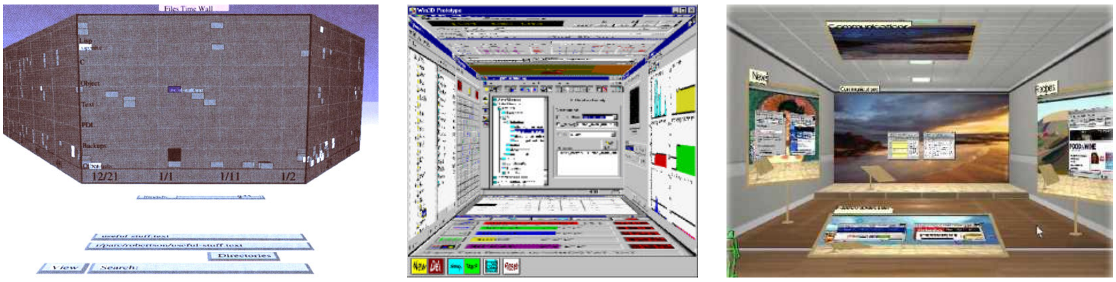


Figure 6.15: The Bucket: a 2.5D visualization approach for managing and interacting with multiple interconnected views. The screenshot shows a sample analysis where the user has selected a gene of interest. The pathways where the gene plays a role are dynamically loaded to the Bucket. The Bucket draws visual links between all occurrences of the gene in the pathways as well as the corresponding row in the contextual heat map. While the separate parallel coordinates and clustered heat map view (on the right side of the workbench) contain the whole expression data set, the contextual heat map rendered in the center of the Bucket shows only the genes that are part of any of the currently loaded pathways inside the Bucket.

Bucket Concept

The Bucket is a metaphor for a view arrangement where multiple related views are rendered on the inner sides and the rim of a square bucket. The users' viewport is restricted to a top view into the Bucket. The bottom of the Bucket contains the view in focus. Contextual views are rendered onto the second level, the Bucket walls. A third level, the rim of the Bucket, holds down-scaled, linked view representations that are related, but not currently in the user's focus, as well as genes, experiments or pathways that have been bookmarked previously.

The Bucket's arrangement of views in a 3D scene takes advantage of the spatial dimension by using it for multiple levels of focus+context. The visual arrangement loosely resembles the Perspective Wall [Mackinlay et al., 1991] (see Figure 6.16(a)), which also applies view stretching and shrinking as a distortion technique. However, the walls of the Bucket are not used for contextual information drawn from the same visualization. Instead, we present separate, but interrelated views in a space-saving arrangement which lends itself to visual linking due to the compact hierarchical arrangement of views. In principle, the arrangement of views is reminiscent of the Perspective Tunnel [Mitchell and Kennedy, 1997] (see Figure 6.16(b)) and the Task Gallery, a 3D desktop manager [Robertson et al., 2000] (Figure 6.16(c)). However, while these systems do make it possible to map 2D content in a similar shaped fashion, they do not provide a solution for inter-linking the content.



(a) (b) (c)

Figure 6.16: Early interactive techniques that arrange 2D content in a 3D scene: (a) Perspective Wall [Mackinlay et al., 1991], (b) Perspective Tunnel [Mitchell and Kennedy, 1997] and (c) Task Gallery [Robertson et al., 2000].

During the development of the Bucket concept, we experimented with different Bucket shapes. We decided to use the variant with a square bottom because of its simplicity and efficient use of screen space. Unlike a hexagon or octagon, a square does not waste space in the corners assuming a rectangular shape of the views. The square allows one focus and four contextual views, which we found sufficient for most problems. The Bucket adapts to the available window by unfolding, if possible. This results in less perspective distortion for the side views when used with landscape-type screen resolutions, but also in unused screen space in the corners.

Zooming The zoom feature (see Figure 6.17) is restricted to two predefined z-values, which were found sufficient after some experimentation. It enables the most detailed inspection of and interaction with the visualization in focus. A zoomed visualization shows all available detail, such as labels and UI elements. A zoom action is triggered by turning the mouse wheel. It is visually supported by an animated camera flight. The contextual views from the wall are placed either to the right or on top of the focus view, depending on the window geometry, thus preserving the contextual information. The rim, containing a list of bookmarked genes (right) and the other related pathways (left), is still visible.

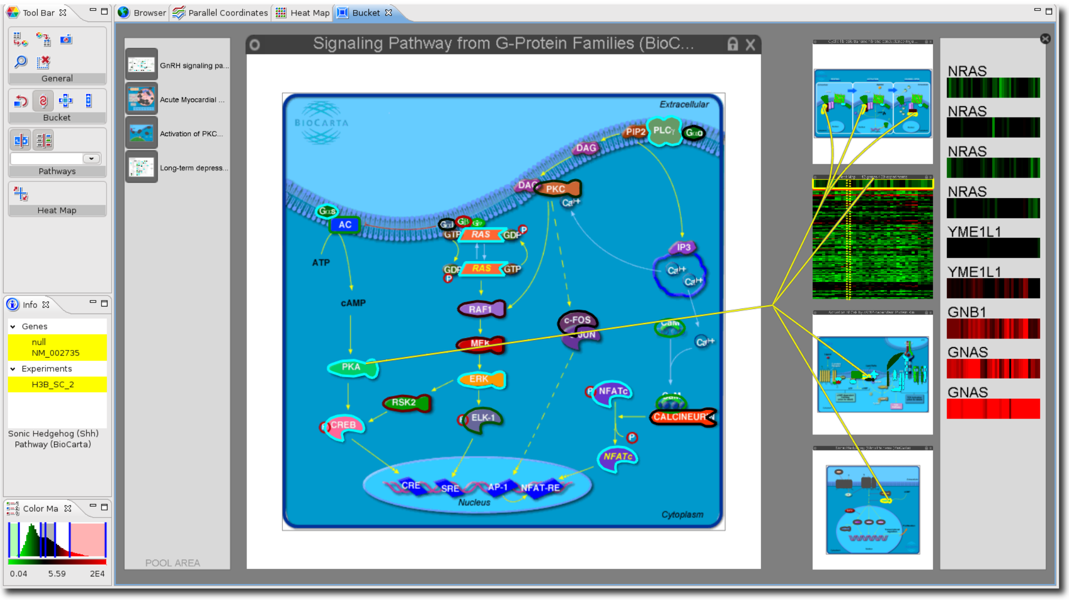
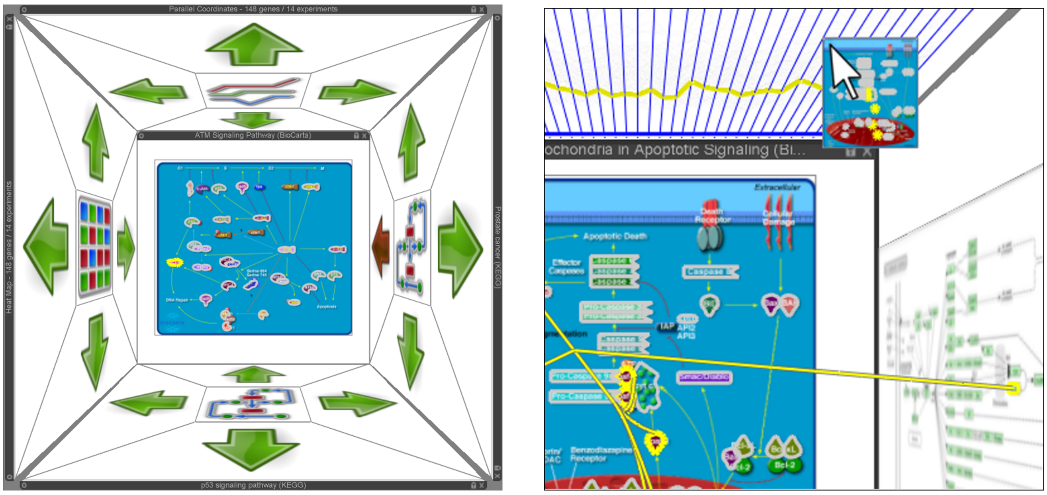


Figure 6.17: The Bucket when zoomed in, showing a 2D arrangement of focus and context views. 2D visual links connect identical entities between views.

View Navigation A very restrictive set of navigation operations turns out to be sufficient, providing the benefit of low cognitive load during navigation. VisLink [Collins and Carpendale, 2007] addresses the issue of navigating in a 3D multi-view arrangement by providing hotkeys for predefined camera positions, while still allowing full 3D navigation – leaving the efficient use of available screen space and the navigation to the user. However, our experiences indicate that a more restricted approach is beneficial – the nature of the Bucket layout does not require full navigational freedom. Views can simply be moved by drag-and-drop. The Bucket supports two different ways to rearrange views: by using a navigation overlay, see Figure 6.18 (a), or by using drag-and-drop (b).

The navigation overlay is activated on a right mouse click, which makes arrows, pointing in the directions where the view can be moved to, appear. When the target of a move action is already occupied by another view, the two visualizations are swapped. For moving the visualization planes in 3D, animated transformations are used which allow the user to visually follow the action. In addition, views can be removed by clicking the remove button in its title bar.


Parallel Coordinates - 148 genes / 14 experiments
ATM Signaling Pathway (BioCarta)
Heat Map - 148 genes / 14 experiments
Prostate cancer (KEGG)
p53 signaling pathway (KEGG)
ochondria in Apoptotic Signaling (Bi...
Apoptotic Death
Extracellular
Death Receptor
Caspase
Bcl-2

(a)

(b)

Figure 6.18: The rearrangement of views inside the Bucket is achieved by either clicking overlaid navigation controls (a) or by drag-and-drop (b).

Visual Links in the Bucket The main goal of the Bucket is to visualize the relations between views and the properties of a selected entity. This is done by applying visual links to show relations between elements among views, as described in Section 3.3.1. In the case of genes, we have multiple occurrences in several different views, since a gene can occur in several pathways. One property of a gene is its expression regulation. In contrast to [Collins and Carpendale, 2007], the views do not contain links between similar entities, but between different representations of the same element.

The intensive use of visual links is very sensitive to optimal spatial positioning of the views relative to each other. For instance [Collins and Carpendale, 2007] defer the optimal placement of views to the user. While this approach does allow flexible setups, it is not necessarily the most efficient, since the user may spend a significant amount of time simply trying to find the optimal placement in 3D space.

As already discussed, the Jukebox suffers from the two major drawbacks: no visual links to the view in focus; and the problem of linking between non-adjacent views in the stack. Due to the special arrangement of views, the Bucket avoids these pitfalls. Direct and short links from the view at the bottom (focus) connect related information contained in the views placed at the walls.

Multi-level edge bundling is used to reduce visual clutter. Edges are bundled first on a per view basis, which is important since a view often contains multiple entries. The bundled nodes from the views are then joined in a common point calculated on the fly.

Suitability of Visualization Methods for the Bucket In principle, the Bucket can be used to show all visualization methods implemented in Caleydo. However, not all of them are equally well represented in such a setup. Basic properties of visualization techniques that make them suitable for distorted analysis were identified:

- It contains data that has many relations to other views in the setup.
- Its visual encoding does not suffer severely from the distortion, as for example parallel coordinates do due to perspective foreshortening.
- It makes use of consistent spatial encoding, thus allowing a user to infer knowledge based purely on the location of an element.

Therefore, visualization techniques such as maps, tree maps, static graphs, heat maps, scatter plots *etc.* are well suited for use in the Bucket. However, in order to make optimal use of the available space for pathway analysis, the Bucket implementation in the Caleydo framework is limited to show pathways and a contextual heat map.

Relating Pathways to Gene Expression By linking the contained heat map to the selected genes, the Bucket permits a new approach to the problem of bringing pathways into context with gene expression. The linking works for all shapes and sizes of nodes and also for 1:n relations. The number of experiments is only limited by the number of distinguishable elements in the heat map.

In pathways, nodes can be represented 0-n times, and one node can encode several genes. The linked heat map always highlights the genes which are mapped to the selected pathway node. Therefore, the gene expression values, for all experiments which are available for this gene, are shown.

In many cases, the expression values of other genes in the pathway should be considered at the same time. This allows experts to analyze the influence of the expression regulation on the pathway. If, for example, one gene in the chain is severely down-regulated, the rest of the path may be influenced. The visual linking of the expression via the designated heat map view (*cf.*, Figure 6.17), is combined with direct on-node color mapping for a single, selected experiment. This strategy allows a user to see an overview of all expression values in the pathways for this one experiment and simultaneously see information on all experiments for the currently selected gene. By selecting another experiment, the color coding on the pathway nodes is updated. This way, all expression values for a pathway can be explored interactively.

For a pathway in focus, we do not color in the node, thus obscuring the caption, but rather use a colored frame and thereby preserve the visibility of the text. The usage of color in this fashion is made possible by using connection lines instead of color highlighting to show identity relations between views.

Nodes that encode several genes cannot be handled by a single on-node color. We therefore render such nodes in a different (false) color, signaling that there is not only one mapping value. The concrete values can then be explored by selecting the gene node and using the linked gene expression views. On-node mapping can be turned off when gene expression is not the focus of the analysis.

Use Cases Revisited

In the course of a requirement analysis with life scientists, and during feedback sessions with prototypes, the previously mentioned distinct workflows for the analysis of biomolecular data were discovered: a gene expression-centric and a pathway-centric approach, see Section 6.3.1.

The former case deals with a gene expression-centric analysis. A concrete example is illustrated in Figure 6.19. A biologist loads a set of experiments which have some pathological difference, for example half of the experiments are based on samples taken from subjects who suffer from diabetes, the other half is healthy. The biologist first filters the genes to exclude inconspicuous genes with the parallel coordinates. He then runs a clustering algorithm and explores the data using the heat map. Once he identifies some genes which clearly show differences between the two groups, he checks for pathways that contain several of these genes. He finds that many of the genes are in fact involved in several pathways related to the condition of the diabetic patients. To explore the role of the genes in the different pathways, he tells the system to load those pathways into the Bucket. If he actually finds a previously unknown indication of involvement, he could then proceed to verify his findings in a clinical experiment.

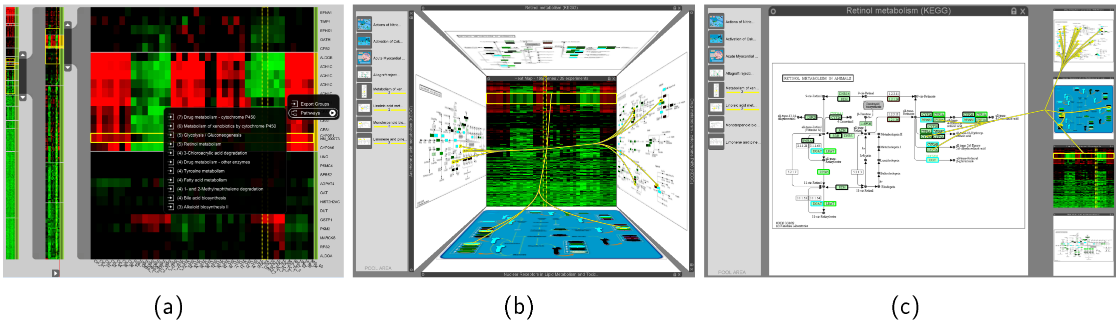

EFNA1
TIMP1
EPHX1
GATM
CPB2
ALDOB
ADH1C
CES1
CYP2E1
NM_000773
CYP2A6
UNG
PSMC4
SFRS2
AGPAT4
OAT
HIST2H2AC
DUT
GSTP1
PKM2
MARCKS
RPS2
ALDOA
Export Groups
Pathways
(7) Drug metabolism - cytochrome P450
(6) Metabolism of xenobiotics by cytochrome P450
(5) Glycolysis / Gluconeogenesis
(5) Retinol metabolism
(4) 3-Chloroacrylic acid degradation
(4) Drug metabolism - other enzymes
(4) Tyrosine metabolism
(4) Fatty acid metabolism
(4) 1- and 2-Methylnaphthalene degradation
(4) Bile acid biosynthesis
(3) Alkaloid biosynthesis II
Retinol metabolism (KEGG)
POOL AREA
(a)
(b)
(c)

Figure 6.19: Illustration of a gene expression-centric analysis. (a) After filtering out inconspicuous genes and running a clustering algorithm, the pathologist finds a cluster in the heat map which has strongly diverging expression patterns for the different conditions. He checks for pathways containing the genes in the group, and in fact, several metabolic pathways contain four or more genes of the cluster. By clicking the pathways they are loaded into the Bucket for exploration (b). There he finds that the different pathways are heavily connected. Looking more closely at one pathway and its genes (c), he finds that the genes of the gene family *CYP* show the differential expression pattern. After checking *PubMed* and *Entrez Gene* with the integrated browser, he learns that this family is a known catalyst for many reactions in the drug metabolism.

In the latter case, a user is interested in a particular pathway. He wants to understand the pathway itself, the function of the particular genes involved and whether the genes play a similar role in other pathways. He wants to know details about a specific gene, search for publications and look it up in one of the large databases like *Entrez Gene*. He may also be interested in the expression regulation values of the genes in the pathway for his experiments, in which case he is primarily interested in possible effects of the regulation on the pathway under investigation. He therefore starts his analysis by opening the Bucket and searching for the pathway he is interested in. Having loaded his gene expression data, he immediately sees the expression values for each selected experiment. He then notices a gene which has interesting properties – for example, the record in the *Entrez* database, which was automatically loaded in the linked browser, tells him that the gene is involved in many forms of cancer. By right-clicking the gene, the system presents all pathways that also contain the gene – and in fact, most of them are cancer-related. However, some are not, which grasps his interest. He now moves a seemingly unrelated pathway into the focus and explores the role of the gene in this pathway.

Summary

The Bucket setup realizes a restrictive approach for arranging 2D views in 3D scene. It is an effective way to visualize relations between different views and its data sets and thus, supports the informed analyst in terms of orientation. The Bucket can naturally accommodate focus+context as well as different levels of detail. It avoids confusion through a clear navigation concept, minimizes visual clutter with multi-level edge bundling and allows the user to manage many views conveniently.

6.3.4 Evaluation of the Bucket and Visual Links

We performed a user study to evaluate different aspects of the Bucket compared to traditional list-based pathway exploration methods normally used by biomedical experts. Our study specifically focused on the quality of the visualization methods to provide a useful context for finding target information in relation to the usage of both single-screen and multi-screen environments. The latter was taken into consideration, since [Yost et al., 2007] showed that multi-display setups can considerably advance the cognition and correlation process of information sources.

We chose not to compare Caleydo to other visualization frameworks, since the goal was to evaluate how much our novel visualization method can improve their previous workflows. A general comparison of visualization techniques such as the Bucket versus traditional multiple views should be conducted with a more general use case and average users, not life science experts. We also chose not to compare our system with other domain-specific software, since the functionality of the applications diverge in such a way that only trivial aspects could be compared.

Setup and Procedure

The physical setup for the evaluation consisted of a desktop computer with two displays connected. Users were presented with several comparable complex search tasks, simulating real-life use cases. Participants performed two different tasks resembling the workflows described in Section 6.3.1, under all four conditions: List-based and Bucket-based search tasks were performed in both single- and dual-monitor setups (see Figure 6.20). The first task involved pathway exploration, in which the participants were asked to detect relations between pathways, searching for a specific pathway and identify a specific gene in the pathway. As a next step, information about the gene in the *Entrez Gene* database had to be found. Finally, the participants were asked to find other pathways where the gene is also involved and determine whether there are other genes that those pathways share. The second task was based on gene expression analysis. Participants were asked to discover a specific pattern in the expression data using brushes in the parallel coordinates browser. The task required exploration of the pathways that contain these genes, and identifying a gene involved in a particular disease. During the first two conditions, the visual links were displayed in the Bucket view. The information on the second monitor (a web browser linking to gene databases for task one and a parallel coordinates browser for task two) was provided in a separate, tabbed window in conditions one and three. The task was subdivided into smaller units that were given step-by-step by the test supervisor. In order to simulate traditionally used list-based search methods (web interfaces like KEGG), we modified the application's user interface. It closely resembled the traditionally used list-based methods, as confirmed by our participants. It should be noted that the list condition actually had some enhancements over pure web interface methods, which would have been very hard to use for a comparative study in its original form.

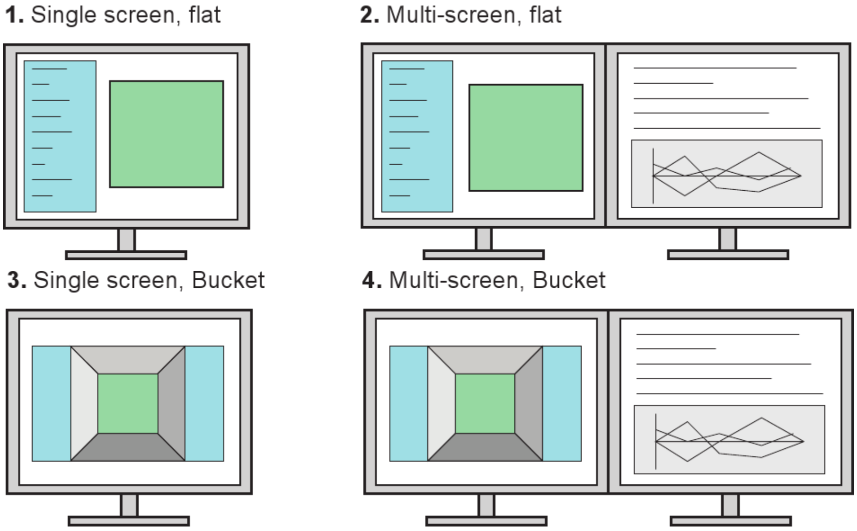


Figure 6.20: The four different setups for the user study.

We employed a 2x2 within-subjects factorial design with the factors view (Bucket, list) and display setup (single-monitor, multi-monitor). Analysis of main effects and interactions were performed at $\alpha = .05$ (see Table 6.1). Bonferroni adjustments were applied for post-hoc comparisons. To counterbalance the conditions, a Latin square distribution was used. All participants were videotaped with their consensus for later reference. The evaluation started with a ten-minute introductory session (including five minutes usage by the participant) in which the relevant functionality of the system was presented. After performing the different tests, participants answered a 7-point Likert scale questionnaire with 16 questions for both view levels and monitor-setups. Open discussions followed, where participants reflected on their experience. The total time of the user study was about 1h 15min per participant.

A third task focusing on pure observation was added with modified conditions to specifically investigate the utility of visual links: The three conditions were list-based, Bucket without visual links and Bucket with visual links, all on a single screen. Participants were asked to evaluate the quality and usefulness of the visual links under these conditions. This task was not performed in multi-screen conditions, since it is independent of the screen setup. We hypothesized the following outcomes:

H1 The Bucket performs better than the list-based mode.

H2 Multi-screen performs better than single screen, both in list-based and Bucket-mode.

H3 The visual links are a significant aid in the identification of relevant information.

For the evaluation, we recruited twelve participants with a background in life sciences. Eight participants (4 male, 4 female) were students (4 PhD, 4 master students) with beginner or intermediate experience, four participants (3 male, 1 female) were senior researchers and practitioners at a medical faculty.

Results

From the twelve original participants we included eleven in our analysis. One questionnaire was removed since it was highly inconsistent by itself and with respect to the interview. The results of the evaluation of the questionnaires is summarized in Figures 6.21, 6.22 and Table 6.1.

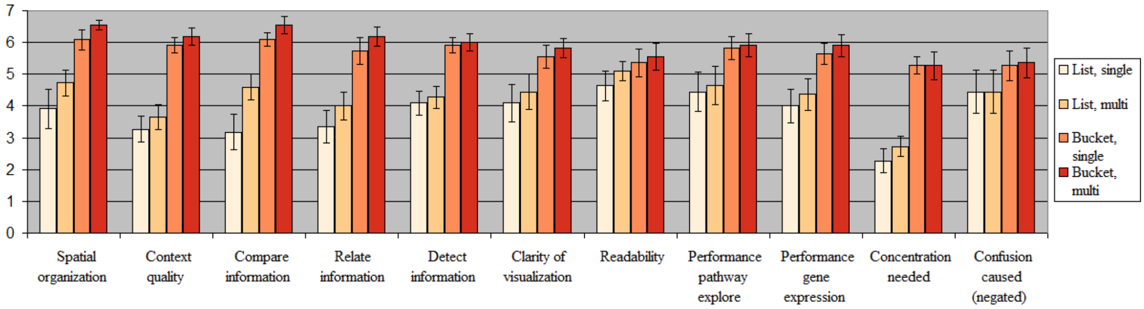
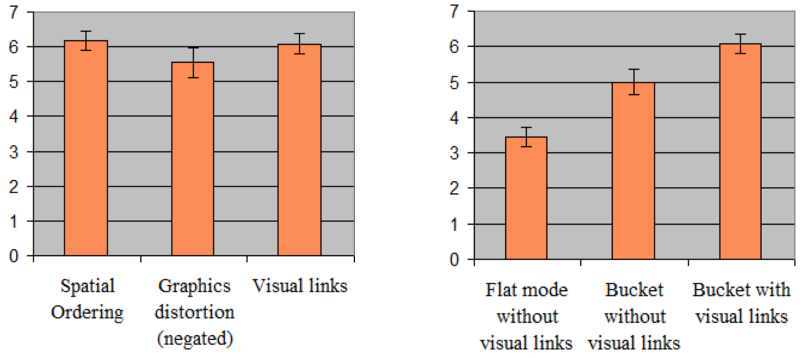


Figure 6.21: Questionnaire results comparing the four different tested conditions for eleven areas of interest, comparing the four different setups of the first two tasks (N=11).

Information Comparison The performed tasks can be characterized as directed searches that aimed at accomplishing a specific, predefined goal. Participants needed to



(a)

(b)

Figure 6.22: (a) Questionnaire results for three questions concerning the Bucket (N=11). (b) Comparison of perceived value of visual links compared to modes without visual links (N=11).

relate multiple sources of information, including different graph types and text-based sources. We found significant main effects of the view and display conditions on both the comparison of information and the quality of context, whereas the viewing condition also had a significant main effect on the detection of information. Additionally, an interaction between view and display was found for the information comparison. Relevant information was detected more easily in the Bucket conditions than in the list-based conditions, which was further improved by using the multi-monitor setup. Participants found the contextual information important for these tasks: The quality of contextual information was rated 'good' in Bucket conditions (in particular the multi-monitor

Table 6.1: Main effects and interactions of view and display conditions.
Significance: * = p<.05, ** = p<.001 (N=11)

Question	**view** **main** ($F_{1,10}$)	**display** **main** ($F_{1,10}$)	**interaction** **view*display** ($F_{1,10}$)
Spatial organization	24.444**	5.904*	1.379
Context quality	46.414**	6.806*	0.313
Compare information	50.975**	6.941*	5.213*
Relate information	30.414**	3.978	2.222
Detect info	14.912**	3.750	0.312
Clarity of visualization	4.290	6.806*	0.132
Readability	1.000	2.168	1.000
Perf. pathway explore	4.646	1.957	1.000
Perf. gene expression	10.542*	3.551	1.000
Concentration	27.121**	0.694	4.808
Confusion (negated)	2.560	1.000	1.000

condition), whereas the list-based multi-monitor was only rated ‘mediocre’. The latter is slightly surprising, since one can clearly compare at least two different information sources in a multi-monitor setup. As we noticed during the interviews, this rating can be traced back to the participants’ long experience of using just a single screen, which may be a learning problem. Overall, we found clear evidence that the detection of information is improved by the visualization aids offered in the Bucket, which performed significantly better than the list-based conditions.

Visualization Method The graphs analyzed in the tests are very dense: a large amount of information is compressed and screen space is limited. Obviously, the readability of the graphs is important to identify relevant information. The way the graphs are presented in the list and Bucket conditions is quite different, especially since graphics which are not in the center of the Bucket are distorted. When participants were asked about the readability of graphics in the different conditions, no significant difference was found: Bucket views even performed a little better on average. The graphics distortion was rated as negligible by most participants. This is quite surprising, since the graphs are clearly distorted at the side panels of the Bucket. In the interviews, some participants stated they would simply put those graphs needed for the analysis into the center of the Bucket. Some participants also said that distortion was not a problem since they can easily flatten the Bucket to a 2D view, removing perspective distortion of the side panel information (see Figure 6.17). In the interviews all participants stated they prefer the Bucket for finding interdependencies over the flat, zoomed mode.

The visual links were rated ‘very useful’ and also believed to speed up search tasks. During the interviews, many participants stated that the visual links aided the search for relevant information considerably. Although visual links do not affect the results of the analysis, it was easier to perceive the entire scene with its selections. In addition, some participants noted that visual links clearly helped them focus on specific parts of the graphs. We observed some participants consciously following the visual links from point to point to detect relevant information. Some also noted that the visual links are especially helpful with the pathway views, and considered them of less importance in the heat map: They argued that the gene expression views highlight the selections well by themselves, whereas the complex textures of pathways benefit from the additional visual clues.

Effectiveness and Complexity Our main goal is to improve the workflow of users exploring pathways and gene expressions. The participants supported the hypothesis that the Bucket improves the workflow (speed and accuracy) significantly in comparison to the traditional list-based methods they are used to. Specifically for the gene expression task, we noted a significant main effect of the view mode (Bucket) on the perceived performance of the task. Participants noted that less concentration is required during the search task using the Bucket, which is in line with the ratings from the previous sections: The view condition had a significant effect on the level of concentration. Participants also were less confused in the Bucket conditions, even in comparison to the multi-monitor list condition. Likewise in single-monitor condition, the Bucket was rated better than the multi-monitor list-based condition in all questions.

Discussion

The evaluation clearly shows that the Bucket is a valuable improvement for pathway exploration over current practice using list-based methods. The Bucket performs significantly better in most conditions for most of the participants (supporting *H1*): in 7 out of 11 questions, we noticed a significant effect of the view condition on the outcome, and the average rating was higher for the Bucket conditions without exception. Three participants were even unable to fulfill the proposed task in the first list-based condition they obtained. Only a single user stated that the list-based method was preferred over the Bucket.

The visual links were very well appreciated, and clearly improve the search task performance in terms of (subjective) speed and lower cognitive load (supporting *H3*). The preference of single-monitor conditions may be related to the lack of experience our users have with multi-monitor configurations. One participant even failed to notice content on the second display entirely. These observations stand in contrast to previous evaluations like [Yost et al., 2007] that reported considerable performance boosts in multi-monitor environments. Thus, *H2* turned out to be false. However, the (informally) observed performance of our participants was clearly better in the multi-monitor setup.

The evaluation shows that the proposed Bucket arrangement is preferred over traditional list-based methods, especially in the areas of *context quality* and *spatial organization*. Participants stated that the required concentration was lower when using the Bucket. It was found that visual links significantly improve the ability to search for information.

6.3.5 Gaze-Based Interaction for Supporting Visual Analysis

Aside from the traditional ways of interacting with visualization systems, such as keyboard and mouse, the direction of a user's gaze can be measured and in turn be incorporated as an additional input resource for providing extended visual aid. In the following section, the tracking information is not employed for selecting data (*i.e.*, mouse interaction) but for an intelligent adaption of 2D and 3D visualization techniques. Derived from the focus+context paradigm, this is called *gaze-focus*. The proposed methods are demonstrated for supporting the interaction with different visualizations: a 2D heat map as well as parallel coordinates; and the 3D Bucket. The content of this section draws upon the material published in [Streit et al., 2009b].

Gaze-based interaction is not a standard user interaction technique, primarily because of the high costs of the required systems. However, this is currently changing, as eye tracking becomes technically possible with low cost equipment [Hiley et al., 2006]. [Fono and Vertegaal, 2005] have shown that gaze-based interaction with windows is preferable over traditional input techniques. They use gaze tracking to select and zoom windows a user is focusing on and also to zoom in a digital media application. They show that task completion time was faster and preferred by users with gaze tracking compared to traditional input devices. However, gaze-based systems suffer from some inherent deficiencies. One problem is the involuntary selection of items (the Midas Touch Effect), which can be overcome to some degree by, for example, selecting only after a fixation has lasted for a certain time (dwell time) or after a manual click [Vertegaal, 2008]. Low cost eye tracking systems have a maximal precision about 1 centimeter on the screen [Hiley et al., 2006], while high-quality systems are more robust and accurate. However, the accuracy of such systems is naturally limited by the inability of the human vision system to exactly focus on a particular spot [Tobii Technology, 2010]. [Ashmore et al., 2005] try to overcome this by magnifying the focused region with a fish-eye lens and then selecting the target within the magnification.

For the prototype implementation the Caleydo system was connected to a professional, monitor-based eye tracking system from SMI[6]. This costly solution comes with a high accuracy which results in an excellent user experience. However, the gaze-focus does not require the accuracy of professional eye tracking systems; instead, a low cost, webcam-based eye tracking module is sufficient for achieving good results.

In the following, it will be differentiated between gaze-based interaction within a single view and interaction that targets handling multiple linked views in the Bucket setup.

Single View Gaze Interaction

Parallel coordinates are well suited to visualize thousands of data points simultaneously over a limited number of dimensions. With an increasing number of dimensions however, details are lost due to the reduced spacing of axes. The handling of truly large amounts of dimensions need special dimensionality reduction approaches, *e.g.*, [Yang et al., 2003]. By using orthogonal stretching, the number of simultaneously perceivable dimensions can be increased to a certain extent. The spacing between the axes and therefore also the readability of interesting regions can be increased. These properties lend themselves perfectly to gaze-based scene manipulation. Since only a small region is observed sharply in the fixation phases of the eye, this region can be enlarged once a user looks at it. The spacing of the other axes is reduced, thereby using less screen real estate while still providing the contextual information.

Caleydo's hierarchical heat map presented in Section 5.3.1 is the second visualization technique for which the usefulness of the gaze interaction is demonstrated. Again, the gaze-input is employed for manipulating the visualization according to the user's gaze. The hierarchical heat map is composed of three levels: the first level provides an overview, while the second and third level allows the analyst to further drill down in the data. Interacting with the densely visualized content is supported by providing more space to the particular level currently focused on by a user.

[6] `http://www.smivision.com`

Multi-View Gaze Interaction

State-of-the-art multiple view systems arrange views side by side on the screen. [Fono and Vertegaal, 2005] show how to zoom in on an application window, which can also be applied to multiple view applications. However, this method is naturally limited by the available screen space and can therefore only be used for a very low number of views. The introduced Bucket approach (see Section 6.3.3) overcomes this problem by using a 2.5D arrangement to manage up to about 20 related views.

While the multi-level approach (Bucket bottom, walls and rim) enables the management and handling of numerous views, it introduces a distortion problem. Especially text is difficult to read when rendered on the Bucket walls. Therefore, the static Bucket setup was extended to a "rubber" bucket by taking the user's gaze into account. In contrast to the single-view implementation that is based on the eye tracking input, in this case the user's head movement was used for manipulating the scene. The reason for this lies in the unintentional feedback effects on the user's eye movements that the scene changes in the Bucket setup would cause. The prototype implementation uses an off-the-shelf Nintendo Wii Remote mounted on the user's head. The user's head movements were captured by a Wii sensor bar mounted onto the monitor. This low-cost solution was inspired by Johnny Chung Lee's VR Desktop Head Tracking approach [Lee, 2008].

The Bucket is rotated according to the user's head movements (see Figure 6.23), reduc-

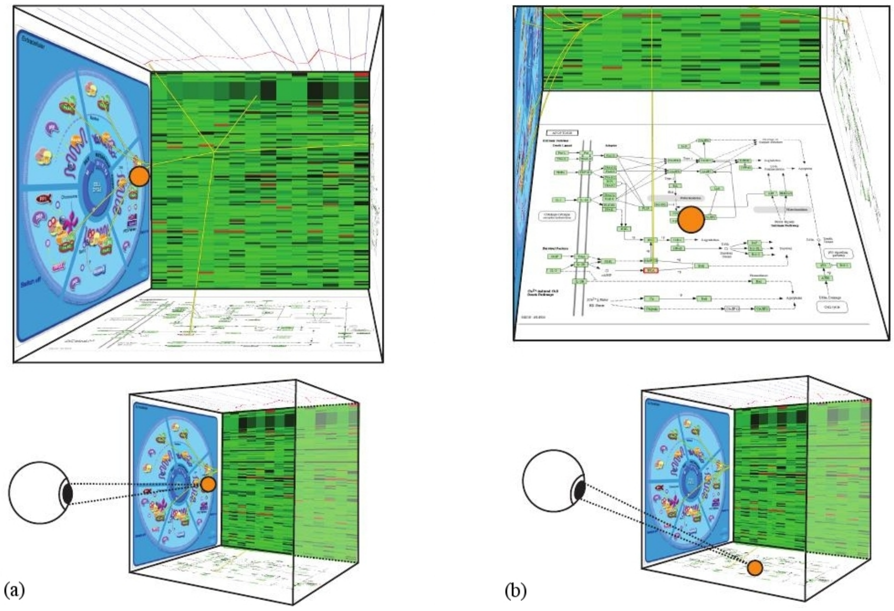


Figure 6.23: Gaze-based interaction with the Bucket. The 2.5D representation is adapted according to the user's focus point (orange). In (a) the user gazes at the left bucket wall while in (b) the user looks at the lower wall.

ing the distortion of the view being looked at. When the user moves his head towards the screen, the focused view in the 3D scene is transformed to the user's direction (diving into the Bucket). In addition, the gaze navigation in the 2.5D representation in combination with the visual links immerses the user into the scene.

In summary, the utilization of gaze input for manipulating 2D and 3D views is valuable for aiding users when interacting with information visualization systems.

6.4 Summary

This chapter introduced a series of visualization techniques that aim at providing orientation support on the level of individual data items (*S1.1*). The prior objective of these techniques is to make the user aware of the interconnections of data items within a single data set but also show their occurrence in related data sets. All of the techniques introduced depend on visual links, a very explicit and expressive way to convey the data dependencies to the analyst.

An analysis system's ability to communicate data relations on this fine grained level of individual data items is an essential prerequisite for orientation support in a complex analysis scenario that includes a broader information landscape. In such cases, a user needs to be additionally oriented on a higher level, where the dependencies between the data sets are presented in a more abstract way. This subject is addressed by the Stack'n'flip system, introduced in Chapter 8.

Chapter 7

Orientation Support Across Applications, Analysts and Displays

Contents

The previous chapter presented techniques which aim to establish the conditions required for orienting users within the information landscape (*S1.1*) in traditional single application, single user analysis scenarios. Although this is a challenge by itself, this chapter introduces two additional aspects in terms of the setup characteristics:

- **application-spanning analysis**, as one single application might not be sufficient for fulfilling all analysis needs, and
- **collaborative analysis**, as one single analyst often has not got the expertise and background knowledge to perform a comprehensive analysis alone.

With the rising demand to incorporate data from different sources and of various types for addressing increasingly complex research questions, both of these aspects will gain in importance. In the following, it will be discussed how an incorporation of these additional variables influences the requirements of visual analysis with respect to orientation. The developments in this direction made in the course of this dissertation are also outlined.

7.1 Support Across Applications

Modern information workers who need to carry out everyday tasks with the help of a computer are confronted with a wide spectrum of different applications, *e.g.*, document readers, mail programs, office software, *etc.* – each of them fulfilling a different set of purposes. Only in rare cases can the whole task be accomplished using just a single application. Often users need to relate information across the boundary of a single application. This is also valid for complex visual analysis tasks, which require the concurrent consideration of heterogeneous data of various types. In many cases, software is highly specialized to a particular kind of data. However, it is unfeasible that a single ”super application“ exists or can be created that covers all of the user's needs. Hence, real world analysis problems will require a combination of existing applications.

However, individual applications are not integrated, making it hard to relate, evaluate or compare information across applications. This manual information matching is error-prone and time-consuming. The requirements in terms of orientation are similar to the multiple view problem in visualization frameworks. The system needs to make the user aware of the relations between views and the data. The standard approach of multiple coordinated view systems employs synchronized highlighting for communicating relations between pieces of information. This only works within a single application, where either each visual representation can access the same data storage or an event mechanism handles the synchronization. However, neither of these applies to independent applications.

The multi-application scenario can be described by the theoretical model as well, because it is irrelevant which application contributes the visual and computational interfaces.

7.1.1 Concept of Visual Links Across Applications

Little research has been done that aims to bridge the gaps between existing applications. The Snap-Together system can be considered as pioneer work in that direction [North and Shneiderman, 2000]. The system offers a light-weight API over which visualization systems can access a common database. This unique point of access allows the system to coordinate a series of operations (load, select, synchronized scrolling, *etc.*) among visualization applications.

Following the spirit of Snap-Together, an approach is proposed that follows similar goals but differs in fundamental aspects. First, it does not force applications to depend on a centralized data management, as this would require profound changes in existing applications. Secondly, instead of pure highlighting of related information, the proposed method uses the more explicit visual links for guiding the user's attention between multiple regions of interest arbitrarily scattered over the display.

In the previous chapter, the value of visual links for composed views, as used in the Jukebox and the Bucket, was discussed. In general, the arguments why visual links should be used as additional and particularly strong visual cues (*cf.*, Section 3.3.1) remain valid for the application spanning scenarios as well. Based on the preliminary work in this direction, the next section introduces the technique of visual links across applications. Manuela Waldner and Werner Puff played the leading role in the development of these ideas and their realization. The initial idea originated out of the domain requirements in the Caleydo project and is based upon the preliminary work on visual links from the Bucket. The technique was published as a full paper at the *Graphics Interface* conference, see [Waldner et al., 2010], and has won the best student paper award.

7.1.2 Basic Workflow and Architecture

The proposed system consists of two major components, as depicted in Figure 7.1.

The **Visual Links Manager** is a lightweight application running as a daemon service in the background. It handles the communication between the applications by providing a slim Remote Procedure Call (RPC) interface. In an initial step, the client applications need to register. The synchronization of selected pieces of information is based on the exchange of ID-Strings. Depending on the application, various ways exist to trigger a selection. Examples are marking a word, selecting an element in a chart, or entering a text in a search field. When the user triggers a selection in the source application, an ID is sent to the manager, which forwards it to all registered client applications. Each application then evaluates the ID individually by searching for it in its currently loaded data. For all positive matches, the bounding rectangle is reported back to the manager, which in turn forwards the collected regions to the second component, the Visual Links Renderer.

The **Visual Links Renderer** then processes the list of regions, calculates the routing of the visual links and renders the connection lines as a desktop overlay. In the sample implementation this component is realized as a plugin of Compiz[1], an OpenGL compositing window manager for Linux. Figure 7.2 shows an example where a user selects a piece of information in application A which is then localized and in turn linked to B, C and D. In order to reduce visual clutter, the connection lines are bundled per application, see C and D. The visual links are rendered as semi-transparent Bézier surfaces. In case the selection ID is in a currently invisible part of a window (*i.e.*, scrolled away), arrows at the windows border indicate this to the user, *cf.*, B and D. The arrow's length encodes the number of hidden selections. In cases where the links obscure important information, the rendering of visual links can be turned off on-demand via a keyboard shortcut. In addition, the user can choose to fade out (continuously decreasing the alpha value) the visual links after a predefined time.

[1] `http://www.compiz.org`

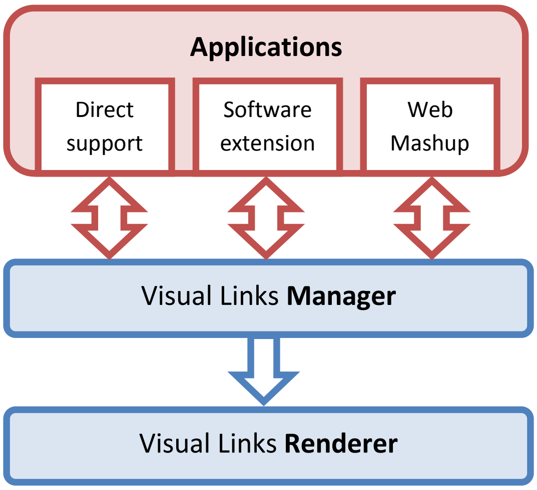


Figure 7.1: Basic architecture of the system's components. The Visual Links Manager realizes the synchronization of selections. The Visual Links Renderer calculates the bundled visual links and draws them on top of the desktop content.

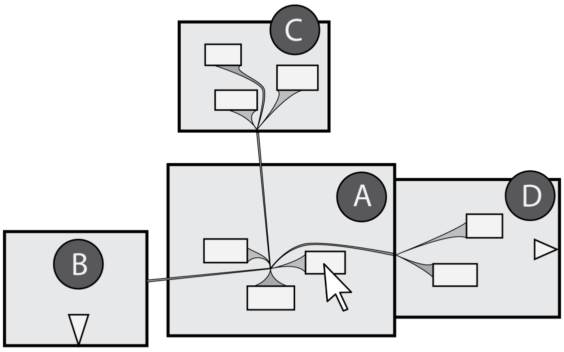


Figure 7.2: Sample of visual links connecting related information across applications A-D. The user triggers the selection in A. In turn, the system connects the occurrences found with the corresponding regions in B, C and D. Arrows indicate hidden parts of a window where the information was found as well.

The system offers three ways to attach existing applications to the linking mechanism:

- Option 1: **Direct Support**
  The application implements an interface class from the manager. This requires full access to the application's source code.
- Option 2: **Web Mashup**
  A mashup application runs in the browser and combines an existing web-service API with the manager interface. An example is the Google maps API which makes it possible to query locations and load the respective map segment.
- Option 3: **Software Extension**
  Many applications provide an extension mechanism for accessing the application's data in a minimally invasive fashion (without touching the actual application's core).

While the latter two approaches are limited to text parsing and string comparison, the first option is the most flexible kind of integration. The application can handle the selection of data with its own specialized interaction techniques. A visualization software, for instance, can provide a mouse-over feature for selecting a region in a scatterplot. The application can then translate the selected visual element to the ID which is then transferred to the manager.

Figure 7.3 presents the visual linking mechanism applied to a biomedical use case. The setup connects information between the Caleydo visualization framework (realizing integration Option 1) and the Firefox web browser (Option 3).

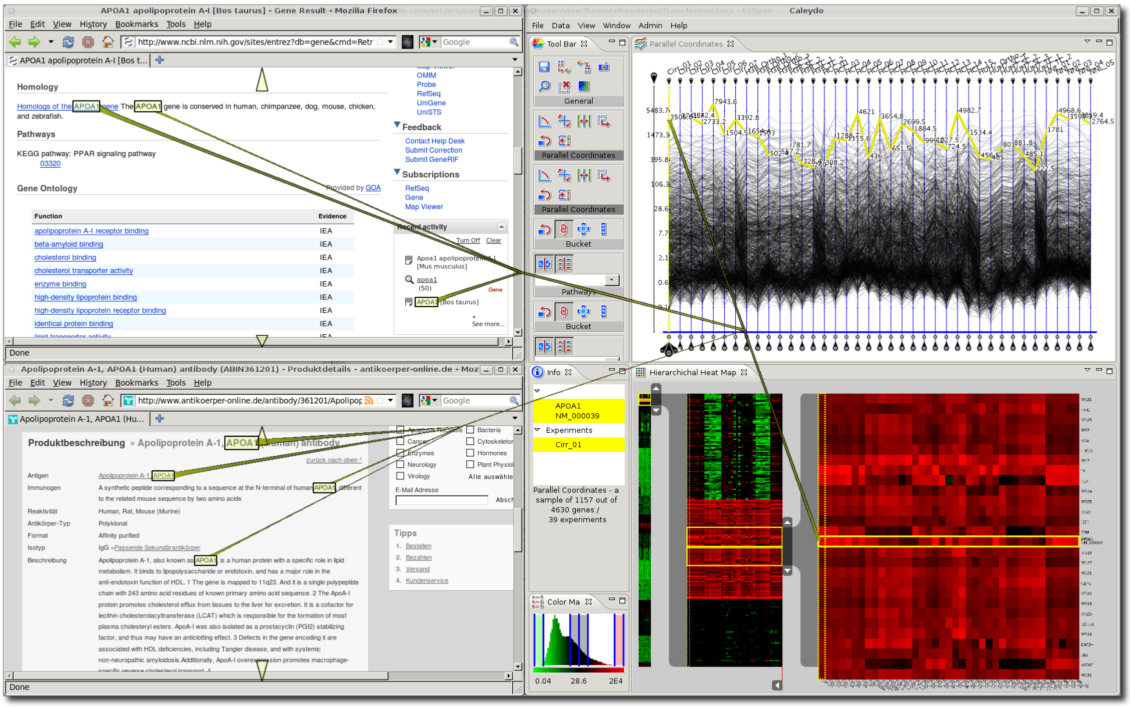

APOA1 apolipoprotein A-I [Bos taurus] - Gene Result - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.ncbi.nlm.nih.gov/sites/entrez?db=gene&cmd=Retr
Homology
Homologs of the APOA1 gene The APOA1 gene is conserved in human, chimpanzee, dog, mouse, chicken, and zebrafish.
Pathways
KEGG pathway: PPAR signaling pathway 03320
Gene Ontology
Function
Evidence
apolipoprotein A-I receptor binding
beta-amyloid binding
cholesterol binding
cholesterol transporter activity
enzyme binding
high-density lipoprotein binding
high-density lipoprotein receptor binding
identical protein binding
Feedback
Subscriptions
Recent activity
Apolipoprotein A-1, APOA1 (Human) antibody (ABIN361201) - Produktdetails - antikoerper-online.de - Mozilla Firefox
http://www.antikoerper-online.de/antibody/361201/Apolipop
Produktbeschreibung » Apolipoprotein A-1, APOA1 (Human) antibody
Antigen
Immunogen
Reaktivität
Antikörper-Typ
Format
Isotyp
Beschreibung
Apolipoprotein A-1, also known as APOA1 is a human protein with a specific role in lipid metabolism.
Caleydo
File Data View Window Admin Help
Tool Bar
Parallel Coordinates
General
Bucket
Pathways
Info
APOA1
NM_000039
Experiments
Cirr_01
Parallel Coordinates - a sample of 1157 out of 4630 genes / 39 experiments
Color Ma
0.04 28.6 2E4
Hierarchical Heat Map
Done

Figure 7.3: Visual links across distinct applications. A user browses online databases for interesting disease genes and selects a particularly interesting one. The Firefox browser plugin (realizing Option 3) feeds the Gene's ID to the manager interface, causing the system to propagate it to the Caleydo application (Option 1). Caleydo queries the gene ID in its internal data structure and highlights the corresponding polyline in the parallel coordinates as well as the row in the heat map, respectively. Analogously, it is also possible to trigger the visual links by selecting any data item in Caleydo.

The usability and user acceptance of the visual links across applications technique has been informally evaluated. The evaluation included seven participants between the age of 25 and 39. The subjects had to perform certain tasks where they had to relate information between various applications – in a setup which comprised combinations of the three integration options. According to observations and qualitative feedback, subjects easily understood the concept without instructions or prior training. All agreed that visual links are a valuable extension for information intensive tasks spanning multiple applications.

7.1.3 Summary

This section presented visual links across applications – a technique that allows users to connect pieces of information between unconnected applications. Three options were proposed for integrating existing applications to the linking mechanism. The technique was demonstrated by means of a biomedical analysis scenario. Further use cases, details on the evaluation as well as on the implementation can be found in the full paper [Waldner et al., 2010].

An interesting extension for increasing the techniques' flexibility would be to add an OCR (optical character recognition)-based alternative for applications where none of the three existing integration options are applicable. In this case, the ID-String could be located via image processing without the need to communicate with the application that provides the content.

While the current mechanism is limited to strings, in a next step a wider set of information could be exchanged between the applications. In principle, it would also be possible to synchronize brushing operations between independent applications and therefore facilitate more complex analysis tasks.

7.2 Support Across Analysts

The strategy of bringing together experts from various fields in order to jointly address complex problems is well established in the field of biology, see [Pennisi, 2005], but also already common in various other domains. Each of these experts has a specific perspective on the data, pays attention to different details, and reasons along the lines of his/her own particular domain. Organizing this multifaceted interplay between large amounts of complex data, multiple domain experts from different areas, and the laborious back and forth between exploration and confirmation of the analysis process is a challenging task. This task is what collaborative environments have set out to support and to advance, as the results that can be gained from an interdisciplinary, collaborative data analysis outweigh technical problems. One essential problem is interaction with the complex, heterogeneous data spaces in these environments. Due to the multidisciplinarity, data is again available in various forms (free text, images, statistical tables, *etc.*), in various representations (tabular, tag clouds, visualizations, *etc.*), and on multiple levels of detail, each of which is meaningful to at least one of the participating domain experts. To allow fruitful collaboration, all of them need to be integrated into one seamless, interactive analysis process.

Although the value of a well defined analysis setup model has been demonstrated by means of various single user setups and techniques, the concept presented can be scaled up to also cover multi-user scenarios. In order to demonstrate the concept's expandability, the biomedical problem area will be looked at again and a concrete set of roles will be introduced. Due to the merged competence and knowledge of these experts, it is possible to collaboratively address complex research questions, which might be overwhelming for a single user. This work is based on the position paper published in [Streit et al., 2009c]. Note that unlike to novel visualization techniques from the previous chapter and the linking across applications, the work on collaboration is not a solution ready to be used, but more a conceptual discussion of next steps to be taken in this direction and how the unified representation of the model can help in this context.

7.2.1 Sample Multi-User Scenario

Domain experts from different fields come together to collaboratively analyze their respective data to make a joint decision on a patient's diagnosis and further treatment plans. In detail, the roles involved and the data these experts focus on are:

- the **oncologist:** CT/MR/X-ray scan of the tumor, treatment history
- the **pathologist:** tissue samples of the tumor biopsy
- the **geneticist:** data on the genome-wide regulation of the genes
- the **biologist:** gene regulation in the context of pathways

Figure 7.4 uses the data model defined in Section 4.6 as a basis and shows the set of data sources each person is able to cover, given his/her expertise and background.

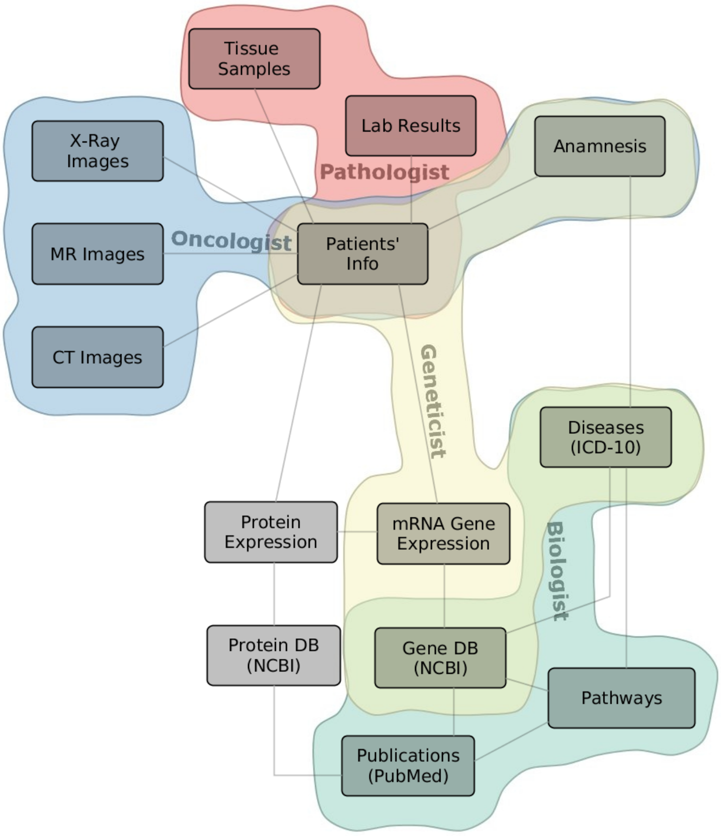


Figure 7.4: Interactive Bubble Set visualization showing which domain experts are able to analyze which data sets. The information is augmented on top of the data model from Section 4.6. This example is generated by using an adapted version of the Bubble Set code supplied with the original paper [Collins et al., 2009].

This additional knowledge on top of the analysis setup model can be employed for answering the following questions:

- **Which experts can work together?**
  An overlap of expertise (*i.e.*, data sets to which multiple roles are assigned) indicate

the interfaces between them, where they potentially meet during the analysis and need to talk to each other. In the example from Figure 7.4 the patients' basic information is such a bridging data set where all roles (except the biologist) are involved. This is in line with an analysis goal that is centered around the treatment of a certain patient.

Having this knowledge at hand makes it possible to support the collaboration process by adapting the infrastructure to actively support these bridging interfaces, as discussed in the following Section 7.2.2.

- **Where are the gaps?**
  Data sets that none of the available experts cover can lead to situations where specific sub-tasks or even a whole analysis towards a given goal becomes impossible. Thus, this knowledge is valuable during the planning stage of an analysis, *i.e.*, making sure that all experts required for reaching an analysis goal will be present.

Note that although each expert has his/her core field of expertise, they often also have profound knowledge in related domains. Therefore, the assignment of users to data is often not binary but rather fuzzy, and this needs to be considered when designing such a multi-user scenario.

So far, we have discussed the potential of collaborative analysis for visual analytics problems and have shown how to define the roles on top of the analysis setup model and how to employ this extra knowledge. However, the inclusion of multiple experts also influences the infrastructure needed in terms of hardware that facilitates collaboration.

7.2.2 Co-Located Multi-User Infrastructure

The field of Computer Supported Cooperative Work (CSCW) categorizes setups according to time and space. The CSCW matrix divides space into four categories, *cf.*, [Baecker et al., 1995, Johansen et al., 1988]: asynchronously vs. synchronous collaboration; and co-located vs. distributed. Although asynchronous as well as distributed scenarios are challenging research fields with a lot of potential for visual analytics applications, the focus here is on co-located collaboration with experts running an analysis concurrently.

In collaborative information seeking, as it is often understood nowadays, only one user is actively performing the interaction, while the colleagues just participate as observers. The example in Figure 7.5(a) shows a case where only the user on the right has access to mouse and keyboard. The co-located collaboration is sometimes even detached, meaning that each user is doing the data exploration on his/her workstation independently. Afterwards the results are discussed together and finally merged into one common outcome or hypothesis. Such a working style can be referred as the traditional collaboration approach. However, in order to create and effectively use such a collaborative information workspace, it is vital to understand the processes involved, and particularly the differences to single analyst scenarios. Established, high-level interaction patterns work well for single user, single data source scenarios. However, they cannot simply be applied to the collaborative analysis of heterogeneous data.

In general, one can talk about three distinct cases:

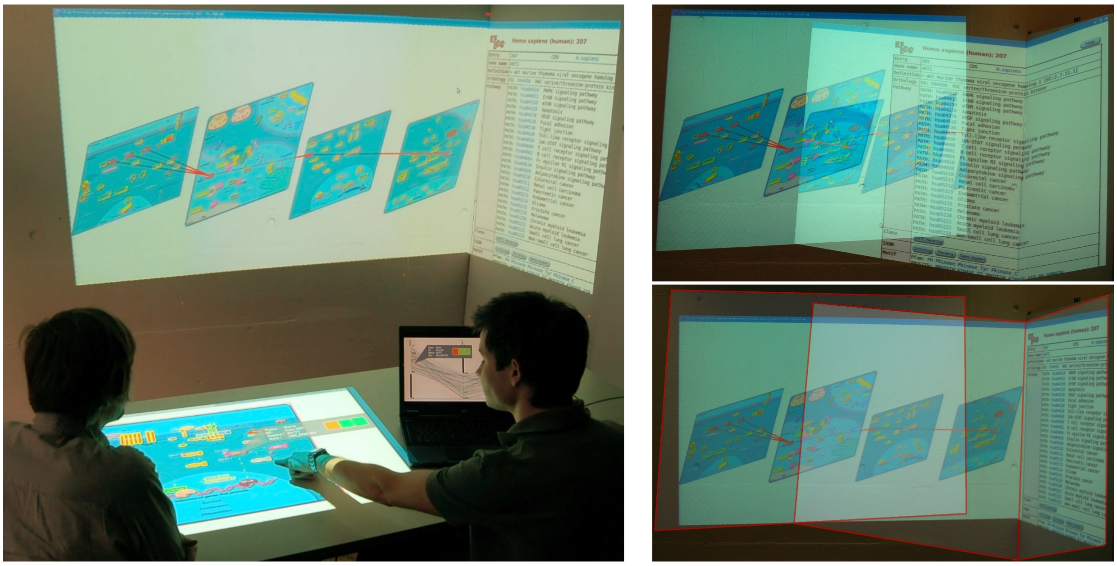
- The **single-user** case, addressed by the techniques presented in the previous chapter. The analysis takes place in a linked multiple view application on a single output device.
- The **static multi-user** case, which is targeted by the adaptation of a visual analysis framework to run in a multi-display environment. The next section introduces such an environment called Caleydoplex. It provides a fixed set of displays and projection areas to facilitate multi-user interaction.
- The **dynamic multi-user** case, where the configuration of the users and devices involved are not static, but changes over time. In such "smart environments“, the device ensemble of available displays is changing, as users connect and disconnect their brought devices (laptops, tablets, smartphones, *etc.*) with the environment during runtime. A detailed discussion of this case realization and its usage for a medical scenario is given in [Thiede et al., 2009].

It can be observed that with each of these cases the complexity of coordinating multiple data sets to be shown on multiple displays for multiple users increases. The challenges this poses are abundant and range from the distribution of the data to the available display devices (or views in the single-user case) to the assurance that privacy concerns are met. The three-stage model driven design concept provides a conceptual and concrete way to model all these complex dependencies and to derive solution approaches that achieve real seamless collaborative data analysis in such multi-analyst scenarios.

7.2.3 Caleydoplex Prototype Setup

Caleydoplex is a prototype setup of a collaborative information workspace built at the Graz University of Technology. It aims to facilitate multiple projection areas and devices that enable experts from different domains to jointly perform a data analysis. Figure 7.5(a) shows an initial prototype where two users run a joint analysis using the Jukebox setup (Section 6.2.2), *cf.*, [Waldner et al., 2008]. As mentioned earlier, in this scenario only one user can control the interaction devices. In the full Caleydoplex setup, however, several analysts are involved, each having access to his own keyboard and mouse where he can interact with a private display in front of him as well as with the public displays that are projected onto the walls. Figure 7.6(a) illustrates this setup as a sketch, the realized prototype is shown in Figure 7.6(b) [Waldner et al., 2009].

Caleydoplex uses a distributed version of Caleydo and runs it in the Deskotheque multi-display environment [Pirchheim et al., 2009]. The Deskotheque infrastructure employs a camera-assisted offline calibration which creates a 3D model of the physical setup. On

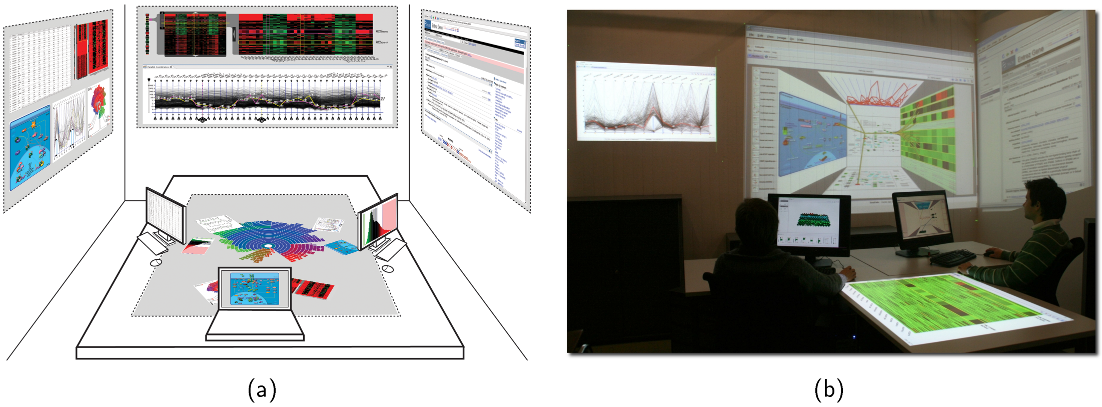


(a) (b)

Figure 7.5: Prototype setup of a multi-display environment including a wall and a table-top projection where two users are running an analysis using the Jukebox (*cf.*, Section 6.2.2). The stack of interconnected pathways is projected onto the wall in front of the users, an on-site web browser onto the wall on the right, the pathway of current interest onto the table and the mapped gene expression is visualized in a parallel coordinates view on the laptop screen. In (a) two users are working with the setup, however, only one has access to the mouse and keyboard. (b) shows the two-projector wall setup without corrective measures (top) and with geometric compensation for the projection around the corner as well as the blending of the overlapping region (bottom).

this basis the system then provides functionality such as geometric compensation that makes projecting onto non-planar surfaces possible, as well as blending of overlapping projections and mouse-pointer warping, see Figure 7.5(b). Similar setups have been described for different application contexts, *e.g.*, for an office environment called *The office of the future* [Raskar et al., 1998] or for an entertainment scenario called *Smart Living Room* [ao and Kirste, 2005]. However, Caleydoplex is specifically targeted at exploratory visual analysis.

Private & Public Displays Caleydoplex, as well as other collaborative information workspaces, differentiates between private and public displays. In the simplest case, each domain expert displays his/her domain data on a private display – *e.g.*, in Figure 7.6(a) three users from different domains are sitting around a table, each with a private view on a single monitor. Besides the plain distribution of views, the users' roles can further be facilitated to provide tailored visualizations, as a user's working domain influences the visualization technique chosen and the terminology used for annotation purposes. Different domains can then be bridged either by a simple coordination of visualizations between the (private) displays or by the combination of data from different sources in public visu-



(a)

(b)

Figure 7.6: Caleydoplex multi-user, multi-display setup for collaborative information visualization at Graz University of Technology [Waldner et al., 2009]. While (a) illustrates the setup as a sketch, (b) shows the prototype setup in action. Each user has a private display where views and labels are adjusted according to the user's profile. Views, data selections and filter operations can be published on the public displays and synchronized in the whole analysis workspace.

alizations. Public displays, *i.e.*, projection walls which are visible for multiple users, can host these integrative visualizations. This also allows multiple users to work on the same task. Therefore, public displays serve as spaces designated to bridge various domains and in turn, also knowledge gaps.

The physical separation between public and private displays can also be used to circumvent privacy issues, by showing sensitive data only on private displays. In a clinical scenario, the biologist may not be allowed to see the clinical history of patients for privacy reasons. The control over the individual displays enables the collaborative environment to grant or deny access to experts depending on their role, either allowing them to roam freely within all available data sources or just within the absolutely necessary parts. Even annotations could differ, providing patient details in private views, but being anonymized in the public views. The anonymization does not affect the linking of the individual views. Thus, selections and other interactions are reflected throughout the whole ensemble of displays.

7.3 Collaborative Information Linking

The first part of this chapter presented the work of visual links for relating information scattered over remote display locations hosted by distinct applications; the second part stressed the potential of collaborative work for the visual analysis and presented first results in this direction. This chapter concludes by outlining the idea of combining the strength of both aspects – resulting in *Collaborative Information Linking* [Waldner and Schmalstieg, 2011]. While this work was pursued by Manuela Waldner, it is briefly summarized here because it emerged from the preliminary joint work and demonstrates how the ideas have inspired further developments.

In principle, the approach is identical to visual linking across applications for a single user, as introduced in Section 7.1.1. However, in the collaborative scenario multiple users operate on a shared workspace. Because a different color is assigned to each individual user, the mouse pointers and visual links are clearly distinguishable. This practice scales well to a large number of users, as the only limiting factor is the additional visual clutter from the links themselves.

The proposed system runs on a large, tiled multi-projector setup which serves as a single high-resolution display. Multi mouse-pointer support allows users to concurrently interact with applications on the shared workspace. Usually, a special purpose groupware solution is run in such setups. However, as suggested in [Lauwers and Lantz, 1990], the proposed system does not tie users to a groupware solution but rather allows them to work with their well-known, standard applications. Consequently, only the windowing system takes care of the collaboration support. The systems infrastructure is an extended version of the single user solution (Section 7.1.1).

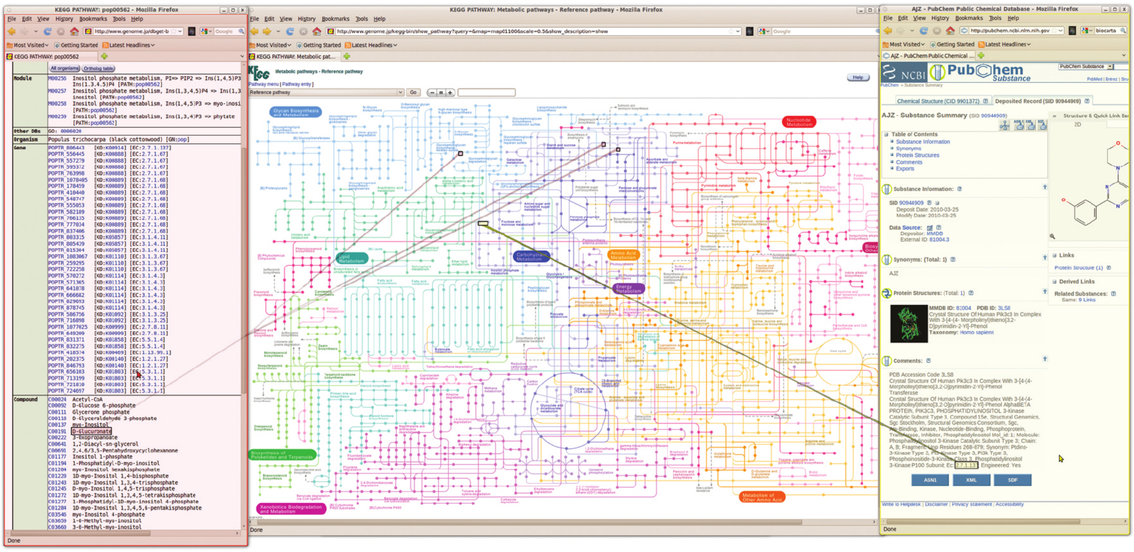

KEGG PATHWAY: pop00562 - Mozilla Firefox
KEGG PATHWAY: Metabolic pathways - Reference pathway - Mozilla Firefox
AJZ - PubChem Public Chemical Database - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Most Visited
Getting Started
Latest Headlines
Metabolic pathways - Reference pathway
Help
AJZ - Substance Summary
Done

Figure 7.7: Collaborative information linking. Multiple users working on a shared, high-resolution workspace. Selected pieces of information are simultaneously visualized by rendering user-specific colored sets of visual links across independent application windows [Waldner and Schmalstieg, 2011].

The suggested solution again realizes the concept of private and public application windows: private windows for conducting individual information retrieval; and shared application windows for joint verification and discussion. In large, high-resolution display setups users tend to establish personal territories [Tse et al., 2004] where they arrange applications for private, independent work. The visual linking solution is also aware of the applications' privacy status. A locking mechanism guarantees that visual links triggered by collaborators do not distract the users while they are interacting with private application windows. However, content from shared application windows is always linked if a selection string is found. Figure 7.7 depicts a collaborative analysis session where two users trigger independent selections. User-specific, color-coded visual links connect the matches in the shared application window.

The collaborative visual link solution also provides a bookmarking mechanism that records selections from all users and therefore also facilitates the management and sharing of findings.

7.4 Summary

This chapter started by discussing the necessity of including unconnected applications in the visual analysis process, as it is increasingly unrealistic to build an all-in-one application that fulfills the needs of all analysts. Therefore, a system was presented that allows users to visually link related pieces of information across the boundaries of individual applications. Like the orientation techniques from the last chapter, this multi-application approach aims at providing orientation support on the level of individual data items (*S1.1*).

The second part of the chapter focused on collaborative information visualization where experts from various domains jointly perform an analysis in order to understand and draw meaningful conclusions from complex, heterogeneous data. It was shown how the information about assigned roles can be added on top of the analysis setup model. How this additional knowledge can be employed for data analysis purposes was then discussed. The section on multi-user considerations was closed with a brief introduction of the Caleydoplex prototype setup, a multi-display environment that is targeted at collaborative visual analysis.

Finally, this chapter was concluded by discussing the work on collaborative information linking that combines both of the main topics of this chapter, thus showing how the research of this thesis has inspired already published follow-up work.

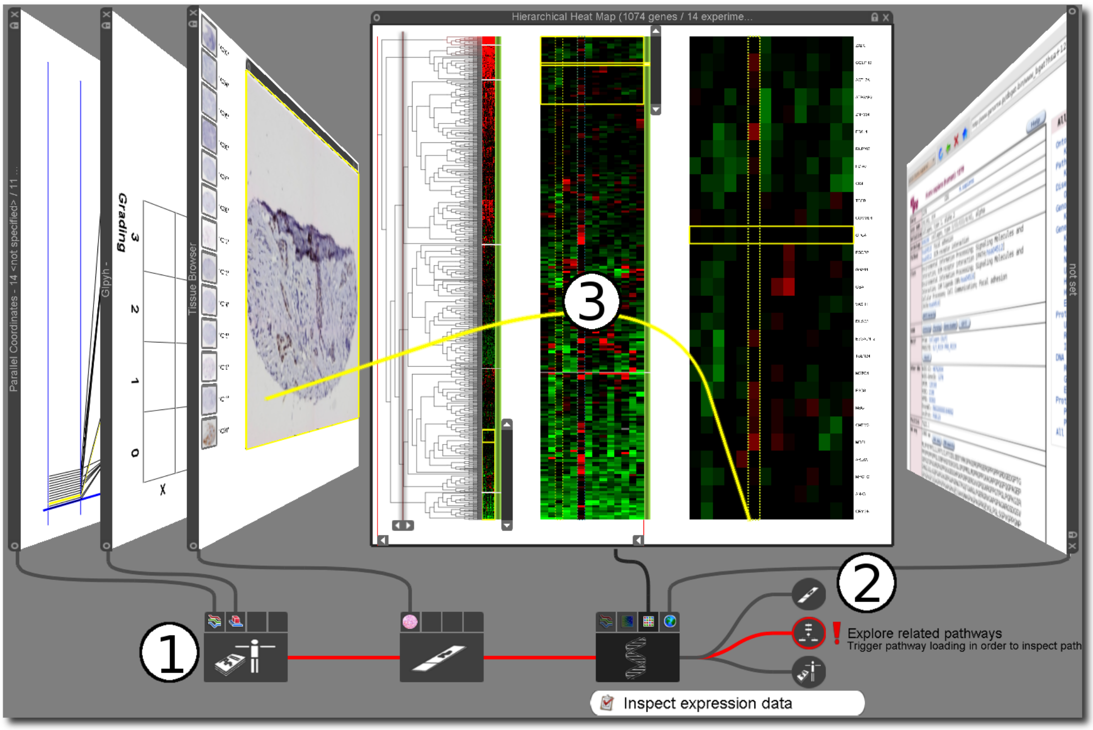
Chapter 8

Guidance Based on Full Three-Stage Model: Stack'n'flip

Contents

This chapter introduces Stack'n'flip, a concrete visual analysis system which makes use of all three stages of the authored model. The system aims at providing full analyst support in the sense of Section 3.1. On the one hand, it orients an analyst within the heterogeneous information landscape (*S1.1*), while at the same time encoding the history of previous steps (*S1.2*) as well as possible next analysis steps (*S1.3*). On the other hand, the system dynamically suggests future steps (*S2*) by means of a predefined workflow captured in the analysis session and domain model. This realizes the guidance support.

The Stack'n'flip system is grouped into two parts: a space for data visualization, similar to what [Shrinivasan and van Wijk, 2008] call the **Knowledge View** (see upper part in Figure 8.1), and a space showing the relations between data, views and analysis paths, similar to their **Navigation View** (lower part in Figure 8.1). While the realization and application goals of this system are very different to those proposed in [Shrinivasan and van Wijk, 2008], the views are conceptually similar. Therefore these terms were adopted. Two factors distinguish Stack'n'flip from other systems: first, the navigation and the knowledge view are seamlessly integrated (showing the relations between views and the data sets they visualize – *S1.1* support), and secondly, the kind of support based on the developed three-stage model goes well beyond provenance and history.


Parallel Coordinates - 14 <not specified> / 11 ...
Glyph -
Grading
3
2
1
0
X
Tissue Browser
Hierarchical Heat Map (1074 genes / 14 experime...
not set
1
2
3
Inspect expression data

Explore related pathways
Trigger pathway loading in order to inspect path

Figure 8.1: A snapshot of a sample analysis session using the Stack'n'flip system. The knowledge view contains a heat map view in the center, a tissue browser on the left, a web browser on the right and additional stacked views on both sides. The succession of large symbols in the navigation view at the bottom represents the analysis path taken, with each symbol showing a data set (1). On top of the data set symbols, smaller icons show which interfaces are available for the data set. Possible future steps or branches (2) are either highlighted red, symbolizing the suggested analysis path, or grey, showing alternative options. Visual links emphasize relations between the views (3).

Some approaches, such as Aruvi [Shrinivasan and van Wijk, 2008], History Mechanism [Kreuseler et al., 2004], Heer *et al.*'s temporal work for Tableau [Heer et al., 2008] and many others (*e.g.*, [Groth and Streefkerk, 2006]) visualize the exploratory process in a history tree – support on the level *S1.2*. This principal idea is taken a step further by not only presenting history information, but also proposing future steps – either showing possible next steps independent of a workflow (*S1.3*), or making suggestions according to a predefined path (*S2*). However, in contrast to the VisComplete approach of Vistrails [Koop et al., 2008], the path suggestions are not derived purely from previous sessions and workflows, but instead made by employing the authored models. In addition, the associations between previous and possible future analysis steps are made explicit on both levels – the navigation view and the knowledge view.

The Stack'n'flip implementation is also a part of the Caleydo visualization framework. The authored model is loaded from a predefined XML representation and is stored in a graph data structure. The interactive support of Stack'n'flip is based on simple graph traversal operations.

8.1 Knowledge View

Exploring multiple data sets naturally lends itself to the usage of multiple coordinated views. However, traditional systems often present those multiple views either in tiled windows or in tabs. This strategy does not correspond well to an analysis path, however, since it is frequently the case that the previous and subsequent data sets may be contextually relevant, while one data set is in focus.

To take this into consideration, a stacking of views is proposed as depicted in Figure 8.1. The views are projected and rendered on 2D planes within a 3D scene, making it in principle related to Apple's$^{\text{TM}}$Cover Flow, but also to the Jukebox (see Section 6.2.2) as well as the Bucket (Section 6.2.2). The view in focus is in the center and parallel to the screen. Other views are stacked to the left and right of the focus view, tilted towards the user. The adjacent views are either from the same data set, or from a data set explored in a previous (on the left) or upcoming (on the right) analysis step. This makes it possible to easily relate data in adjacent views. In addition to conventional highlighting of selected items, visual links are shown between related entities in adjacent views. The 2.5D layout was chosen, because the evaluation of the Bucket has shown that it is an effective method for working with multiple interconnected views. However, pure 2D layouts, avoiding problems arising from distortion, are possible as well.

The stacking approach allows the analyst to flip through the different views and also to re-examine, or to adapt a filter on a previously explored data set. If he finds that the filters on a previous level need refinement, he simply brings the view into focus and updates the filters. The changes are immediately reflected in all associated views. Due to the implicit sorting of the data sets according to the workflow, the chosen views along the workflow are next to each other in the stacked view representation. For example, a patient represented as a column in the centered heat map view can then be visually linked to the corresponding tissue image (*cf.*, (3) in Figure 8.1).

8.2 Navigation View

The contribution of the proposed approach is not primarily the view arrangement, but the orientation provided by a "map" through the information landscape – the navigation view. When designing such a navigation view, it is important to find a balance between the amount of information presented and the requirement to give as much space as possible to the knowledge view, which contains the actual information.

The map was realized by depicting the network of data sets as large symbols (see (1) in Figure 8.1). Transitions in the data model between loaded data sets are visible at any time, while all possible transitions are shown only when hovering over the associated symbol (see (2) in Figure 8.1). A red exclamation mark followed by a short description indicates that a precondition needs to be met before the analyst can continue to a data set. By picking one of those possible next data sets, the associated data is loaded and shown in the knowledge view. Its symbol is added to the navigation view permanently.

The association between interfaces and data sets, contained in the setup model, is shown as icons on top of the data set symbol. Opening a new interface for a particular data set is achieved by clicking the interface icon.

In case of a guided analysis (*S2* support), the information available through the analysis session model is employed to highlight the recommended path, while still showing other options to proceed (*i.e.*, switching from guidance to orientation support). The highlighting is realized in red ((2) in Figure 8.1). Recommended interfaces for performing the next task are also shown in red and are opened by default when the data set symbol is clicked. A short description of the current task is presented at the bottom of the navigation view.

The recommendations for future analysis steps are dynamically determined by traversing the graph of the three-stage model. A lookup operation in the compound graph of the three-stage model results in the information to which data sets the task is connected to and which visual and computational tools are associated. This extracted knowledge is then visually conveyed by the highlighted path and symbols in the navigation view. In cases where the analyst does not follow the suggestions, the system switches to the informed mode where only support on the level *S1* is provided by means of the information contained in the analysis setup and the domain model.

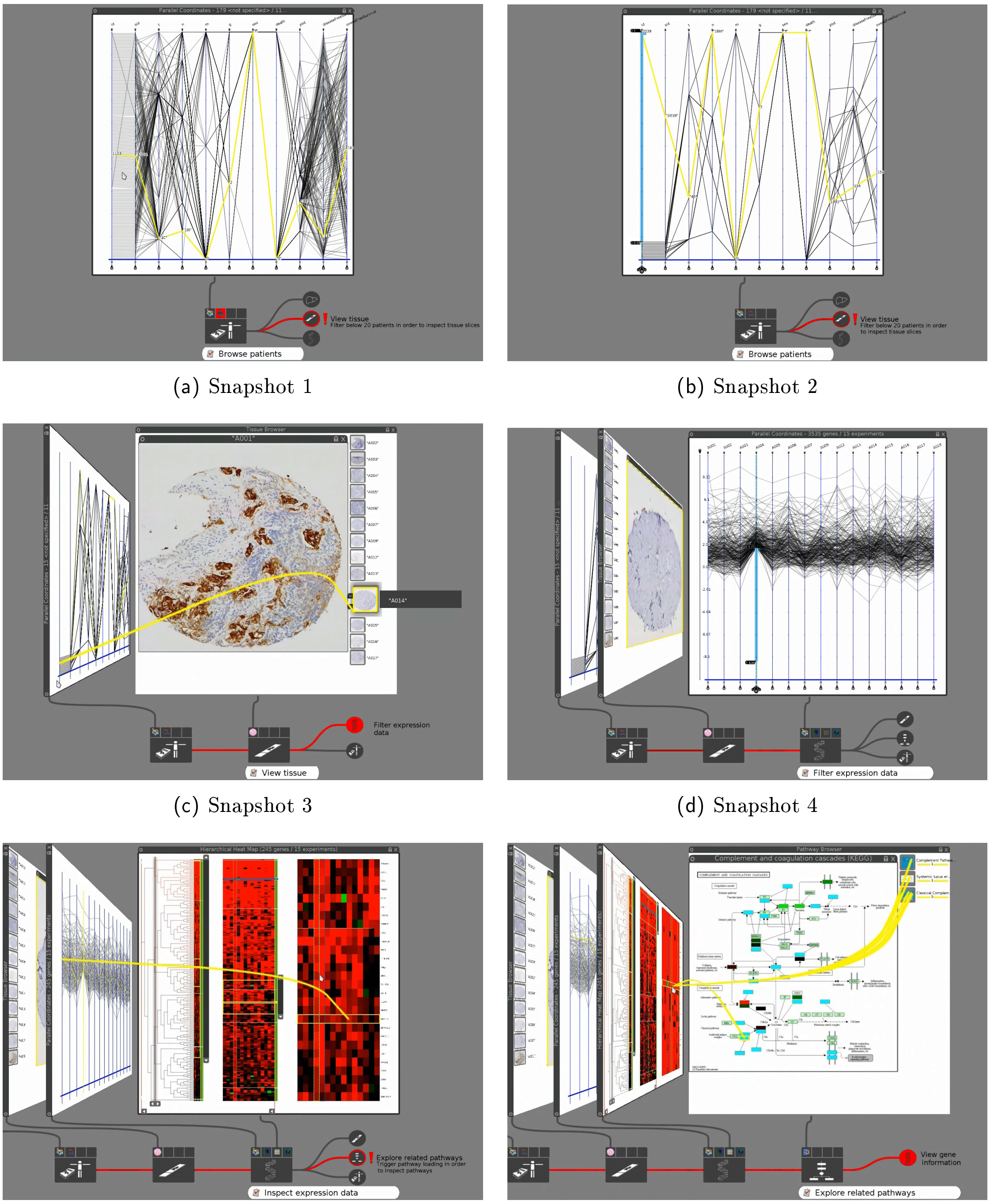
8.3 Fusion of Navigation View and Knowledge View

A key contribution of Stack'n'flip is the seamless integration of navigation and knowledge view. Open, active views are connected with a curve to their interface symbol on top of the data set symbol, thereby clarifying the relationship between the view and its data set. This association of data sets and views makes it explicit which data set is shown in which view, and also allows the unambiguous use of the same visualization technique for different data sets (*i.e.*, providing orientation support on the level *S1.1*).

This merging of interactive visualization with analysis context is related to Image Graphs [Ma, 1999], the P-Set Model [Jankun-Kelly et al., 2007] as well as the Graphical Histories [Heer et al., 2008]. However, Image Graphs and the P-Set Model capture only the analysis process operating on a homogeneous data set. In contrast, Heer's Graphical History view does handle heterogeneous data, but is restricted to history information and therefore does not support real guidance or orientation in the sense of Stack'n'flip.

8.4 Use Case Revisited: Sample Analysis Session

The following sample analysis session demonstrates how the Stack'n'flip system guides an analyst along a predefined workflow. The system employs the full three-stage model for realizing the analyst support. For demonstration purposes, the use case defined in Section 4.6.3 will be taken up.


Parallel Coordinates - 179 <not specified> / 11
View tissue
Filter below 20 patients in order to inspect tissue slices
Browse patients
Parallel Coordinates - 179 <not specified> / 11
View tissue
Filter below 20 patients in order to inspect tissue slices
Browse patients
Tissue Browser
"A001"
"A014"
Filter expression data
View tissue
Parallel Coordinates - 3535 genes / 15 experiments
Filter expression data
Hierarchical Heat Map (245 genes / 15 experiments)
Explore related pathways
Trigger pathway loading in order to inspect pathways
Inspect expression data
Pathway Browser
Complement and coagulation cascades (KEGG)
View gene information
Explore related pathways

(a) Snapshot 1

(b) Snapshot 2

(c) Snapshot 3

(d) Snapshot 4

(e) Snapshot 5

(f) Snapshot 6

Figure 8.2: Sample analysis session with the Stack'n'flip system.

The series of snapshots provided in Figure 8.2 shows an analyst realizing Task 2 to Task 8 of the workflow:

- **Snapshot 1:** Having completed the preprocessing on the vast patient data by statistical means (*cf.*, Task 1), the analyst starts with an already reduced set of patients. Initially, the knowledge view shows only parallel coordinates which are associated with the starting data set. In the parallel coordinates view, each axis encodes one attribute of the patient records and each polyline represents one patient. Following the task description on the bottom of the screen, the analyst browses the patient data (Task 2). The navigation view indicates the possible next steps to related data sets. The linked data sets are organ, tissue and gene expression data. However, the system also suggests continuing with the tissue data (highlighted red) and thus following the workflow.
- **Snapshot 2:** The exclamation mark indicates a precondition that has to be met before the analyst can proceed. Thus, the analyst visually filters the data set using the brushing tools in the parallel coordinates view. Once the data is filtered sufficiently (< 20), the associated symbol and the connecting curves are highlighted. As soon as the analyst clicks the pipe or the data set symbol, the default view for the selected data set is opened, showing the tissue images of the filtered patients. The tissue view is placed in the center and the parallel coordinates view is moved to the left side. Thus, the views stacked on the left hand side provide a visual history of the analysis process.
- **Snapshot 3:** The analyst is then instructed to perform the next task within the analysis workflow: browsing the tissue slices (Task 3). Visual links helps the analyst see which tissue image is associated to which polyline in the parallel coordinates view. Having gained an overview of the tissue slices from the various patients, the analyst is guided towards the next data set along the chosen workflow: the gene expression data. After the visual inspection of the tissue images, the user decides to keep all of the remaining patients (Task 4 “Discard patients”) and continues along the workflow.
- **Snapshot 4:** To prepare the data for clustering (Task 5), the analyst needs to filter the data set in order to meet the precondition of the clustering algorithm. The authoring makes sure that a suitable visual interface to perform the needed analysis task was chosen – which is a parallel coordinates view in this case, since it supports visual filtering. Therefore, in the current analysis situation, two parallel coordinates views are opened. However, due to the connection pipes between the the visual representation in the knowledge view and its data set symbol in the navigation view, the user is always aware which data set is associated to which visual representation.
- **Snapshot 5:** The analyst then runs a clustering algorithm, the result of which he explores in the heat map (Task 6). There he finds an interesting gene and triggers the loading of pathways to explore its biological context (Task 7).

- **Snapshot 6:** The system loads all pathways in which the gene plays a role. Visual links indicate the location of the gene in the pathways. Then the analyst can continue to inspect detailed information on the gene in the integrated browser view (Task 8).

Continuing with the suggested steps one by one in the guided analysis scenario guarantees that the workflow will eventually terminate by reaching the analysis goal. Finally, the analyst can base the treatment decision on the knowledge gained during the guided analysis session.

8.5 Discussion of the System

The author believes that the Stack'n'flip approach is general enough to be utilized in many different forms. In fact, as it mainly describes how to visually handle transitions in heterogeneous data analysis, it is applicable to a wide range of existing visualization frameworks.

As such, this system provides orientation (support level *S1*) when exploring heterogeneous data spaces by providing a history of previously explored data sets, a list of possible connected data sets (in the navigation view) as well as employed visualizations (through the stacking in the knowledge view) and is therefore suitable for the informed analyst. This is especially important in comprehensive analysis of data from different sources, as it requires the analyst to switch back and forth between different views and data sets, refining for example selections or filters. Each switch requires mental effort and is potentially confusing for the analyst. By making such switches seamless and keeping the source view as contextual information, the mental effort can be reduced significantly.

The guided analyst (support level *S2*) benefits from the explicit path laid out for him, while the navigation view shows possible alternatives – thereby encouraging a deviation from the predefined path (and therefore a switch from guided analyst to informed analyst) for a deeper understanding of the data.

Chapter 9

Conclusion

This dissertation investigated how an analysis system can assist a user in the analysis of a complex interwoven set of data. The analyst support a user needs is twofold:

- **orientation support**, where the user's mental map should be built and consequently maintained
- **guidance support**, where the oriented user is directed along a workflow towards a specific analysis goal.

In order to be able to realize these two levels of analyst support, this thesis proposed to design a model in which unified representation is achieved via an authoring process. This model extends a definition of the information landscape, as it also contains details on suitable visualization and computational methods to access the individual data sets as well as domain specific tasks that are combined to workflows, leading to a specific analysis goal. With the authored model as a conceptual basis, the thesis progressed by introducing a series of visualization techniques that realize the analyst support on both levels.

Summing up, the three main contributions of this thesis are:

- A novel **Model-Driven Design Concept** which captures the complexity of visual analysis in a structured way – including data sources, interfaces to access the data as well as tasks performed on the data. The model serves as the semantic basis for providing orientation to users in the first place, and, based on this, to guide them through an analysis session along an authored workflow towards a specific goal.
- **Visualization Techniques for Supported Analysis**: within a data set, across multiple data sets, across applications as well as across displays including multiple analysts.
- The design and implementation of the **Caleydo Visual Analysis Framework** which is the foundation for the realization of the proposed visualization techniques.

9.1 Further Implications

The thesis elaborated on the possibilities the model provides in terms of orientation and guidance. However, the externalization of the information about the setup can potentially be utilized for further purposes, as outlined in the following.

Visual Analytics goes well beyond simply providing the necessary tools for an analysis scenario – it also aims at helping the analysts in choosing the appropriate techniques by defining several processes as best practice solutions for given analytical objectives. These are based on high-level guidelines, such as Keim's Visual Analytics Mantra [Keim et al., 2006] or Shneiderman's well-established Information Seeking Mantra [Shneiderman, 1996]. Both have found their way into the design of Visual Analytics systems, as they give valuable advice on which kind of tools to provide at which point in the analysis. These processes can be understood as abstract design patterns for visual analysis software. However, they are too abstract to actually specify concrete visual analysis techniques that can be used on a concrete set of data. Hence, most approaches derive concrete suggestions for the analysis from low-level events (mouse clicks, *etc.*) recorded during previous analysis sessions.

In between high-level mantras and low-level mouse clicks, a gap emerges that neither can fill. A mid-level approach, like the one proposed, makes it possible to formulate analysis sessions as abstractly as needed in order to serve as reusable patterns and at the same time be specific enough to be used for concrete user support, thus merging the best of both worlds. However, in the proposed design approach, high-level mantras are still incorporated. Task 1-6 in Figure 9.1 is one example where Keim's Visual Analytics Mantra – "*analyse first - show the important - zoom, filter and analyse further - details-on-demand*" is evident.

The benefits of using the proposed model are twofold: on the one hand, it can be employed by a visual analysis system to provide analyst support on different levels, as already discussed in detail; on the other hand, it can help in the design phase of a complex analysis scenario. The following paragraphs discuss further potential benefits gained by defining a model which is more comprehensive.

Data Selection

The proposed concept makes it possible to dynamically select a set of relevant data sets for a specific analysis goal. Selecting a reduced list of data sets needed in an analysis session makes the analysis more targeted towards the goal. In addition, the system can anticipate the next steps of an analyst and preprocess, pre-fetch or pre-layout data in otherwise idle times. For example, fetching of large tissue images from databases can be triggered before the analyst traverses the data set during the interaction. An example for a time-consuming preprocessing step is clustering of gene expression data, based on a selection of patients. Since an analyst can always choose a path different from the preferred one, an option would

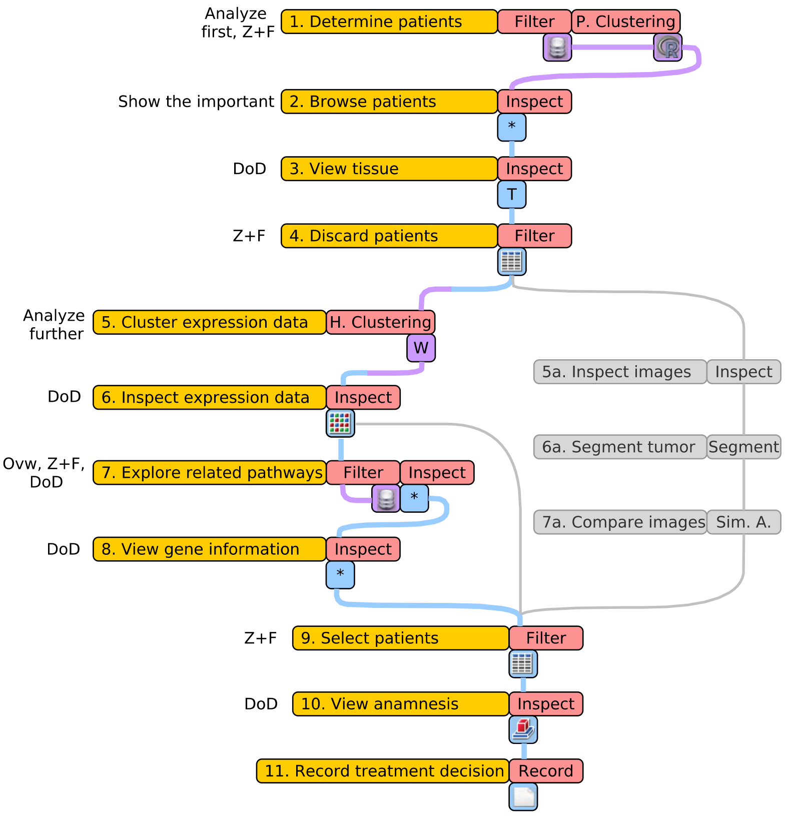


Figure 9.1: Sample analysis path showing the chosen interfaces. Jumps between computational (purple) and visual (blue) interfaces denote switches from the data to the view domain and vice versa. High-level interaction mantras can be found as reoccurring patterns.

be to pre-fetch data first for the preferred path, and then for other possible paths, if enough processing power, memory and/or bandwidth is available. By conducting such operations in a separate thread, such a system can utilize modern multi-core systems, resulting in a significant speed-up.

Missing Data or Interface Identification

When defining the analysis session model with an analysis goal in mind, interfaces or data sources needed to perform a task might be missing from the analysis setup. Due to the structured authoring process, however, missing interfaces or data sets are immediately obvious to the domain expert. At this early stage the domain expert can try to fill these gaps by requesting the missing data sets or interfaces from the data manager or visual analysis expert, respectively.

Generalization of Workflows

Analyst support based on history and provenance information is an integral part of various Visual Analytics systems (*e.g.*, [Willett et al., 2007, Bavoil et al., 2005]), as already discussed in Section 3.2. However, by logging low-level application events, the collected information is tightly coupled to one specific setup and cannot be reused for guidance purposes within different applications and tools. With the proposed association of tasks to application and domain independent operators, implementation internal matters are detached from the actual semantic path information. In principle, this indirection makes it possible to employ the collected path information in different analysis setups as well. It is even possible to unhinge the workflow with the associated domain independent operator sequences from a specific setup in order to find an alternative combination of analysis tools.

Post Analysis Optimization

Based on the analysis session model, it is possible to log the workflow path actually taken by a user during an analysis session. Figure 9.1 depicts an example path including the interfaces used for each step. Switches between the visual (purple) and computational (blue) domain are of special interest as these are often not seamless and therefore imply a higher mental effort for the user. The extracted knowledge can be utilized to:

- **optimize the workflow**
  By comparing the suggested path with the one taken by the user, a feedback loop can be introduced in the authoring process.
- **optimize the analysis framework**
  Based on the insights gained, the underlying application can be modified to better reflect the user's needs.

Tailor User Interface

The additional information given by the models can be used to tailor the user interface to the given data, the user preferences, and the tasks to be performed. For instance, feature-rich analysis software can be reduced to the minimum that is needed for the current task. However, just as a user should be able to switch during an analysis from orientation support to guidance and vice versa, the user should also be able to access the full feature set at any time.

9.2 Future Work

Today, many domain-specific visual analysis systems are being developed where the interplay between data, view, and task is hard-wired in the software. Thus, the information that is covered by the model-driven concept is present, but immutable. This makes systems inflexible and often incapable of reacting to changing requirements. Consequently, it would be desirable to mesh the authoring process with the system configuration process. This would allow the authors involved (*cf.*, the identified roles from Section 4.1: data manager, visualization expert and domain expert) to dynamically configure the visual analysis application to their specific needs. In a next step, this authoring process could also be integrated with the analysis sessions themselves – thus allowing the user to become the author. The domain expert could, for instance, interactively add tasks to the model and use them as building blocks for the arrangement of new workflows.

The current Stack'n'flip implementation provides guidance based on predefined models via the compact navigation view. Following the proposed online authoring approach, the navigation view could be switched on demand to a full authoring interface with on-the-fly model editing capabilities. This tight integration of authoring and data analysis has the potential to support a wide range of Visual Analytics applications.

As already mentioned, the Stack'n'flip system realizes guidance in classic multi-view applications. The system is only one possible way of employing the model for comprehensive analyst support and therefore leaves a lot of space for alternative approaches. However, setups that span independent applications, as well as collaborative multi-display environments impose new challenges for the user in terms of the assistance needed. For this reason, the idea of applying the model-driven approach to these scenarios as well seems to be a research direction with great potential. In particular, the realization of a guided analysis along a workflow that utilizes visual links across applications is a promising avenue for future research.

Bibliography

[Ackoff, 1989] R. L. Ackoff. From data to wisdom. *Journal of Applied System Analysis*, vol. 16, pp. 3–9, 1989.

[Ahl and Allen, 1996] V. Ahl and T. F. H. Allen. *Hierarchy theory: a vision, vocabulary, and epistemology*. Columbia University Press, 1996. ISBN 0231084802.

[Andrianantoandro et al., 2006] E. Andrianantoandro, S. Basu, D. K. Karig, and R. Weiss. Synthetic biology: new engineering rules for an emerging discipline. *Molecular Systems Biology*, vol. 2, p. 2006.0028, 2006. doi:10.1038/msb4100073.

[ao and Kirste, 2005] J. L. E. ao and T. Kirste. Ambient intelligence: Towards smart appliance ensembles. In *From Integrated Publication and Information Systems to Information and Knowledge Environments*, vol. 3379 of *Lecture Notes in Computer Science*, pp. 261–270. Springer, 2005. doi:10.1007/978-3-540-31842-2_26.

[Ashburner et al., 2000] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature Genetics*, vol. 25, no. 1, pp. 25–29, 2000. doi:10.1038/75556.

[Ashmore et al., 2005] M. Ashmore, A. T. Duchowski, and G. Shoemaker. Efficient eye pointing with a fisheye lens. In *Proceedings of the Conference on Graphics Interface (GI '05)*, pp. 203–210. Canadian Human-Computer Communications Society, 2005. ISBN 1568812655.

[Asslaber and Zatloukal, 2007] M. Asslaber and K. Zatloukal. Biobanks: transnational, european and global networks. *Briefings in Functional Genomics and Proteomics*, vol. 6, no. 3, pp. 193–201, 2007. doi:10.1093/bfgp/elm023.

[Baecker et al., 1995] R. M. Baecker, J. Grudin, W. Buxton, and S. Greenberg. *Readings in Human-Computer Interaction: Toward the Year 2000*. Morgan Kaufmann, second edn., 1995. ISBN 1558602465.

[Baldonado et al., 2000] M. Q. W. Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the ACM Conference on Advanced Visual Interfaces (AVI '00)*, pp. 110–119. ACM Press, 2000. ISBN 1581132522. doi:10.1145/345513.345271.

[Ball and Eick, 1996] T. Ball and S. G. Eick. Software visualization in the large. *Computer*, vol. 29, no. 4, pp. 33–43, 1996. doi:10.1109/2.488299.

[Barsky et al., 2008] A. Barsky, T. Munzner, J. Gardy, and R. Kincaid. Cerebral: Visualizing multiple experimental conditions on a graph with biological context. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '08)*, vol. 14, no. 6, pp. 1253–1260, 2008. doi:10.1109/TVCG.2008.117.

[Bavoil et al., 2005] L. Bavoil, S. Callahan, C. Scheidegger, H. Vo, P. Crossno, C. Silva, and J. Freire. VisTrails: enabling interactive Multiple-View visualizations. In *Proceedings of the IEEE Conference on Visualization (VIS '05)*, pp. 135–142. IEEE Computer Society Press, 2005. ISBN 0780394623. doi:10.1109/VISUAL.2005.1532788.

[Becker and Rojas, 2001] M. Becker and I. Rojas. A graph layout algorithm for drawing metabolic pathways. *Bioinformatics*, vol. 17, no. 5, pp. 461–467, 2001. doi:10.1093/bioinformatics/17.5.461.

[Becker, 1987] R. A. Becker. Dynamic graphics for data analysis. *Statistical Science*, vol. 2, no. 4, pp. 355–383, 1987. doi:10.1214/ss/1177013104.

[Bertin, 1983] J. Bertin. *Semiology of graphics*. University of Wisconsin Press, 1983. ISBN 0299090604.

[Bourqui et al., 2007] R. Bourqui, L. Cottret, V. Lacroix, D. Auber, P. Mary, M. Sagot, and F. Jourdan. Metabolic network visualization eliminating node redundance and preserving metabolic pathways. *BMC Systems Biology*, vol. 1, no. 29, 2007. doi:10.1186/1752-0509-1-29.

[Brandes et al., 2004] U. Brandes, T. Dwyer, and F. Schreiber. Visualizing related metabolic pathways in two and a half dimensions. In *Graph Drawing (GD '03)*, pp. 111–122. Springer, 2004. ISBN 3540208313. doi:10.1007/978-3-540-24595-7_10.

[Brennan et al., 2006] S. Brennan, K. Mueller, G. Zelinsky, I. Ramakrishnan, D. Warren, and A. Kaufman. Toward a Multi-Analyst, collaborative framework for visual analytics. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST '06)*, pp. 129–136. IEEE Computer Society Press, 2006. ISBN 1424405912. doi:10.1109/VAST.2006.261439.

[Carr et al., 1986] D. B. Carr, R. J. Littlefield, and W. L. Nichloson. Scatterplot matrix techniques for large n. In *Proceedings of the Symposium on the Interface of*

Computer Sciences and Statistics, pp. 297–306. Elsevier North-Holland, 1986. ISBN 0444700188. doi:10.2307/2289444.

[Chen, 2003] H. Chen. Compound brushing. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '03)*, pp. 181–188. IEEE Computer Society Press, 2003. ISBN 0780381548. doi:10.1109/INFVIS.2003.1249024.

[Chen et al., 2009] M. Chen, D. Ebert, H. Hagen, R. S. Laramee, R. van Liere, K. Ma, W. Ribarsky, G. Scheuermann, and D. Silver. Data, information, and knowledge in visualization. *IEEE Computer Graphics and Applications*, vol. 29, pp. 12–19, 2009. doi:10.1109/MCG.2009.6.

[Chen, 1976] P. P. S. Chen. The entity-relationship model toward a unified view of data. *ACM Transactions on Database Systems (TODS '76)*, vol. 1, no. 1, pp. 9–36, 1976. doi:10.1145/320434.320440.

[Chung et al., 2005] H. Chung, C. H. Park, M. R. Han, S. Lee, J. H. Ohn, J. Kim, J. Kim, and J. H. Kim. ArrayXPath II: mapping and visualizing microarray gene-expression data with biomedical ontologies and integrated biological pathway resources using scalable vector graphics. *Nucleic Acids Research*, vol. 33, no. Web Server issue, pp. W621–W626, 2005. doi:10.1093/nar/gki450.

[Collins and Carpendale, 2007] C. Collins and S. Carpendale. VisLink: revealing relationships amongst visualizations. *Proceedings of the IEEE Transactions on Visualization and Computer Graphics (InfoVis '07)*, vol. 13, no. 6, pp. 1192–1199, 2007. doi:10.1109/TVCG.2007.70521.

[Collins et al., 2009] C. Collins, G. Penn, and S. Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *Proceedings of the IEEE Transactions on Visualization and Computer Graphics (InfoVis '09)*, vol. 15, no. 6, pp. 1009–1016, 2009. doi:10.1109/TVCG.2009.122.

[Collins et al., 2003] F. S. Collins, E. D. Green, A. E. Guttmacher, and M. S. Guyer. A vision for the future of genomics research. *Nature*, vol. 422, no. 6934, pp. 835–847, 2003. doi:10.1038/nature01626.

[Dean and Ghemawat, 2008] J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008. doi:10.1145/1327452.1327492.

[Dietzsch et al., 2006] J. Dietzsch, N. Gehlenborg, and K. Nieselt. Mayday–a microarray data analysis workbench. *Bioinformatics*, vol. 22, no. 8, pp. 1010–1012, 2006. doi:10.1093/bioinformatics/btl070.

[Dietzsch et al., 2009] J. Dietzsch, J. Heinrich, K. Nieselt, and D. Bartz. SpRay: a visual analytics approach for gene expression data. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST '09)*, pp. 179–186, 2009. ISBN 1424452835. doi:10.1109/VAST.2009.5333911.

[Dijkstra, 1959] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, vol. 1, pp. 269–271, 1959. doi:10.1007/BF01386390.

[Doleisch et al., 2003] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *Proceedings of the Symposium on Data Visualisation (VISSYM '03)*, pp. 239–248. Eurographics Association, 2003. ISBN 1581136986.

[Eisen et al., 1998] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences USA*, vol. 95, no. 25, pp. 14863–14868, 1998. doi:10.1073/pnas.95.25.14863.

[Ellis and Dix, 2006] G. Ellis and A. Dix. Enabling automatic clutter reduction in parallel coordinate plots. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '06)*, vol. 12, no. 5, pp. 717–724, 2006. doi:10.1109/TVCG.2006.138.

[Evanko, 2010] D. Evanko (Editor). *Supplement on visualizing biological data*, vol. 7 no. 3s. Nature Methods, 2010.

[Fairchild et al., 1988] K. M. Fairchild, S. E. Poltrock, and G. W. Furnas. SemNet: Three-Dimensional graphic representation of large knowledge bases. In *Cognitive Science and its Applications for Human-Computer Interaction*, pp. 201–233. Lawrence Erlbaum Associates, 1988. ISBN 0898598842.

[Fekete et al., 2003] J. Fekete, D. Wang, N. Dang, A. Aris, and C. Plaisant. Interactive poster: Overlaying graph links on treemaps. In *Proceedings of the IEEE Symposium on Information Visualization Conference Compendium (InfoVis '03)*, pp. 82–83. IEEE Computer Society Press, 2003.

[Fischer et al., 2008] F. Fischer, F. Mansmann, D. A. Keim, S. Pietzko, and M. Waldvogel. Large-Scale network monitoring for visual analysis of attacks. In *Proceedings of the Workshop on Visualization for Computer Security (VizSec '08)*, pp. 111–118. Springer, 2008. ISBN 3540859314. doi:10.1007/978-3-540-85933-8_11.

[Fono and Vertegaal, 2005] D. Fono and R. Vertegaal. EyeWindows: evaluation of eye-controlled zooming windows for focus selection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*, pp. 151–160. ACM Press, 2005. ISBN 1581139985. doi:10.1145/1054972.1054994.

[Frey and Dueck, 2007] B. J. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, vol. 315, no. 5814, pp. 972–976, 2007. doi:10.1126/science.1136800.

[Gamma et al., 1995] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman, 1995. ISBN 0201633612.

[Garg et al., 2008] S. Garg, J. Nam, I. Ramakrishnan, and K. Mueller. Model-driven visual analytics. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST '08)*, pp. 19–26. IEEE Computer Society Press, 2008. ISBN 1424429356. doi:10.1109/VAST.2008.4677352.

[Gehlenborg et al., 2010] N. Gehlenborg, S. I. O'Donoghue, N. S. Baliga, A. Goesmann, M. A. Hibbs, H. Kitano, O. Kohlbacher, H. Neuweger, R. Schneider, D. Tenenbaum, and A. Gavin. Visualization of omics data for systems biology. *Nature Methods*, vol. 7, no. 3, pp. 56–68, 2010. doi:10.1038/nmeth.1436.

[Gotz and Wen, 2009] D. Gotz and Z. Wen. Behavior-driven visualization recommendation. In *Proceedings of the Conference on Intelligent User Interfaces (IUI '09)*, pp. 315–324. ACM Press, 2009. ISBN 1605581682. doi:10.1145/1502650.1502695.

[Gotz and Zhou, 2009] D. Gotz and M. X. Zhou. Characterizing users' visual analytic activity for insight provenance. *Information Visualization*, vol. 8, no. 1, pp. 42–55, 2009. doi:10.1057/ivs.2008.31.

[Gribov et al., 2010] A. Gribov, M. Sill, S. Luck, F. Rucker, K. Dohner, L. Bullinger, A. Benner, and A. Unwin. SEURAT: visual analytics for the integrated analysis of microarray data. *BMC Medical Genomics*, vol. 3, no. 1, p. 21, 2010. doi:10.1186/1755-8794-3-21.

[Groth and Streefkerk, 2006] D. P. Groth and K. Streefkerk. Provenance and annotation for visual exploration systems. *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 6, pp. 1500–1510, 2006. doi:10.1109/TVCG.2006.101.

[Hackos and Redish, 1998] J. T. Hackos and J. C. Redish. *User and Task Analysis for Interface Design*. First edn., 1998. ISBN 0471178314.

[Hagen and Carlstedt-Duke, 2004] H. Hagen and J. Carlstedt-Duke. Building global networks for human diseases: genes and populations. *Nature Medicine*, vol. 10, no. 7, pp. 665–667, 2004. doi:10.1038/nm0704-665.

[Hall et al., 2009] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009. doi:10.1145/1656274.1656278.

[Hauser et al., 2002] H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing of extended parallel coordinates. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '02)*, pp. 127–130. IEEE Computer Society Press, 2002. ISBN 076951751X. doi:10.1109/INFVIS.2002.1173157.

[Healey, 1996] C. G. Healey. Choosing effective colours for data visualization. In *Proceedings of the IEEE Conference on Visualization (Vis '96)*, pp. 263–ff. IEEE Computer Society Press, 1996. ISBN 0897918649. doi:10.1109/VISUAL.1996.568118.

[Heer and Agrawala, 2006] J. Heer and M. Agrawala. Software design patterns for information visualization. *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 853–860, 2006. doi:10.1109/TVCG.2006.178.

[Heer and Agrawala, 2007] J. Heer and M. Agrawala. Design considerations for collaborative visual analytics. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST '07)*, pp. 171–178. IEEE Computer Society Prss, 2007. ISBN 1424416592. doi:10.1109/VAST.2007.4389011.

[Heer et al., 2005] J. Heer, S. K. Card, and J. A. Landay. prefuse: a toolkit for interactive information visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*, pp. 421–430. ACM Press, 2005. ISBN 1581139985. doi:10.1145/1054972.1055031.

[Heer et al., 2008] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '08)*, vol. 14, no. 6, pp. 1189–1196, 2008. doi:10.1109/TVCG.2008.137.

[Hiley et al., 2006] J. B. Hiley, A. H. Redekopp, and R. Fazel-Rezai. A low cost human computer interface based on eye tracking. In *Engineering in Medicine and Biology Society (EMBS '06)*, pp. 3226–3229, 2006. ISBN 1424400325. doi:10.1109/IEMBS.2006.260774.

[Hoellerer et al., 2007] T. H. Hoellerer, G. G. Robertson, D. D. Thiel, D. C. Robbins, and M. R. van Dantzich. United states patent: 7263667 - methods, apparatus and data structures for providing a user interface which facilitates decision making, 2007.

[Hoffmann et al., 2008] R. Hoffmann, P. Baudisch, and D. S. Weld. Evaluating visual cues for window switching on large screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*, pp. 929–938. ACM Press, 2008. ISBN 1605580111. doi:10.1145/1357054.1357199.

[Holten, 2006] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '06)*, vol. 12, no. 5, pp. 741–748, 2006. doi:10.1109/TVCG.2006.147.

[Huang et al., 2008] D. W. Huang, B. T. Sherman, and R. A. Lempicki. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature Protocols*, vol. 4, no. 1, pp. 44–57, 2008. doi:10.1038/nprot.2008.211.

[Ingram et al., 2010] S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner, and T. Möller. DimStiller: workflows for dimensional analysis and reduction. In *Proceedings of the IEEE Conference on Visual Analytics in Science and Technology (VAST '10)*. IEEE Computer Society Press, 2010. ISBN 1424494880. doi:10.1109/VAST.2010.5652392.

[Isenberg and Fisher, 2009] P. Isenberg and D. Fisher. Collaborative brushing and linking for co-located visual analytics of document collections. *Computer Graphics Forum (EuroVis '09)*, vol. 28, no. 3, pp. 1031–1038, 2009. doi:10.1111/j.1467-8659.2009.01444.x.

[Jankun-Kelly and Ma, 2001] T. Jankun-Kelly and K. Ma. Visualization exploration and encapsulation via a Spreadsheet-Like interface. *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, no. 3, pp. 275–287, 2001. doi:10.1109/2945.942695.

[Jankun-Kelly et al., 2007] T. J. Jankun-Kelly, K. Ma, and M. Gertz. A model and framework for visualization exploration. *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 2, pp. 357–369, 2007. doi:10.1109/TVCG.2007.28.

[Jianu et al., 2010] R. Jianu, K. Yu, L. Cao, V. Nguyen, A. Salomon, and D. Laidlaw. Visual integration of quantitative proteomic data, pathways, and protein interactions. *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 4, pp. 609–620, 2010. doi:10.1109/TVCG.2009.106.

[Johansen et al., 1988] R. Johansen, J. Charles, R. Mittman, and P. Saffo. *Groupware: Computer Support for Business Teams*. Free Press, 1988. ISBN 0029164915.

[Johnson and Shneiderman, 1991] B. Johnson and B. Shneiderman. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the IEEE Conference on Visualization (Vis '91)*, p. 284–291, 1991. ISBN 0818622458. doi:10.1109/VISUAL.1991.175815.

[Kanehisa et al., 2006] M. Kanehisa, S. Goto, M. Hattori, K. F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, and M. Hirakawa. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res*, vol. 34, no. Database issue, pp. 354–357, 2006. doi:10.1093/nar/gkj102.

[Kang et al., 2009] Y. Kang, C. Gorg, and J. Stasko. Evaluating visual analytics systems for investigative analysis: Deriving design principles from a case study. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST '09)*, pp. 139–146. IEEE Computer Society Press, 2009. ISBN 1424452835. doi:10.1109/VAST.2009.5333878.

[Karp and Paley, 1994] P. D. Karp and S. M. Paley. Automated drawing of metabolic pathways. In *Processdings of the Conference on Bioinformatics and Genome Research*, 1994.

[Kashofer et al., 2009] K. Kashofer, M. M. Tschernatsch, H. J. Mischinger, F. Iberer, and K. Zatloukal. The disease relevance of human hepatocellular xenograft models: molecular characterization and review of the literature. *Cancer Letters*, vol. 286, no. 1, pp. 121–128, 2009. doi:10.1016/j.canlet.2008.11.011.

[Keim et al., 2010] D. A. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann (Editors). *Mastering The Information Age - Solving Problems with Visual Analytics*. Eurographics, 2010.

[Keim et al., 2006] D. A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler. Challenges in visual data analysis. In *Proceedings of the Conference on Information Visualisation (IV '06)*, pp. 9–14, 2006. ISBN 0769526020. doi:10.1109/IV.2006.31.

[Keim et al., 2009] D. A. Keim, F. Mansmann, and J. Thomas. Visual analytics: how much visualization and how much analytics? *SIGKDD Explorations*, vol. 11, no. 2, pp. 5–8, 2009. doi:10.1145/1809400.1809403.

[Klukas and Schreiber, 2006] C. Klukas and F. Schreiber. Dynamic exploration and editing of KEGG pathway diagrams. *Bioinformatics*, vol. 23, no. 3, pp. 344–350, 2006. doi:10.1093/bioinformatics/btl611.

[Kochevar and Wanger, 1995] P. Kochevar and L. Wanger. The tecate data space exploration utility. In *Proceedings of the Symposium on Interactive 3D Graphics (I3D '95)*, pp. 157–164. ACM Press, 1995. ISBN 0897917367. doi:10.1145/199404.199431.

[Koop et al., 2008] D. Koop, C. E. Scheidegger, S. P. Callahan, H. T. Vo, J. Freire, and C. T. Silva. VisComplete: automating suggestions for visualization pipelines. *IEEE Transactions on Visualization and Computer Graphics (Vis '08)*, vol. 14, no. 6, pp. 1691–1698, 2008. doi:10.1109/TVCG.2008.174.

[Kosara et al., 2006] R. Kosara, F. Bendix, and H. Hauser. Parallel sets: Interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 558–568, 2006. doi:10.1109/TVCG.2006.76.

[Kreuseler et al., 2004] M. Kreuseler, T. Nocke, and H. Schumann. A history mechanism for visual data mining. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '04)*, pp. 49–56. IEEE Computer Society Press, 2004. ISBN 0780387793. doi:10.1109/INFVIS.2004.2.

[Kreuseler and Schumann, 2002] M. Kreuseler and H. Schumann. A flexible approach for visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 39–51, 2002. doi:10.1109/2945.981850.

[Krzywinski et al., 2009] M. Krzywinski, J. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra. Circos: An information aesthetic for comparative genomics. *Genome Research*, vol. 19, no. 9, pp. 1639–1645, 2009. doi:10.1101/gr.092759.109.

[Lauwers and Lantz, 1990] J. C. Lauwers and K. A. Lantz. Collaboration awareness in support of collaboration transparency: requirements for the next generation of shared window systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90)*, pp. 303–311. ACM Press, 1990. ISBN 0201509326. doi:10.1145/97243.97301.

[Lee, 2008] J. C. Lee. Hacking the nintendo wii remote. *IEEE Pervasive Computing*, vol. 7, no. 3, p. 39–45, 2008. doi:10.1109/MPRV.2008.53.

[Lex, 2008] A. Lex. Exploration of gene expression data in a visually linked environment, 2008. Master thesis, Graz University of Technology.

[Lex et al., 2010a] A. Lex, M. Streit, E. Kruijff, and D. Schmalstieg. Caleydo: Design and evaluation of a visual analysis framework for gene expression data in its biological context. In *Proceeding of the IEEE Symposium on Pacific Visualization (PacificVis '10)*, pp. 57–64. IEEE Computer Society Press, 2010a. ISBN 424466856. doi:10.1109/PACIFICVIS.2010.5429609.

[Lex et al., 2010b] A. Lex, M. Streit, C. Partl, K. Kashofer, and D. Schmalstieg. Comparative analysis of multidimensional, quantitative data. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '10)*, vol. 16, no. 6, pp. 1027–1035, 2010b. doi:10.1109/TVCG.2010.138.

[Lieberman et al., 2010] M. D. Lieberman, S. Taheri, H. Guo, F. Mir-Rashed, I. Yahav, A. Aris, and B. Shneiderman. Visual exploration across biomedical databases. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 2, pp. 536–550, 2010. doi:10.1109/TCBB.2010.1.

[Lindroos and Andersson, 2002] H. Lindroos and S. G. E. Andersson. Visualizing metabolic pathways: comparative genomics and expression analysis. *Proceedings of the IEEE*, vol. 90, no. 11, pp. 1793–1802, 2002. doi:10.1109/JPROC.2002.804687.

[Ma, 1999] K. Ma. Image graphs – a novel approach to visual data exploration. In *Proceedings of the IEEE Conference on Visualization (Vis '99)*, pp. 81–88, 1999. ISBN 078035897. doi:10.1109/VISUAL.1999.809871.

[Mackinlay et al., 1991] J. D. Mackinlay, G. G. Robertson, and S. K. Card. The perspective wall: detail and context smoothly integrated. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '91)*, pp. 173–176. ACM Press, 1991. ISBN 0897913833. doi:10.1145/108844.108870.

[Marrs et al., 1993] K. A. Marrs, S. A. Steib, C. A. Abrams, and M. G. Kahn. Unifying heterogeneous distributed clinical data in a relational database. In *Proceedings of the Symposium on Computer Applications in Medical Care*, pp. 644–648, 1993.

[Martin and Ward, 1995] A. R. Martin and M. O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *Proceedings of the IEEE Conference on Visualization (Vis '95)*, p. 271. IEEE Computer Society Press, 1995. ISBN 0818671874. doi:10.1109/VISUAL.1995.485139.

[McAffer and Lemieux, 2005] J. McAffer and J. Lemieux. *Eclipse Rich Client Platform: Designing, Coding, and Packaging Java(TM) Applications*. Addison-Wesley Professional, 2005. ISBN 0321334612.

[Meyer et al., 2009] M. Meyer, T. Munzner, and H. Pfister. MizBee: a multiscale synteny browser. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '09)*, vol. 15, no. 6, pp. 897–904, 2009. doi:10.1109/TVCG.2009.167.

[Meyer et al., 2010] M. Meyer, B. Wong, M. Styczynski, T. Munzner, and H. Pfister. Pathline: A tool for comparative functional genomics. *Computer Graphics Forum (EuroVis '10)*, vol. 29, no. 3, pp. 1043–1052, 2010. doi:10.1111/j.1467-8659.2009.01710.x.

[Michal, 1999] G. Michal. *Biochemical Pathways. Biochemie-Atlas*. Spektrum Akademischer Verlag, 1999.

[Mitchell and Kennedy, 1997] K. Mitchell and J. Kennedy. The perspective tunnel: An inside view on smoothly integrating detail and context. In *Proceedings of the Eurographics Workshop on Visualisation*, 1997. ISBN 3211830499.

[Mlecnik et al., 2005] B. Mlecnik, M. Scheideler, H. Hackl, J. Hartler, F. Sanchez-Cabo, and Z. Trajanoski. PathwayExplorer: web service for visualizing high-throughput expression data on biological pathways. *Nucleic Acids Research*, vol. 33, no. Web Server issue, pp. 633–637, 2005. doi:10.1093/nar/gki391.

[Mueller et al., 2009] H. Mueller, R. Reihs, S. Sauer, K. Zatloukal, M. Streit, L. Alexander, B. Schlegl, and D. Schmalstieg. Connecting genes with diseases. In *Proceedings of the Conference on Information Visualisation (IV '09)*. IEEE Computer Society Press, 2009. ISBN 0769537337. doi:10.1109/IV.2009.86.

[Mueller et al., 2008] H. Mueller, K. Zatloukal, M. Streit, and D. Schmalstieg. Interactive exploration of medical data sets. In *Proceedings of the Conference of BioMedical*

Visualisation, pp. 29–35. IEEE Computer Society Press, 2008. ISBN 0769532844. doi:10.1109/MediVis.2008.13.

[Munzner, 2009] T. Munzner. A nested process model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '09)*, vol. 15, no. 6, pp. 921–928, 2009. doi:10.1109/TVCG.2009.111.

[Munzner et al., 2003] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou. TreeJuxtaposer: scalable tree comparison using Focus+Context with guaranteed visibility. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '03)*, pp. 453–462. ACM Press, 2003. ISBN 1581137095. doi:10.1145/1201775.882291.

[Natarajan and Ganz, 2009] S. Natarajan and A. Ganz. Distributed visual analytics for collaborative emergency response management. In *Proceedings of the IEEE Conference on Engineering in Medicine and Biology Society (EMBC '09)*, vol. 2009, pp. 1714–1717, 2009. ISBN 1424432967. doi:10.1109/IEMBS.2009.5333481.

[Nielsen, 2009] M. Nielsen. A guide to the day of big data. *Nature*, vol. 462, no. 7274, pp. 722–723, 2009. doi:10.1038/462722a.

[North and Shneiderman, 2000] C. North and B. Shneiderman. Snap-Together visualization: A user interface for coordinating visualizations via relational schemata. In *Proceedings of the ACM Conference on Advanced Visual Interfaces (AVI '00)*, pp. 128–135. ACM Press, 2000. ISBN 1581132522. doi:10.1145/345513.345282.

[O'Day and Jeffries, 1993] V. L. O'Day and R. Jeffries. Orienteering in an information landscape: how information seekers get from here to there. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (INTERACT '93 and CHI '93)*, pp. 438–445, 1993. ISBN 0897915755. doi:10.1145/169059.169365.

[O'Donoghue et al., 2010] S. O'Donoghue, A. Gavin, N. Gehlenborg, D. Goodsell, J. Hériché, C. Nielsen, C. North, A. Olson, J. Procter, D. Shattuck, T. Walter, and B. Wong. Visualizing biological data-now and in the future. *Nature Methods*, vol. 7, no. 3s, pp. 2–4, 2010. doi:10.1038/nmeth.f.301.

[Palmer and Rock, 1994] S. Palmer and I. Rock. Rethinking perceptual organization: the role of uniform connectedness. *Psychonomic Bulletin and Review*, vol. 1, no. 1, p. 29–55, 1994. doi:10.3758/BF03200760.

[Panzitt et al., 2007] K. Panzitt, M. M. O. Tschernatsch, C. Guelly, T. Moustafa, M. Stradner, H. M. Strohmaier, C. R. Buck, H. Denk, R. Schroeder, M. Trauner, and K. Zatloukal. Characterization of HULC, a novel gene with striking up-regulation in hepatocellular carcinoma, as noncoding RNA. *Gastroenterology*, vol. 132, no. 1, pp. 330–342, 2007. doi:10.1053/j.gastro.2006.08.026.

[Pavlopoulos et al., 2008] G. Pavlopoulos, S. O'Donoghue, V. Satagopam, T. Soldatos, E. Pafilis, and R. Schneider. Arena3D: visualization of biological networks in 3D. *BMC Systems Biology*, vol. 2, no. 1, p. 104, 2008. doi:10.1186/1752-0509-2-104.

[Pellegrini et al., 2001] M. Pellegrini, M. Thompson, J. Fierro, and P. Bowers. Computational method to assign microbial genes to pathways. *Journal of Cellular Biochemistry*, vol. 84, no. 34s, pp. 106–109, 2001. doi:10.1002/jcb.10071.

[Pennisi, 2005] E. Pennisi. How will big pictures emerge from a sea of biological data? *Science*, vol. 309, no. 5731, p. 94, 2005. doi:10.1126/science.309.5731.94.

[Perer and Shneiderman, 2008] A. Perer and B. Shneiderman. Systematic yet flexible discovery: guiding domain experts through exploratory data analysis. In *Proceedings of the ACM Conference on Intelligent User Interfaces (IUI '08)*, pp. 109–118. ACM Press, 2008. ISBN 1595939876. doi:10.1145/1378773.1378788.

[Perlin and Fox, 1993] K. Perlin and D. Fox. Pad: an alternative approach to the computer interface. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '93)*, SIGGRAPH '93, pp. 57–64. ACM Press, 1993. ISBN 0897916018. doi:10.1145/166117.166125.

[Pirchheim et al., 2009] C. Pirchheim, M. Waldner, and D. Schmalstieg. Deskotheque: Improved spatial awareness in Multi-Display environments. In *Proceedings of the IEEE Conference on Virtual Reality (VR '09)*, 2009. ISBN 1424439430. doi:10.1109/VR.2009.4811010.

[Piringer et al., 2009] H. Piringer, C. Tominski, P. Muigg, and W. Berger. A Multi-Threading architecture to support interactive visual exploration. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '09)*, vol. 15, no. 6, pp. 1113–1120, 2009. doi:10.1109/TVCG.2009.110.

[Puerta, 1997] A. Puerta. A model-based interface development environment. *IEEE Software*, vol. 14, no. 4, pp. 40–47, 1997. doi:10.1109/52.595902.

[R Development Core Team, 2010] R Development Core Team. *R: A Language and Environment for Statistical Computing*, 2010. ISBN 3900051070.

[Raskar et al., 1998] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The office of the future: a unified approach to image-based modeling and spatially immersive displays. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*, pp. 179–188. ACM Press, 1998. ISBN 0897919998. doi:10.1145/280814.280861.

[Risch et al., 1996] J. Risch, R. May, S. Dowson, and J. Thomas. A virtual environment for multimedia intelligence data analysis. *IEEE Computer Graphics and Applications*, vol. 16, no. 6, pp. 33–41, 1996. doi:10.1109/38.544070.

[Roberts, 2007] J. C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Proceedings of the Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV '07)*, pp. 61–71. IEEE Computer Society Press, 2007. ISBN 0769529038. doi:10.1109/CMV.2007.20.

[Robertson et al., 2000] G. Robertson, M. van Dantzich, D. Robbins, M. Czerwinski, K. Hinckley, K. Risden, D. Thiel, and V. Gorokhovsky. The task gallery: a 3D window manager. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '00)*, pp. 494–501. ACM Press, 2000. ISBN 1581132166. doi:10.1145/332040.332482.

[Robertson et al., 1991] G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone trees: animated 3D visualizations of hierarchical information. In *Proceedings of the SIGCHI Conference on Human factors in Computing systems (CHI '91)*, pp. 189–194. ACM Press, 1991. ISBN 0897913833. doi:10.1145/108844.108883.

[Rojdestvenski, 2003] I. Rojdestvenski. Metabolic pathways in three dimensions. *Bioinformatics*, vol. 19, no. 18, pp. 2436–2441, 2003. doi:10.1093/bioinformatics/btg342.

[Saldanha, 2004] A. J. Saldanha. Java treeview - extensible visualization of microarray data. *Bioinformatics*, vol. 20, no. 17, pp. 3246 –3248, 2004. doi:10.1093/bioinformatics/bth349.

[Saraiya et al., 2005] P. Saraiya, C. North, and K. Duca. Visualizing biological pathways: requirements analysis, systems evaluation and research agenda. *Information Visualization*, vol. 4, no. 3, pp. 191–205, 2005. doi:10.1057/palgrave.ivs.9500102.

[Sarkar et al., 1993] M. Sarkar, S. S. Snibbe, O. J. Tversky, and S. P. Reiss. Stretching the rubber sheet: a metaphor for viewing large layouts on small screens. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST ' 93)*, pp. 81–91. ACM Press, 1993. ISBN 089791628X. doi:10.1145/168642.168650.

[Scheidegger, 2009] C. E. Scheidegger. *Provenance of Exploratory Tasks in Scientific Visualization: Management and Applications.* Ph.D. thesis, University of Utah, 2009.

[Schena et al., 1995] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, vol. 270, no. 5235, pp. 467–470, 1995. doi:10.1126/science.270.5235.467.

[Schlegl, 2009] B. Schlegl. Visual analytics for gene expression data, 2009. Master thesis, Graz University of Technology.

[Schmidt-Gann et al., 2009] G. Schmidt-Gann, K. Schmid, M. Uehlein, J. Struck, A. Bergmann, D. Schmalstieg, M. Streit, A. Lex, D. G. van der Nest, M. van Griensven, and H. Redl. Gene- and protein expression profiling in liver in a Sepsis-Baboon model. In *Proceedings of the Conference on Shock*, 2009.

[Scholtz, 2009] J. Scholtz. Interactive poster: A proposal for sharing user requirements for visual analytic tools. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST '09)*, 2009. ISBN 1424452835. doi:10.1109/VAST.2009.5333474.

[Schulz, 2010] H. Schulz. *Explorative Graph Visualization*. Ph.D. thesis, University of Rostock, 2010.

[Seo and Shneiderman, 2002] J. Seo and B. Shneiderman. Interactively exploring hierarchical clustering results. *Computer*, vol. 35, no. 7, pp. 80–86, 2002. doi:10.1109/MC.2002.1016905.

[Shneiderman, 1996] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages (VL '96)*, pp. 336–343, 1996. ISBN 081867508X. doi:10.1109/VL.1996.545307.

[Shneiderman and Aris, 2006] B. Shneiderman and A. Aris. Network visualization by semantic substrates. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '06)*, vol. 12, no. 5, pp. 733–740, 2006. doi:10.1109/TVCG.2006.166.

[Shoemake, 1985] K. Shoemake. Animating rotation with quaternion curves. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '85)*, pp. 245–254. ACM Press, 1985. ISBN 0897911660. doi:10.1145/325334.325242.

[Shrinivasan and Gotz, 2009] Y. Shrinivasan and D. Gotz. Connecting the dots in visual analysis. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST '09)*. IEEE Computer Society Press, 2009. ISBN 1424452835. doi:10.1109/VAST.2009.5333023.

[Shrinivasan and van Wijk, 2008] Y. B. Shrinivasan and J. J. van Wijk. Supporting the analytical reasoning process in information visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*, pp. 1237–1246. ACM Press, 2008. ISBN 1605580111. doi:10.1145/1357054.1357247.

[Stary, 2000] C. Stary. TADEUS: seamless development of task-based and user-oriented interfaces. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 30, no. 5, pp. 509–525, 2000. doi:10.1109/3468.867859.

[Stasko and Zhang, 2000] J. Stasko and E. Zhang. Focus+Context display and navigation techniques for enhancing radial, Space-Filling hierarchy visualizations. In *Proceedings of the IEEE Symposium on Information Vizualization (InfoVis '00)*, pp. 57–65. IEEE Computer Society Press, 2000. ISBN 0769508049. doi:10.1109/INFVIS.2000.885091.

[Stolte et al., 2002] C. Stolte, D. Tang, and P. Hanrahan. Polaris: a system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 52–65, 2002. doi:10.1109/2945.981851.

[Streit, 2007] M. Streit. Metabolic pathway visualization using gene-expression data, 2007. Master thesis, Graz University of Technology.

[Streit et al., 2008] M. Streit, M. Kalkusch, K. Kashofer, and D. Schmalstieg. Navigation and exploration of interconnected pathways. *Computer Graphics Forum (EuroVis '08)*, vol. 27, pp. 951–958, 2008. doi:10.1111/j.1467-8659.2008.01229.x.

[Streit et al., 2007] M. Streit, M. Kalkusch, and D. Schmalstieg. Interactive visualization of metabolic pathways. In *Poster Compendium of the IEEE Conference on Visualization (Vis '07)*. IEEE Computer Society Press, 2007.

[Streit et al., 2010] M. Streit, A. Lex, H. Doleisch, and D. Schmalstieg. Does software engineering pay off for research? lessons learned from the caleydo project. In *Poster Compendium of the Eurographics Workshop on Visual Computing for Biomedicine (VCBM '10)*. Eurographics, 2010.

[Streit et al., 2009a] M. Streit, A. Lex, M. Kalkusch, K. Zatloukal, and D. Schmalstieg. Caleydo: Connecting pathways and gene expression. *Bioinformatics*, vol. 25, no. 20, pp. 2760–2761, 2009a. doi:10.1093/bioinformatics/btp432.

[Streit et al., 2009b] M. Streit, A. Lex, H. Müller, and D. Schmalstieg. Gaze-Based interaction for information visualization. In *Proceedings of the Conference on Computer Graphics and Visualization and Image Processing (CGVCVIP '09)*, 2009b.

[Streit et al., 2011] M. Streit, H. Schulz, A. Lex, D. Schmalstieg, and H. Schumann. Model-Driven design for the visual analysis of heterogeneous data. *To appear in: IEEE Transactions on Visualization and Computer Graphics*, 2011.

[Streit et al., 2009c] M. Streit, H. Schulz, D. Schmalstieg, and H. Schumann. Towards Multi-User Multi-Level interaction. In *Technical Report LMU-MI-2010-2, Ludwig Maximilias University Munich: Proceedings of the Workshop on Collaborative Visualization on Interactive Surfaces (part of VisWeek '09)*, pp. 5–8, 2009c. ISSN 1862-5207.

[Suderman and Hallett, 2007] M. Suderman and M. Hallett. Tools for visually exploring biological networks. *Bioinformatics*, vol. 23, no. 20, pp. 2651 –2659, 2007. doi:10.1093/bioinformatics/btm401.

[Thiede et al., 2009] C. Thiede, C. Tominski, and H. Schumann. Service-Oriented information visualization for smart environments. In *Proceedings of the Conference* *on Information Visualisation (IV '09)*. IEEE Computer Society Press, 2009. ISBN 0769537337. doi:10.1109/IV.2009.54.

[Thomas and Kielman, 2009] J. Thomas and J. Kielman. Challenges for visual analytics. *Information Visualization*, vol. 8, no. 4, pp. 309–314, 2009. doi:10.1057/ivs.2009.26.

[Thomas and Cook, 2005] J. J. Thomas and K. A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Ctr, 2005. ISBN 0769523234.

[Tobiasz et al., 2009] M. Tobiasz, P. Isenberg, and S. Carpendale. Lark: Coordinating co-located collaboration with information visualization. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '09)*, vol. 15, no. 6, pp. 1065–1072, 2009. doi:10.1109/TVCG.2009.162.

[Tobii Technology, 2010] Tobii Technology. Tobii eye tracking - an introduction to eye tracking and tobii eye tracker. Tech. rep., 2010.

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

[Toyoda et al., 2003] T. Toyoda, Y. Mochizuki, and A. Konagaya. GSCope: a clipped fisheye viewer effective for highly complicated biomolecular network graphs. *Bioinformatics*, vol. 19, no. 3, pp. 437–438, 2003. doi:10.1093/bioinformatics/btg001.

[Tsay et al., 2010] J. Tsay, B. Wu, and Y. Jeng. Hierarchically organized layout for visualization of biochemical pathways. *Artificial Intelligence in Medicine*, vol. 48, no. 2-3, pp. 107–117, 2010. doi:10.1016/j.artmed.2009.06.002.

[Tse et al., 2004] E. Tse, J. Histon, S. D. Scott, and S. Greenberg. Avoiding interference: how people use spatial separation and partitioning in SDG workspaces. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW '04)*, pp. 252–261. ACM Press, 2004. ISBN 1581138105. doi:10.1145/1031607.1031647.

[Tufte, 1983] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, Coneeticut, second edn., 1983.

[Unger, 2010] A. Unger. *Visual Support for the Modeling and Simulation of Cell Biological Processes*. Ph.D. thesis, University of Rostock, 2010.

[Unger and Schumann, 2009] A. Unger and H. Schumann. Visual support for the understanding of simulation processes. In *Proceeding of the IEEE Symposium on Pacific Visualization (PacificVis '09)*, pp. 57–64, 2009. ISBN 1424444045. doi:10.1109/PACIFICVIS.2009.4906838.

[van Wijk, 2006] J. J. van Wijk. Bridging the gaps. *IEEE Computer Graphics and Applications*, vol. 26, no. 6, pp. 6–9, 2006. doi:10.1109/MCG.2006.120.

[Vertegaal, 2008] R. Vertegaal. A fitts law comparison of eye tracking and manual input in the selection of visual targets. In *Proceedings of the Conference on Multimodal Interfaces (ICMI '08)*, pp. 241–248. ACM Press, 2008. ISBN 1605581989. doi:10.1145/1452392.1452443.

[Viau et al., 2010] C. Viau, M. J. McGuffin, Y. Chiricota, and I. Jurisica. The FlowVizMenu and parallel scatterplot matrix: Hybrid multidimensional visualizations for network exploration. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '10)*, vol. 16, no. 6, pp. 1100–1108, 2010. doi:10.1109/TVCG.2010.205.

[Waldner et al., 2009] M. Waldner, A. Lex, M. Streit, and D. Schmalstieg. Design considerations for collaborative information workspaces in Multi-Display environments. In *Technical Report LMU-MI-2010-2, Ludwig Maximilias University Munich: Proceedings of the Workshop on Collaborative Visualization on Interactive Surfaces (part of VisWeek '09)*, pp. 36–39, 2009. ISSN 1862-5207.

[Waldner et al., 2008] M. Waldner, C. Pirchheim, M. Streit, and D. Schmalstieg. Overcoming display boundaries for multiple view visualization. In *Poster Compendium of the Workshop on Giga-Pixel Displays & Visual Analytics (GIANT '08)*, 2008.

[Waldner et al., 2010] M. Waldner, W. Puff, A. Lex, M. Streit, and D. Schmalstieg. Visual links across applications. In *Proceedings of the Conference on Graphics Interface (GI '10)*, pp. 129–136. Canadian Human-Computer Communications Society, 2010. ISBN 1568817125.

[Waldner and Schmalstieg, 2011] M. Waldner and D. Schmalstieg. Collaborative information linking: Bridging knowledge gaps between users by linking across applications. In *Proceeding of the IEEE Symposium on Pacific Visualization (PacificVis '11)*, pp. 115–122. IEEE Computer Society Press, 2011.

[Wang et al., 2009] X. Wang, D. H. Jeong, W. Dou, S. Lee, W. Ribarsky, and R. Chang. Defining and applying knowledge conversion processes to a visual analytics system. *Computers and Graphics*, vol. 33, no. 5, pp. 616–623, 2009. doi:10.1016/j.cag.2009.06.004.

[Ward, 1994] M. O. Ward. XmdvTool: integrating multiple methods for visualizing multivariate data. In *Proceedings of the IEEE Conference on Visualization (Vis '94)*, pp. 326–333. IEEE Computer Society Press, 1994. ISBN 0780325214. doi:10.1109/VISUAL.1994.346302.

[Ware, 2004] C. Ware. *Information visualization : perception for design*. Morgan Kaufman, San Francisco CA, second edn., 2004. ISBN 1558608191.

[Weinstein, 2008] J. N. Weinstein. A postgenomic visual icon. *Science*, vol. 319, no. 5871, pp. 1772–1773, 2008. doi:10.1126/science.1151888.

[Westenberg et al., 2010] M. A. Westenberg, J. B. Roerdink, O. P. Kuipers, and S. A. van Hijum. SpotXplore: a cytoscape plugin for visual exploration of hotspot expression in gene regulatory networks. *Bioinformatics*, vol. 26, no. 22, pp. 2922 –2923, 2010. doi:10.1093/bioinformatics/btq535.

[Wilkinson, 2009] L. Wilkinson. The history of the cluster heat map. *The American Statistician*, vol. 63, no. 2, pp. 179–184, 2009. doi:10.1198/tas.2009.0033.

[Willett et al., 2007] W. Willett, J. Heer, and M. Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '07)*, vol. 13, no. 6, pp. 1129–1136, 2007. doi:10.1109/TVCG.2007.70589.

[Wilson and Bergeron, 1999] R. M. Wilson and R. D. Bergeron. Dynamic hierarchy specification and visualization. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '99)*, pp. 65–72. IEEE Computer Society Press, 1999. ISBN 0769504310. doi:10.1109/INFVIS.1999.801859.

[Yang et al., 2003] J. Yang, M. O. Ward, E. A. Rundensteiner, and S. Huang. Visual hierarchical dimension reduction for exploration of high dimensional datasets. In *Proceedings of the Eurographics Symposium on Data Visualisation (VISSYM '03)*, p. 19–28. Eurographics, 2003. ISBN 1581136986.

[Yang et al., 2006] Y. Yang, E. S. Wurtele, C. Cruz-Neira, and J. A. Dickerson. Hierarchical visualization of metabolic networks using virtual reality. In *Proceedings on Virtual Reality Continuum and its Applications (VRCIA '06)*, pp. 377–381. ACM Press, 2006. ISBN 1595933247. doi:10.1145/1128923.1128992.

[Yost et al., 2007] B. Yost, Y. Haciahmetoglu, and C. North. Beyond visual acuity: the perceptual scalability of information visualizations for large displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*, pp. 101–110. ACM Press, 2007. ISBN 1595935939. doi:10.1145/1240624.1240639.

Curriculum Vitae

Personal Data

Name	Dipl.-Ing. *Marc Streit* Bakk.rer.soc.oec.
Born	February 26, 1983 in Klagenfurt, Austria
Nationality	Austrian

Education

Since Aug. 2009	Associated PhD student in the Research Training Group dIEM oSiRiS, University of Rostock, Germany
July - Aug. 2009	Visiting researcher at the University of Rostock in the Research Training Group dIEM oSiRiS
Since July 2007	Doctoral program in computer science at the Graz University of Technology.
June 2007	Master's degree (Dipl.-Ing.) from the Graz University of Technology *(with highest distinction)*
2006–2007	Master's studies in Information Management at the Graz University of Technology
July 2006	Bachelor's degree (Bakk. rer. soc. oec) from the Graz University of Technology
2002–2006	Bachelor's studies in Software Engineering and Management at the Graz University of Technology
June 2002	Graduation from the HTBLA Mössingerstraße *(with highest distinction)*
1997-2002	Engineering school (Höhere Techn. Bundeslehranstalt) HTBLA Klagenfurt Mössingerstraße
1993–1997	Secondary school (Gymnasium) at the Bundesgymnasium Klagenfurt Mössingerstraße