Graz University of Technology
Institute for Computer Graphics and Vision

Dissertation

Visualization of Multidimensional Data with Applications in Molecular Biology

---

Alexander Lex
Graz, Austria, March 2012

Advisor
Prof. Dr. Dieter Schmalstieg
Graz University of Technology

Co-Advisor
Dr. Nils Gehlenborg
Harvard Medical School

Referee
Prof. Dr. Robert Kosara
University of North Carolina at Charlotte

To Heidi

However, despite understandable celebration of these achievements, sober reflection reveals many challenges ahead.

Mark I. McCarthy et al. on our understanding of the genetic basis of common phenotypes.

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.

Graz, Austria	**March 1, 2012**	
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.

Graz, Österreich	**1. März, 2012**	
Ort	Datum	Unterschrift

Abstract

Visualization is important when analyzing multidimensional datasets, since it can help humans discover and understand complex relationships in data. Whereas analyzing large individual datasets is both important and difficult, many problems can only be solved when considering multiple datasets simultaneously. This dissertation introduces novel visualization techniques that can be employed for both, visualizing individual datasets and visualizing relationships among multiple datasets alike. The concept is based on stratifying (dividing) datasets into homogeneous subsets, which can then be visualized individually. The relationships lost due to the division are re-introduced by drawing visual links between the subsets. Conceptually it is irrelevant whether the subsets are from one or from multiple datasets, which makes a seamless integration of multiple, cross-referenced datasets possible. The subsets can be visualized in multiple forms. Multiform visualization gives users the freedom to choose the visualization technique most suitable for the data type, the degree of homogeneity, the level of detail, and the current task – for each of the subsets individually. The division of datasets also makes focus and context, as well as drill-down techniques straightforward to realize. A set of interaction techniques enable seamless transition from a global overview down to details on individual data items.

While the visualization techniques introduced in this thesis are generally applicable, they are designed to support researchers working in molecular biology. Specifically, we support collaborators in two different scenarios: in uncovering the genetic causes of steatohepatitis, a precursory disease to cirrhosis of the liver, and in analyzing cancer subtypes. We evaluated our methods with cases studies and report on how investigators reproduced known findings and discovered new insights with the introduced visualization techniques.

In addition to discussing the analysis of multidimensional datasets, we also describe an integrative approach to analyze general heterogeneous datasets. We show how modeling of the analysis setup can be employed to support users. Finally, we introduce cross-application and context-preserving visual links, which can be used for highlighting in heterogeneous datasets.

Keywords. Information Visualization; Visual Analytics; Multidimensional Data Analysis; Heterogeneous Data Analysis; Multiform Visualization; Biology Visualization; Molecular Biology Visualization; Visual Links.

Kurzfassung

Der Einsatz von Visualisierung bei der Analyse hochdimensionaler Daten ist wichtig, um Menschen beim Erkennen komplexer Zusammenhänge zu unterstützen. Obwohl die Analyse von einzelnen großen Datensätzen sowohl komplex als auch von großer Bedeutung ist, können viele Probleme nur unter der gleichzeitigen Berücksichtigung mehrerer Datensätze gelöst werden. In dieser Dissertation werden neuartige Visualisierungstechniken eingeführt, die sowohl für die Analyse einzelner als auch für die simultane Analyse mehrerer Datensätze verwendet werden können. Das Konzept basiert auf einer Stratifizierung (Teilung) der Datensätze in homogene Teilmengen, die dann individuell dargestellt werden können. Die Beziehungen, die durch die Teilung verloren gehen, werden durch *Visual Links* wieder hergestellt. Da es konzeptionell unerheblich ist, ob einzelne Teilmengen aus dem selben oder aus verschiedenen Datensätzen stammen, können mehrere Datensätze nahtlos integriert werden. Die Teilmengen können dabei mit verschiedenen Visualisierungstechniken dargestellt werden. Dies erlaubt Benutzern oder Benutzerinnen, die richtige Visualisierungstechnik für den Datentyp, den Grad der Homogenität, den Detailgrad der Daten und die aktuellen Aufgabe auszuwählen. Durch die Teilung sind auch *Focus and Context* sowie *Drill-Down* Techniken einfach zu realisieren.

Obwohl die in dieser Arbeit eingeführten Visualisierungstechniken generell anwendbar sind, wurden Sie mit dem Ziel, ForscherInnen im Bereich der Molekularen Biologie bei ihrer Arbeit zu unterstützen, entwickelt. Im Speziellen werden zwei Anwendungsfälle aufgegriffen: die Untersuchung der genetischen Ursachen von Steatohepatitis, einer Krankheit, die häufig Leberzirrhose zur Folge hat, sowie die Analyse von Subtypen von Krebs. Die eingeführten Visualisierungstechniken wurden anhand von Fallstudien evaluiert. Dabei konnten die Analysten sowohl bekanntes Wissen reproduzieren als auch neue Einsichten in die Daten gewinnen.

Zusätzlich zu den Analysemethoden für multidimensionale Datensätze wird auch die integrative Analyse allgemeiner heterogener Datensätze behandelt. Wir zeigen, wie, basierend auf einem Modell der Daten, Benutzer und Benutzerinnen bei der Analyse unterstützt werden können. Abschließend werden applikationsübergreifende und kontexterhaltende *Visual Links* eingeführt, die für *Highlighting* in heterogenen Datensätzen verwendet werden können.

Acknowledgments

Pursuing a PhD does not have a lot in common with being an undergraduate student, and while in my case it was a paying profession, I believe it does not have a lot in common with a regular job either. It is an all-in job with long hours and weekends spent in the lab. But it also gives you more freedom to pursue your ideas than any other job does, and it gives you the chance to travel around the world and meet many interesting and inspiring people. Research nowadays is very much team-oriented and consequently, there are a lot of people that contributed directly or indirectly to this thesis. Here I want to give them proper credit.

First of all, I am very grateful to my advisor, Prof. Dieter Schmalstieg, for teaching me how to do research and for always having an open mind for every idea that came along. I really appreciate the atmosphere at the institute. Likewise, many thanks go to my co-advisor Nils Gehlenborg, who supervised me during and since my research visit at Harvard Medical School. Nils is an amazing researcher at the interface of biology, bioinformatics and visualization and I can say without reservation that I learned a lot from him. Many thanks also to Prof. Robert Kosara, who agreed to be a referee for this thesis.

The person who contributed most to this thesis is without a doubt my friend and colleague Marc Streit. We have been working together since our first day at the university ten years ago, continued our teamwork in student projects and finally in creating Caleydo and writing numerous papers. Being able to work with someone who is also a close friend is a great experience. Thank you Marc for everything on and off the job.

Hans-Jörg Schulz, from the University of Rostock, taught me that it is the concept, and nothing but the concept, that matters – if there is a good concept, there will be an application for it somewhere. Thank you, Hansi, for all the great discussions and the joined work.

I am very thankful to many other colleagues and co-authors, specifically Manuela Waldner, Christian Partl, Markus Steinberger, Prof. Heidrun Schumann, Clemens Holzhüter, Werner Puff, Michael Kalkusch, Bernhard Schlegel, Thomas Geymayer, Ernst Kruijff and Helmut Doleisch. Special thanks go to Prof. Peter Park for enabling me to work at Harvard Medical School.

Without our biological collaborators much of the work would not have been possible. I hereby acknowledge Karl Kashofer, Prof. Kurt Zatloukal and Martin Asslaber from the Medical University of Graz; Prof. Heinz Redl, Gudrun Schmidt-Gann, Katharina Schmid and Monika Schuller from the Ludwig Boltzmann Institute for Experimental and Clinical Traumatology; Ian Watson and Steven Quayle from the Dana-Farber Cancer Institute; as well as Aaron McKenna, Andrew Cherniak and Michael Noble from the Broad Institute

of MIT and Harvard.

Day-to-day interaction and inspiring discussions contribute to the great climate at the Institute for Computer Graphics and Vision. Many thanks to my co-workers Denis Kalkofen, Bernhard Kainz, Prof. Gerhard Reitmayer, Eduardo Veas, Prof. Horst Bischof, Mark Dokter, Christina Fuchs, Renate Hönel, Markus Tatzgern, Stefanie Zollman, Tobias Langlotz, Markus Grabner, Erick Mendez, Gerhard Schall and many more.

What remains is to thank the people most important in my life, who helped me get to where I am now. No one deserves more credit than my wonderful wife Heidi. Thank you for your love, your support and understanding, for your amazing proof-reading skills and for spending your life with me. Thanks to my parents, Edith and Georg, for your support, for believing in me and giving me an awesome childhood. Thanks to my brother Richard for all the fun we had and still have, to his partner Michaela and their son Niko, my godson. I would also like to thank my grandmothers Marianne and Margarethe. To my extended family, with whom I am very close: thank you all for everything.

Last but not least it is important to me to mention my friends: thank you Armin, for defeating me in squash all the time, Bernie, for riding the best slopes in the Alps with me, Robert, for drinking whiskey and traveling with me, and thanks to all the other guys and girls!

I acknowledge funding from the Austrian Science Fund through the Caleydoplex (P22902) and the VIPEM (L427-N15) projects, as well as from the Austrian Research Promotion Agency trough the inGeneious (385567) and the Genoptikum (813398) projects.

Contents

List of Figures

Chapter 1

Introduction

Contents

High-dimensional, tabular data is collected and analyzed in many application domains. Examples range from businesses, where transactions, customers, products, etc., are stored in relational databases, to the social sciences, where demographic data is collected, to scientific domains, such as molecular biology, where states and activities of genes, proteins, etc., are measured. This prevalence of multidimensional data is owed to two properties. First, multidimensional data fits naturally to many real world observations or calculations where data is collected for a set of attributes of a dimension. Examples are a *given name*, *surname* and *age* of the dimension *person*; or *speed*, *acceleration* and *position* of a *vehicle* at a particular time; as well as the *expression* of multiple *genes* of one biological *sample*. Second, multidimensional data has many desirable technical properties. It is structured, and therefore easy to store, manipulate, parse, process and compress. There are many standardized formats and plenty of tools to manipulate, analyze and plot multidimensional data. Also, all different scales of data, qualitative as well as quantitative, are suitable to be stored in tabular form.

Information visualization is the scientific discipline dealing with the visual display of abstract data. Abstract data in this sense refers to data that is not primarily spatially referenced, as, for example, medical imaging data is. Information visualization techniques therefore are typically free to employ spatial encoding, i.e., they can use the position and size of graphical marks to display the data. The rational of visually displaying information is that humans can better process, understand, analyze and find relationships in appropriately encoded data compared to looking at raw data. Information visualization is a medium for humans to understand complex data, extract information and ultimately gain

knowledge and make decisions. The field of **visual analytics** is strongly related to information visualization, but takes a more holistic approach. While information visualization is primarily concerned with visual encoding, visual analytics also considers issues of data mining, data management, infrastructure, etc. [98, p. 12].

It is very hard for humans to extract information from even moderately sized tables. To compare two data tables of 100 rows by 100 columns, for example, 20.000 successive instants of perceptions are necessary. Reading the same information can be instantaneous when the data is displayed graphically [15, p. 3]. Combined with the aforementioned prevalence of tabular data, it is not surprising that many visualization techniques for multidimensional data have been developed. The main challenge when visualizing multidimensional data is scale: visualizing a table of 20 by 20 is straight-forward, but visualizing a table with hundreds of dimensions and thousands of records is not trivial.

Datasets of this scale, however, are very common in molecular biology. The field of molecular biology deals with uncovering the function of genes and other genetic or epigenetic processes. Understanding biomolecular data has a profound impact on mankind's knowledge about fundamental questions, such as how cells work, but also has very practical applications, such as which drug is most suitable for a particular illness.

This thesis deals primarily with the visualization and visual analysis of biomolecular, multidimensional data. The central paradigm of this thesis is a **divide and conquer approach** for multidimensional data. We postulate the following three hypotheses:

I **Division Hypothesis** – Dividing (stratifying) inhomogeneous, multidimensional datasets into homogeneous groups allows analytical algorithms to create better results, thereby making the subgroups more meaningful. This, however, obscures the relationships among the groups in the dataset. We hypothesize that, given the right choice of visual encoding, it is possible to re-introduce the connections lost due to the stratification. The division hypothesis is the subject of Chapter 5.

II **Multiform Hypothesis** – Given a dataset that is divided into homogeneous groups, we hypothesize that it is beneficial to be able to let users choose the visual encoding for each of the groups individually, i.e., represent the data in one of multiple forms. We argue that choosing different levels of abstraction, different visual encodings and visualization techniques, as well as different levels of detail, enables analysts to choose the right level of abstraction and the right visual encoding for the degree of homogeneity of a group, for the task of a user, and for the size of the dataset. The multiform hypothesis is elaborated in Chapter 6.

III **Cross-Referenced Data Hypothesis** – We postulate that the stratification, connection, and multiform techniques can not only be used to analyze one inhomogeneous dataset, but are equally applicable for integrating multiple cross-referenced datasets. Cross-referenced datasets are multiple heterogeneous datasets that have certain restrictions on the structure of the data. Most importantly, integrating multiple datasets can be used to judge the validity of stratifications and refine them when necessary. Also, we argue that it is of value to apply a stratification derived from one dataset to other datasets. The integration of multiple datasets is the topic of Chapter 7.

A secondary topic of this dissertation is the integration of general heterogeneous datasets, discussed in Chapter 8. We distinguish between structured and unstructured heterogeneous datasets. For the former we present a model-based approach for orientation and guidance in heterogeneous data spaces. Unstructured heterogeneous datasets are not in the scope of this thesis. However, we introduce methods for visual linking, which can also be used in unstructured heterogeneous data analysis scenarios.

1.1 Structure of this Thesis

In the remainder of this chapter we first discuss the properties of the types of data considered in this thesis. We give a formal definition of multidimensional data and discuss homogeneity/inhomogeneity of individual multidimensional datasets, followed by an analysis and classification of heterogeneous data. We then define terms we use throughout this thesis, followed an elaboration of the contributions of this work. The chapter is concluded with a declaration of collaborations and a listing of the publications which this thesis is based on.

The examples, use cases, and applications in this thesis are mainly from the domain of molecular biology. To brief the reader, we give an introduction into molecular biology in Chapter 2. Before going into detail about the techniques employed to satisfy the hypothesis, we discuss the related work in Chapter 3, followed by an introduction to Caleydo in Chapter 4. Caleydo is the visualization framework that most of the techniques presented are part of. Chapters 5-7 form the core of this thesis and discuss the methods addressing the hypotheses. General heterogeneous data is covered in Chapter 8, before the thesis is concluded in Chapter 9.

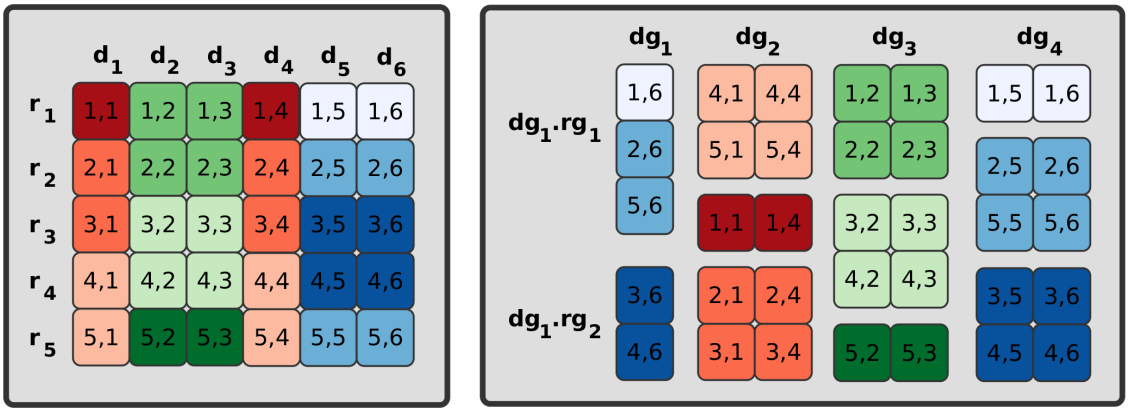
1.2 Data Preliminaries

Before we discuss the analysis and visualization of data, it is necessary to define the form and the scope of data considered in this thesis. The most basic notion of data is a **data item**, which describes an atomic property, such as a single number, a single word or a single relationship. We define a **dataset** as a discrete collection of data items, irrespective of a structure or data type. Typically, but not necessarily, the data items in a dataset have some common semantics, such as being from the same source, or describing related observations or measurements. Datasets may be unstructured, containing, for example, free text, or structured, as, for example, graphs or multidimensional datasets are.

We will begin by discussing multidimensional datasets, which are the most important type of dataset with respect to this thesis, its stratification into groups, and aspects of homogeneity. We then discuss heterogeneous data and introduce two classes of heterogeneity.

1.2.1 Multidimensional Data and Grouping

Colloquially, multidimensional data can be understood as tabular data, containing columns (dimensions) and rows (records). Multidimensional datasets typically have an identifier



(a) (b)

Figure 1.1: Example of a multidimensional dataset with inhomogeneities and of an introduced grouping. (a) A table with six dimensions and five records. One possible aspect of homogeneity of dimensions is coded in the hue, a possible homogeneity of the records is encoded using saturation. (b) A grouping of dimensions and records. Dimensions can be part of more than one group. For example, d_6 is part of both dg_1 as well as dg_4. Also, dg_1 illustrates that the homogeneity of records is not pre-determined. Here the homogeneity indicated by the hue is not reflected in the grouping.

(a key) for both, records and dimensions, which provides the semantics of the records and dimensions. Multidimensional data analysis refers to the study of multiple variables, i.e., dimensions or records. These variables can be classified into dependent and independent variables, where the dependent variable can be described as a function of the independent variables [13]. Some sources distinguish between the study of multiple independent variables, which they refer to as multidimensional data analysis, and the study of multiple dependent variables which they call multi-variate data analysis [13]. However, in visualization this strong mathematical definition is typically relaxed to a broader definition of multiple variables, regardless of the dependent-independent relationship [206]. We will adhere to the latter convention.

We formally define a **multidimensional dataset** as a matrix $M = \{v_{ij} | 1 \leq i \leq n, 1 \leq j \leq m\}$ where the columns $D = \{d_1, ..., d_m\}$ are the dimensions and the rows $R = \{r_1, ..., r_n\}$ contain the data records. Each matrix cell v_{ij} is a value in row r_i of the dimension d_j. This is illustrated in Figure 1.1(a).

As discussed before, stratifying (splitting, grouping) the dataset into homogeneous **groups** has many advantages. We define a grouping of dimensions $DG = \{dg_1, ..., dg_u | dg \in \mathcal{P}(D)\}$ ($\mathcal{P}(D)$ denotes the powerset of D) where each dimension can be assigned to multiple groups. Figure 1.1(b) shows an example where d_6 is assigned to dg_1 and dg_4. For each dg_i in G we create a set of record groups $RG_i = \{rg_1, ..., rg_v\}$ which contains the records (restricted to the dimensions in dg_i), where r_j can only be part of one record group. We denote an individual group defined over both dimensions and records as $dg_i.rg_j$.

Inhomogeneity or homogeneity is a fundamental property of many multidimensional datasets. It can be observed on the set of dimensions, as illustrated using different hues in Figure 1.1(a), and on the set of records, as shown using different levels of saturation in Figure 1.1(a). We distinguish inhomogeneity from the slightly different notion of data diversity. The latter defines high diversity as an even distribution of values [141], which is a property of a rather homogeneous dataset. We use the term **inhomogeneity** to refer to differences within a single dataset, whereas we use **heterogeneity** to refer to multiple datasets. In principle, three different sources of inhomogeneity within a dataset can be discriminated:

- **semantics** – of different meanings: the more unrelated the data is in terms of meaning, the more inhomogeneous it is,
- **characteristics** – of different types: the more the data types and value ranges vary, the more inhomogeneous they are,
- **statistics** – of different behaviors or distributions: the less evenly the data is distributed over a value range, the more inhomogeneous it is.

While characteristics and statistics inhomogeneities are inherent in the data, semantics have to be specified separately. This can be done, for instance, with an ontology or manually by a user. Also, the different sources of inhomogeneity are not mutually exclusive. It is common that several sources of inhomogeneity are present at the same time in a dataset. The relevance of the three levels of inhomogeneity for dimensions and records is explained in the following.

Inhomogeneous Dimensions: In terms of **semantics**, inhomogeneities can often be found among dimensions with no inherent connection on the level of what they are meant to encode. For example, the columns *first name* and *last name* belong together because they compose the information *name* and the columns *street*, *city*, and *zip code* form the information *address*. However, *first name* and *zip code* are semantically unrelated. Such groupings are not obvious and have to be specified by the user employing common knowledge, or through meta-data.

The dimensions' **characteristics** detail a dimension's type, of which we distinguish four: **bounded numerical**, **unbounded numerical**, **exclusive categorical**, and **inclusive categorical**. An example of inhomogeneity between different dimensions would be two bounded numerical types with very different bounds given, e.g., $[0 \dots 1]$ and $[10^6 \dots 10^7]$, which are hard to analyze together, numerically or visually. The same is the case for dimensions of exclusive categorical data, such as sex, which is an either-or category, and inclusive categorical data, such as professional memberships in, for example, IEEE, ACM and Eurographics. Such characteristics can be interactively defined [134] or given in a standardized format such as qnch*.

Statistics, in contrast, are derived directly from the data using methods such as correspondence analysis, which determine related dimensions that are likely to belong

*http://qnch.org

together because of correlated values.

Inhomogeneous Records: Similar to dimensions, records can be affected by **semantic** inhomogeneity, which is given by external knowledge. This occurs frequently for categorical values; e.g., the professions *high school teacher* and *university professor* relate more to one another than to *restaurant chef*, because both belong to the educational sector. Again, this knowledge is not present in the data itself and has to be provided by the user or through an ontology.

Inhomogeneities stemming from a record's **characteristics**, can be, for example, missing or undefined values. Undefined values are, for example, those that are outside of a dimension bound given by meta-data. Observation of these inhomogeneities is important; these records need to be set aside because they cannot be analyzed together with the regular records. Their communication is nevertheless important for the analysis [37].

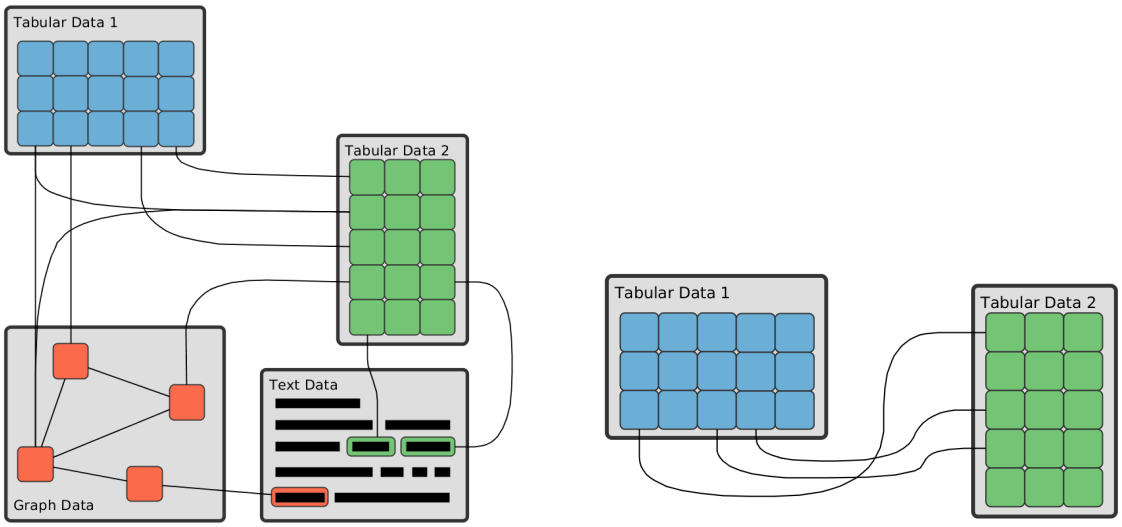
Inhomogeneities uncovered via **statistical** methods such as clustering occur when the data records are distributed unevenly and thus form clusters at certain points or intervals of the overall value range. Data records that have been assigned to the same cluster are thus more alike and form a more homogeneous group of data with respect to the similarity measure used for clustering.

1.2.2 Heterogeneous Datasets

The analysis of data from multiple, heterogeneous sources has been recognized as a major challenge of visual analytics (e.g., in the European research agenda for visual analytics by Keim et al. [98, p. 19], or its American counterpart by Thomas and Cook [183, p. 100]). Up to this point, we have mainly discussed multidimensional data in a single dataset. Nevertheless, many methods discussed in this thesis are equally applicable to multiple, heterogeneous datasets. We distinguish between two types of heterogeneous datasets:

General heterogeneous datasets: We consider any set of datasets that can not be trivially joined into a single, semantically meaningful dataset as heterogeneous datasets. Two datasets that do not share any common properties, data items or identifiers are general heterogeneous dataset. However, to be of value in an integrated analysis, there has to be some relationship between the datasets. These relationships do not have to adhere to any convention. Examples for such relationships are a textual dataset, which contains an identifier from a multidimensional dataset, or a graph, where a node attribute can be related to a dimension, record, or data item of a multidimensional dataset. Figure 1.2(a) illustrates these two examples. Other examples are multimedia, imaging, volume or other spatial data, which may be referenced to an identifier of another datasets. General heterogeneous datasets cannot be easily subjected to the divide and conquer approach postulated. They are, nevertheless, often crucial for an analysis, as they can contain meta-information, or give a broader context. Consequently it is essential to integrate general heterogeneous datasets in an advanced analysis scenario. How this can be done is the topic of Chapter 8.

Cross-referenced Datasets: Cross-referenced datasets are a subclass of general heterogeneous datasets, with restrictions on the type of the dataset and the relationships between



(a) (b)

Figure 1.2: Types of heterogeneous datasets. (a) References between two **general heterogeneous datasets**, `Text Data` and `Graph Data`, and two multidimensional datasets. `Text Data` contains references to one record and one dimension of `Tabular Data 2` (b) Relationships between two **cross-referenced datasets**. The columns of `Tabular Data 1` can be referenced to the rows of `Tabular Data 2`. It is not required that every variable in the datasets match.

.

the datasets. Cross-referenced datasets are restricted to multidimensional datasets that share a type of identifier, or have types of identifiers that can be mapped to each other. Shared types of identifier may exist for either dimensions, or records, or both. These *shared keys* are common in relational database schemes. An example for a cross-referenced dataset is a table with test-results for patients and another table with demographic data. When datasets are cross-referenced, it is possible to retrieve the variables of one entry and then retrieve additional variables from the cross-referenced dataset. Often such a resolution is needed to provide context in an analysis. An example is the observation of hormone levels in a patient, which can only be judged in the context of the patient's age. Figure 1.2(b) illustrates a simple case where the dimensions of one dataset corresponds to the records of another one. As tabular datasets can easily be transposed, such a dimension-record relationship can be changed to a dimension-dimension or record-record relationship. We do not require *referential integrity* – there may be no reference for some entries in the other dataset. This is important, as real-life datasets are often incomplete. Cross-referenced datasets can be integrated into a single dataset. This, however, has several disadvantages. It might result in hybrid types of identifiers for either dimensions or records. Also, parts of the matrix may be empty. We will discuss the implications for analyzing cross-referenced datasets as part of the considerations for Hypothesis III in Chapter 7.

1.3 Vocabulary

Based on the previously introduced concepts we define the vocabulary used in the remainder of this thesis. We use **entry** to refer to either dimensions or records, when a concept is equally applicable to both. We already established a group of dimensions as **dimension group**, and a group of records within these dimensions as **record group**. The visual representations of a record group are called **brick**. The visual equivalent of the dimension group is the **column**, which is a stacking of bricks. We distinguish those terms because bricks and columns are a visual entity which can also display other forms of data.

The process of grouping dimension or records can also be thought of as dividing or splitting the dataset. We do not semantically distinguish between grouping and dividing, as it is only a matter of perspective. We refer to the division of dimension or records as **stratification** (in reference to the term *stratified sampling* used in statistics) to better reflect the intention behind the process: to divide the data into homogeneous groups.

1.4 Contributions

For each of the aforementioned hypotheses a visualization technique is proposed. We demonstrate the validity of the solutions through comprehensive case studies.

To address the division hypothesis, we developed the **Matchmaker** technique. We demonstrate how to manually or automatically stratify the dimensions into dimension groups so that clustering on the records (automatic stratification) can find meaningful partitionings. The conquer step is realized by stacking bricks of the same dimension group on top of each other, i.e., the association of dimensions is retained using position. Matchmaker uses heatmaps to encode data in a brick. The assignment of records to the bricks, the order of records within bricks and the overall order of bricks are chosen in meaningful ways. The relationship between columns is retained by employing connectedness between the bricks: we use visual links to connect related entities. Drill-dow techniques guarantee a seamless transition from a global overview down to individual data items. The validity of Hypothesis I, and consequently the utility of the Matchmaker technique is demonstrated in a case study detailing a complex micro-array analysis scenario. A second use case is presented that shows that the technique can also be used to judge the quality of clustering algorithms.

Based on the divide and conquer strategy of the Matchmaker technique, we developed **VisBricks** to satisfy the multiform hypothesis. VisBricks generalizes the basic ideas of Matchmaker by introducing different classes of bricks. We demonstrate that different views are suitable for different types of data and tasks. We also introduce a number of advanced interaction techniques to accommodate the wider range of possible arrangements compared to Matchmaker. We take up the microarray analysis scenario used for Matchmaker to validate the VisBricks approach and the multiform hypothesis.

We further generalize the Matchmaker and VisBricks techniques to integrate multiple, cross-referenced datasets in the **StratomeX** technique. As with multiple, cross-referenced datasets the setup of the visualization becomes challenging by itself, we present a meta-visualization to aid investigators in this task. We introduce two new types of columns that can be used to integrate other types of data. We give an extensive example of a compre-

hensive use case, where we use StratomeX to analyze cancer subtypes. We demonstrate the utility of StratomeX (and thereby implicitly of Matchmaker and VisBricks) on three real-life application scenarios, and thereby validate the cross-referenced data hypothesis

Finally, we discuss methods to integrate general heterogeneous data. We present a model for designing heterogeneous data analysis frameworks as well as a prototype based on the model. We connect heterogeneous datasets using visual links, either within an application, but also among multiple independent applications. We report on the results of user studies that were conducted to validate the visual linking strategies.

1.5 Collaboration Statement

Aside from the supervisors of this thesis, **Prof. Dieter Schmalstieg**, **Dr. Nils Gehlenborg**, and **Prof. Robert Kosara**, many colleagues have contributed to the work described in this thesis. In this section, the most important collaborators are mentioned, including a statement of their contributions.

Dr. Marc Streit was the closest collaborator and was involved in all but one publications, with significant contributions to all of them. He contributed on a conceptual level, to the implementation and to the write-ups. He has also been a core developer of the Caleydo Visualization Framework. He is the principle author of the paper on model-driven design for heterogeneous data analysis [179], from which Chapter 8 draws.

Dr. Hans-Jörg Schulz is co-author of many publications. For this thesis, his contributions to the multiform visualization [112] (discussed in Chapter 6) are most significant. Together with Marc Streit, he is one of the main authors of the paper on model-driven design for heterogeneous data analysis [179], from which Chapter 8 draws.

Christan Partl contributed, as a master student, to the implementation of all core papers of this thesis [112, 114, 115]. He also contributed as a Caleydo framework developer.

Dr. Manuela Waldner was involved in the publications related to visual linking [174, 197], discussed in Chapter 8, on a conceptual level. She also participated in the implementation and led the user studies.

Markus Steinberger is the developer of the context-preserving visual links approach [174] discussed in Chapter 8, and contributed significantly to all parts of this paper.

Prof. Heidrun Schumann supervised the work on model-driven design for heterogeneous data analysis [179] and on uncertainty visualization in heatmaps [78], which was lead by Clemens Holzhüter.

Other collaborators include **Michael Kalkusch**, who started the Caleydo visualization framework, **Werner Puff**, who contributed to the Caleydo framework development and the across-application visual linking [197], **Bernhard Schlegel**, who, as a master student, was the principal developer of the hierarchical heatmap and the clustering algorithms, **Thomas Geymayer** who, as a master student, was the principal developer of the filter pipeline [57], and **Dr. Ernst Kruijff**, who contributed to the user study and the write-up in the Caleydo overview paper [113].

The biological background, use cases, data, and feedback were provided and contributed by colleagues from the **Institute for Pathology at the Medical University Graz**, most importantly Prof. Kurt Zatloukal, Dr. Karl Kashofer, and Dr. Martin Asslaber; the **Center for Biomedical Informatics at Harvard Medical School**, especially Prof. Peter J. Park; Ian Watson and Steven Quayle from the **Dana-Farber Cancer Institute**; as well as Aaron McKenna, Andrew Cherniak and Michael Noble from the **Broad Institute of MIT and Harvard**; and finally by the **Ludwig Boltzmann Institute for Experimental and Clinical Traumatology**, where Prof. Heinz Redl, Dr. Gudrun Schmidt-Gann, Dr. Katharina Schmid and Monika Schuller collaborated.

1.6 Related Publications

The content of this thesis is based on several publications with many co-authors. This section briefly describes which part of a publication is reflected in which chapter of the thesis and discusses the contribution of this thesis' author.

1.6.1 Primary Publications

The following publications contain the core concepts, ideas, realizations, and case studies presented in this thesis.

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), 2010 [114].
This papers describes the Matchmaker technique. It is the primary source of Chapter 5. Also, parts of the data analysis section in this chapter is based on the material.

A. Lex, H. Schulz, M. Streit, C. Partl, and D. Schmalstieg. *VisBricks: Multiform Visualization of Large, Inhomogeneous Data.* IEEE Transactions on Visualization and Computer Graphics (InfoVis '11), 2011 [112].
This papers describes the VisBricks technique. It is the primary source of Chapter 6. Also, parts of the data analysis in this chapter is based on the material.

A. Lex, M. Streit, H. Schulz, C. Partl, D. Schmalstieg, P. J. Park, and N. Gehlenborg. *StratomeX: Visual Analysis of Large-Scale Heterogeneous Genomics Data for Cancer Subtype Characterization.* Conditionally accepted for: Computer Graphics Forum (EuroVis '12), 2012 [115].
This papers describes the StratomeX technique and contains the cancer subtype analysis use case. It is the primary source of Chapter 7. Also, the biological background in this section draws from the content of the paper.

A. Lex, P. J. Park, and N. Gehlenborg. *Supporting Subtype Characterization through Integrative Visualization of Cancer Genomics Data Sets.* In Proceedings of The Cancer Genome Atlas' 1st Annual Scientific Symposium: Enabling Cancer Research Through TCGA. Washington, D.C., USA, 2011 [111].
This abstract and poster describes an early prototype of the StratomeX technique and was presented at a biological conference to elicit feedback from the community.

1.6.2 Secondary Publications

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), 2010 [113].
This paper describes the hierarchical heatmap and the Bucket technique for view placement in 2.5D space, which are part of Chapter 4. The main idea for the Bucket technique was developed by Marc Streit, otherwise the first two authors contributed equally. Ernst Kruijff was responsible for the design of the user study on the bucket visualization technique.

M. Streit, **A. Lex**, M. Kalkusch, K. Zatloukal, and D. Schmalstieg. *Caleydo: Con- necting pathways and gene expression.* Bioinformatics, 2009 [178].
This abstract describes the Caleydo framework to a bioinformatics community. The content is a subset of the paper listed above.

M. Streit, H. Schulz, **A. Lex**, D. Schmalstieg, and H. Schumann. *Model-Driven Design for the Visual Analysis of Heterogeneous Data.* IEEE Transactions on Visualization and Computer Graphics, 2011 [179].
The concept for this paper was primarily developed by Marc Streit and Hans-Jörg Schulz, with contributions from all other authors. The implementation was done by Marc Streit and the author. It describes a general model for heterogeneous data and a prototype implementation for the model, which is part of Chapter 8.

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), 2010 [197].
The concept for this paper was developed by all authors. Most of the work and implementation was done by Manuela Waldner and Werner Puff. It describes a general, application-spanning framework for visual linking, which is described in Chapter 8.

M. Steinberger, M. Waldner, M. Streit, **A. Lex**, and D. Schmalstieg. *Context-Preserving Visual Links.* IEEE Transactions on Visualization and Computer Graphics (InfoVis '11) 2011 [174].
All authors contributed to the concept for this paper. The technical approach was developed by Markus Steinberger and Manuela Waldner. The realization was primarily executed by Markus Steinberger. The paper describes how to route visual links so they do not disturb a base representation. It also contains a user study on the utility of visual links. The content is part of Chapter 8.

T. Geymayer, **A. Lex**, M. Streit, and D. Schmalstieg. *Visualizing the Effects of Logically Combined Filters.* In Proceedings of the 15th International Conference on Information Visualisation (IV'11), 2011 [57].
This paper describes a visualization technique for filters which are used in preprocessing of datasets, which is described in Chapter 4. The concept was developed by Marc Streit and the author equally, Thomas Geymayer contributed improvements. The implementation was done by Thomas Geymayer, with support from Marc Streit and the author.

C. Holzhüter, **A. Lex**, D. Schmalstieg, H. Schulz, H. Schumann, and M. Streit. *Visualizing uncertainty in biological expression data.* In Proceedings of the SPIE Conference on

Visualization and Data Analysis (VDA '12), 2012 [78].
The concept for the paper was developed by all authors, but the work was lead by Clemens Holzhüter. The implementation was executed by Clemens Holzhüter, Marc Streit and the author. Content of the paper is discussed in Chapter 4.

M. Waldner, **A. Lex**, M. Streit, and D. Schmalstieg. *Design Considerations for Collaborative Information Workspaces in Multi-Display Environments.* Proceedings of the Workshop on Collaborative Visualization on Interactive Surfaces (VisWeek '09), 2009, [196].
This paper describes implications of doing visual analysis in multi-display environments. The concept was developed by Manuela Waldner and the author, with contributions from the other authors.

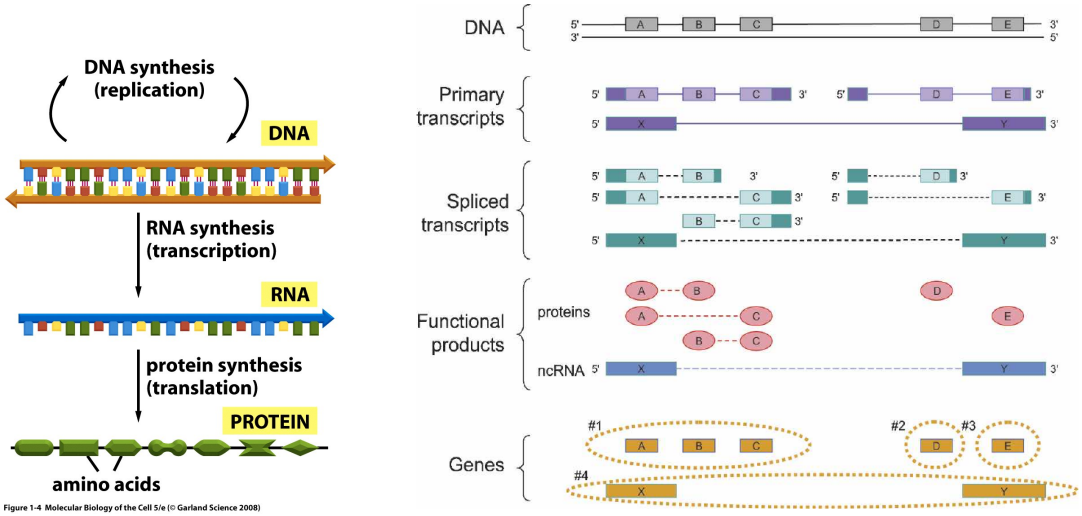
Chapter 2

Background in Molecular Biology

Contents

Molecular Biology is a sub-field of biology dealing with the molecular basis of biological processes. The related and intertwined field of **genetics** studies the hereditary process in living organisms. It is the hereditary process that distinguishes life from everything else [2, p. 1]. The hereditary information of all living things is stored in desoxyribonucleic acid (DNA), which is contained in cells. The two fundamentally different types of organisms are prokaryotes, simple, mostly single-cellular organisms with no cell nucleus, and eukaryotes whose cells contain membrane-based structures, most importantly the cell nucleus. The latter store their DNA in the nucleus where it is packed in pairwise sets of chromosomes. Prokaryotes are sub-divided into bacteria and archaea, eukaryotes include all species which are made up of large, complex conglomerates of cells, including humans.

DNA encodes its information by aligning nucleotides that are made up of one of four bases (adenine, guanine, cytosine, or thymine), and a sugar-phosphate (the link to the neighboring bases on the same strand). The sequential variation of nucleotides encodes the hereditary information. **Genomics**, a sub-field of genetics, is concerned with the entire sequence of DNA and genetic mapping, often using computational methods. DNA contains two strands, which are linked by weak hydrogen bonds along complementary bases (adenine binds to thymine and guanine to cytosine). The strands are twisted into a double helix form, wrapped around histones, and then tightly packed into chromosomes. DNA itself only stores information, the gene products that are based on the code stored in the DNA do the actual work. The most important gene products are **proteins**. Proteins cause the chemical reactions that make up most biological process. It is the DNA that encodes the proteins, but the process of how and how much of a protein is created is highly complex. The fundamental process of creating functional gene products is called **expression** and is illustrated in Figure 2.1(a). DNA is transcribed into ribonucleic acid (RNA), a single-stranded complementary molecule. Some parts of the RNA are then



(a) (b)

Figure 2.1: DNA, RNA, proteins and genes (a) Segments of DNA are transcribed into RNA. RNA is then spliced and translated to proteins [2, Fig 1-4]. (b) The different biomolecular products and how they interdepend. Three primary transcripts of DNA are produced in this example. Those transcripts are then spliced alternatively and translation produces five proteins and one noncoding RNA (ncRNA). These functional products are the basis of the definition of a gene: the three overlapping regions (in red, containing A, B, C) that produce proteins are collapsed to make up one gene (shown in yellow) – it follows that a gene can encode multiple proteins. The separate regions D and E make up one gene each. Also, the ncRNA (X and Y), based on the alternative primary transcript, is a functional product and is therefore considered a gene [56].

spliced (i.e., parts of the RNA are cut out), which produces messenger RNA (mRNA) that carries the coding information for a protein. The mRNA is then used in a complex translation process to produce proteins.

A region of the DNA that codes for proteins is called a **gene**. However, as the whole process is highly complex, a precise definition of a gene is not trivial. The following definition by Gerstein et al. [56] is now widely accepted: "The gene is a union of genomic sequences encoding a coherent set of potentially overlapping functional products". The implications of this definition are illustrated in Figure 2.1(b) – a single gene can produce multiple proteins. Also, the DNA sequence that is part of a gene can be a part of other genes as well. Parts of the DNA that do not produce proteins but other functional products are also considered to be a gene.

Proteins are polymers consisting of a sequence of amino acids [80, p. 92]. There are 20 different amino acids, each with different chemical properties. The amino acids are joined and fold to three-dimensional structures. The combination of the amino acids make up the unique chemical features of each protein [80, p. 94]. The study of the function and

the structure of proteins on a large scale is called **proteomics**. Proteins can perform structural functions, e.g., they can act as a messengers, transporters, or can influence the regulatory process itself. Most importantly, however, proteins catalyze chemical reactions, which are the building blocks of the biological processes in cells. Proteins fulfilling the role of catalysts are called **enzymes**. The interaction of many enzymes and the involved chemicals (metabolites) make up a metabolic **pathway**, which causes diverse processes. An example is the Glycolysis pathway that provides energy to the cell. The large-scale study of chemical processes involving metabolites is referred to as **metabolomics**.

2.1 Gene Expression Regulation

How much of a functional product is produced is described by **gene expression**. The levels of gene expression are controlled by various regulatory mechanisms, some of them epigenetic [58]. **Epigenetics** describes the modifications of the final outcome of genetic processes that are not caused by a change in the DNA sequence. It is defined as "the study of any potentially stable and, ideally, heritable change in gene expression or cellular phenotype that occurs without changes in Watson-Crick base-pairing of DNA" [58]. Epigenetics is a rather new field and many causes for epigenetic effects are yet unknown. One of the known epigenetic mechanisms is **DNA methylation**, which attaches methyl groups to specific regions of the DNA and thereby suppresses transcription. Another mechanism is **histone modification**, which changes the way DNA is packed around histones. Tightly packed regions are typically less expressed while easily accessible regions are more expressed.

Another part of the gene regulatory machinery are **microRNAs** (miRNAs) [21], which are short RNA molecules (around 22 base-pairs) that, unlike mRNA, are not translated into proteins (they are noncoding RNA), but regulate the translation of mRNAs into proteins. They do so by binding to complementary sequences of mRNA and thus repress the expression or silence genes altogether, but they are also known to increase expression in some cases. Albeit not yet fully understood, the locus of miRNAs also frequently coincides with hotspots for chromosomal abnormalities or loci suspected to be involved in cancer [21]. miRNAs are also suspected to regulate other ncRNAs.

2.2 Genomic Variability

While large parts of the genome are remarkably conserved across individuals, small changes in the sequence of bases cause the phenotypical variation we observe among individuals. While such variability is normal and even necessary, other changes, such as copy-number variations can have a profound negative (or sometimes positive) impact on the phenotype.

The most common form of sequence variation in the genome are **single nucleotide polymorphisms** (SNPs) [182]. The term polymorphism is used to denote an allele frequency of more than one percent in a population [45], and thereby distinguishes it from rarer forms of variations such as gene mutations. SNPs are responsible for many of the phenotypical variations, which make individuals different from each other. SNPs occur on average every 1000-2000 basepairs [182]. While SNPs are considered a normal varia-

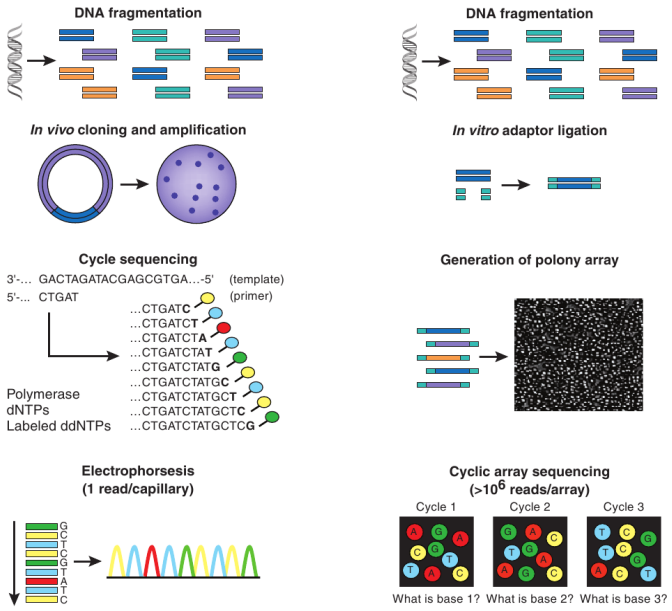
tion of the genome, combinations of particular SNP configurations are known to influence diseases.

Structural variations in contrast, describe variations such as insertions, deletions, rearrangements, inversions and translocations [208]. There are two types of variations: those that are dosage-altering, i.e., they have an effect on how much of a functional product is produced, such as insertions and deletions, and those that are dosage-invariant. Small-scale insertions and deletions are usually referred to as **indels**. Indels are common even in healthy individuals, with about 30.000 in the average genome [208]. However, such small-scale variations can also have a significant effect on the phenotype, especially when they are close or on genes. These **gene mutations** can lead to changes in the structure or function of the protein, which can have serious effects, for instance, if they affect tumor suppressor genes [25].

Mutations of a larger scale are **copy number variations** (CNVs) [45]. CNVs are genomic mutations that can occur, for instance, when the genomic DNA of a cell is copied incorrectly during cell division. Whereas gene mutations only affect single or a very small number of consecutive positions in the genome, these alterations may affect hundreds to tens of thousands of positions or even whole chromosomes. To be regarded as a CNV, the affected region has to have at least a length of 1 kilobase [45, 208]. Regions of the genome may be either amplified, resulting in an increased number of copies of the genes in that region, or lost, resulting in a decreased number of copies of the genes. Since normal cells carry only two copies of most genes (one in each of the complementary chromosomes; some genes in males have only one copy due to the Y-chromosome), they can either lose one copy – a "heterozygous deletion" – or both copies, resulting in a "homozygous deletion". On the other hand, there is no theoretical limit to the number of times a gene can be amplified. An increased number of copies of a gene, for instance, often leads to increased gene expression levels and vice versa. Copy number variation, however, is not only a source of irregularities, but also contributes to normal genetic variation between individuals [45].

2.3 Measuring Biomolecules

DNA microarrays, or DNA chips, are used to measure post-transcriptional products of DNA, such as mRNA [70]. By measuring the abundance of such mRNA molecules the activity of a gene – the *gene expression level* – can be determined. For genome-wide studies the gene expression level is typically used as an indicator for the amount of protein that is being produced for the corresponding gene. Another application of DNA microarrays is to screen for SNPs by applying fragmented DNA which binds to allele-specific probes. Microarrays can be produced in many different ways, for example, by ink-jet spotting or using in-situ photolitographic processes [70]. Microarrays are available at different "bandwidths", from a full array of up to 10^6 test sites, down to a selected number of genes used for diagnostic purposes [70]. The spots for hybridization consist of known sequences. A sample is applied to the whole chip, and, depending on the abundance of a specific sequence in the post-transcriptional product, binds with varying intensity to the complementary spots. Due to previous staining with fluorescent materials the intensity of the binding can then be read using imaging methods. Microarray data is unitless



(a) (b)

Figure 2.2: First and second generation sequencing. (a) The process of shotgun Sanger sequencing. First DNA is fragmented and then cloned. A fluorescently labeled molecule is attached to one of the four bases. The fluorescent markers can then be read in a linear sequence. (b) The process of cyclic-array sequencing, the most commonly uses second generation sequencing method. DNA is again fragmented, but then cloned in-vitro. The crucial difference to first-generation sequencing is that many fragments are placed in a polony array simultaneously. During several cycles the position of a base in each fragment is determined through imaging [165].

and must go through normalization (in reference to other samples) and quality control processes before it is ready for analysis [55].

DNA sequencing determines the order of the nucleotide bases in a DNA or RNA molecule. While several historic methods exist, the most important ones are Sanger sequencing (first generation) and second generation (or next generation) sequencing techniques. The process of sequencing with both techniques is illustrated in Figure 2.2(a). Sanger sequencing was used to assemble the first complete human genome in 2001 [83]. Sanger sequencing is more expensive due to the more complex in-vivo cloning and the serial reading of the sequence. Next generation sequencing significantly reduces the per-base cost of sequencing, is easier to conduct, and requires less infrastructure. It, however, produces much shorter sequences and the process results in more errors, which has to be counteracted using algorithmic methods. In fact the principal challenge of second-generation sequencing is the data management and analysis [165]. The cost of sequencing a whole genome has dropped from 100 million dollars in 2001, to ten million dollars in

2007, to 10.000 dollars in 2011 [203], making sequencing technology available to many users. Next generation sequencing is suitable to detect and measure many kinds of biological phenomena, such as all kinds of genomic variablity (including SNPs, mutations and CNVs), transcriptomic data such as gene expression, microRNA quantification, and protein interaction, just to name a few [165]. Consequently, sequencing of DNA and RNA is not only used to determine sequences, but can cover most use cases of microarrays as well. Figure 2.2(b) illustrates the process of polony sequencing, a prominent representative of cyclic array sequencing techniques [166]. All major commercial next-generation sequencing techniques fall into this category. **Cyclic array sequencing** is based on simultaneously decoding a two-dimensional array with millions or even billions of sequencing features [166]. Features are immobilized (their position is fixed) on a medium. In each of multiple cycles, a single base position within each feature is determined and recorded using imaging technology. After the cycles, the sequence can be inferred by analyzing the imaging information obtained during the cycles for each feature. Other sequencing technologies include **sequencing by hybridization**, where differential binding to an oligonucleotide array can be used to analyze the sequence, or **microelectrophoresis**, which follows the basic Sanger sequencing strategy, but miniaturizes the components through microfabrication techniques, to achieve a more parallel process [166].

Other methods of measurements are **mass spectrometry**, where the weight to electric charge ratio is used to identify the compounds, or **nuclear magnetic resonance** which can give insight into the molecular structure of compounds [55]. Both technologies are relevant for the field of proteomics and metabolomics.

2.4 Implications

The implications of the increased understanding of the hereditary process, ever since Gergor Mendel discovered the basic model of inheritance in the 1860s, cannot be overestimated. Asides from the fundamental knowledge of the process of life, molecular biology has applications in the diagnosis, prediction and treatment of diseases. While the causes of mongenic (Mendelian) diseases are comparatively well-understood and at the same time rare, virtually all major diseases are multifactorial with a polygene component. Examples are diabetes, cancer, and coronary heart disease [124]. The latter two make up roughly 50% of all deaths, with cancer taking a share of 23.2% [209] (US figures for 2007). Many individual genes that play a role in cancer have been identified. The most prominent example is maybe BRCA1, a gene involved in the repair of breast tissue. If BRCA1 is mutated the risk of developing breast cancer is significantly increased [128]. Clinically relevant gene expression patterns can, for example, be found in glioblastoma multiforme, a brain tumor. Verhaak et al. [192] recognized that glioblastoma subtypes can be characterized by gene expression profiles, and that the subtypes reported responded differently to treatment options. Molecular biology, however, does not only help classifying diseases, but leads to drug discovery and to personalized medicine [25]. The necessity of information technology tools for this kind of research is undisputed [26].

In this thesis we demonstrate the utility of the proposed visualization techniques for two use cases: a gene expression study on a mouse model to uncover the causes of liver cirrhosis,

and the aforementioned subtyping of glioblastoma multiform. The two use cases are quite different. The former is a study of a small scale, with about 50 experiments. It is conducted using microarray experiments with the goal to find the genetic causes for pathological changes in the liver which in turn may be linked to cirrhosis. The latter is a study conducted as part of "The Cancer Genome Atlas"(TCGA)* project, a large scale effort involving more than 150 researchers and more than two dozens of research institutions. Glioblastoma multiforme is one of 20 tumor types selected for comprehensive study, where 500 tissue samples are collected and a wide array of biomolecular data is recorded using next generation sequencing for each tumor type. We show that the techniques proposed are of value for both use cases.

*http://cancergenome.nih.gov/

Chapter 3

Related Work

Contents

The topics covered in this thesis draw from a number of areas of information visualization. We begin with an analysis of relevant multidimensional visualization techniques, and continue by covering more specific topics such as divide and conquer visualization techniques, or visualization in the biological sciences. We give as much detail as necessary to put the contributions of this thesis in the context of the state of the art, and give hints to more detailed resources where appropriate.

3.1 Multi-Dimensional Data Visualization

Multidimensional data can be displayed in many forms. There are three approaches to convey the information and structure of high-dimensional datasets:

1. Tabular display of the data in symbolic form (written numbers or text), using, for example, spreadsheets.

2. Display of statistical properties (either numerically or graphically).

3. Visual encoding of the data.

While the direct display of data in electronic spreadsheets is the most precise for reading individual values, it is difficult for humans to read global trends from tables. For decision making, however, the relationships emerging from the entire dataset are crucial [14, p. 1]. We have previously discussed the inability of humans to efficiently analyze large quantities of symbolically encoded data. Also, as the available screen space is limited, the number

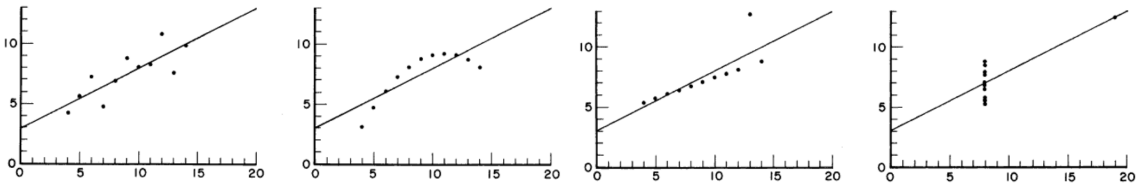


Figure 3.1: *Anscombe's quartett* [4]. The four different datasets plotted here have many identical statistical properties, e.g., the mean and variance of both dimensions, the correlation and the linear regression between the dimension. However, when looking at the plots it is obvious that the data is in fact quite different.

of symbols that can be displayed at any given time is limited. While this is also true for the graphical representation of datasets, the scale at which graphical representation fails is vastly different. It has been shown that more than a million items can be displayed without using abstraction (e.g., [43, 96]).

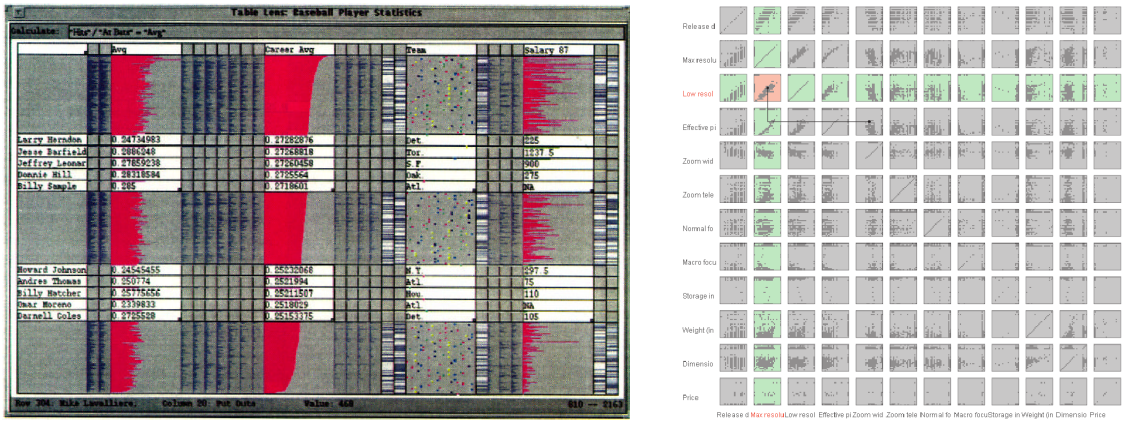
One might think that the overall relationships might best be uncovered using statistical analysis, but as Anscombe's quartet [4] impressively demonstrates (see Figure 3.1), statistical analysis can be ambiguous. Nevertheless, plots of statistical attributes, for example, histograms [140], can provide a valuable first glance at the distribution of a dataset. Of course, statistical or other computational analysis support an analysis. The field of Visual Analytics takes the approach of tightly integrating visualization and analytics methods in a continuous interplay [99].

A hybrid form of direct display and graphical representation is the *table lens* [145], shown in Figure 3.2(a). It utilizes a lens metaphor [51, 157] as a Focus+Context technique, distinguishing regions of "focus" - where all information is available using symbolical representations, and regions of "context", where small, abstract representations are used. In these context regions numerical variables are plotted using bars while categorical variables are encoded using color and position.

Keim [95] and Oliveira and Levkowitz [44] classify multidimensional visualization techniques into *geometric* techniques, where the data is projected to two or three-dimensional Euclidean space, employing the visual variables [15] position, size and orientation; **icon-based techniques**, which utilize visual variables such as shape, size, color and hue to encode data onto an icon; **pixel-oriented** techniques, where each data value corresponds to a pixel (or more general, to a mark), the color of which encodes the magnitude of the value; as well as **hierarchical** and **graph-based** techniques. Additionally, hybrid approaches are also very common. We discuss examples of geometric, pixel-based and hierarchical techniques, as they are most relevant in the context of this thesis.

3.1.1 Geometric Techniques

Scatterplots are a very simple, yet powerful geometric technique to visualize the relationships between two dimensions. Scatterplots use points (marks) in the Cartesian coordinate system, which is determined by a function of the value of the attribute (e.g., linear mapping, logarithmic mapping, of fish-eye functions). A single scatterplot can show only two dimensions at once. To accommodate more dimensions, several strategies are

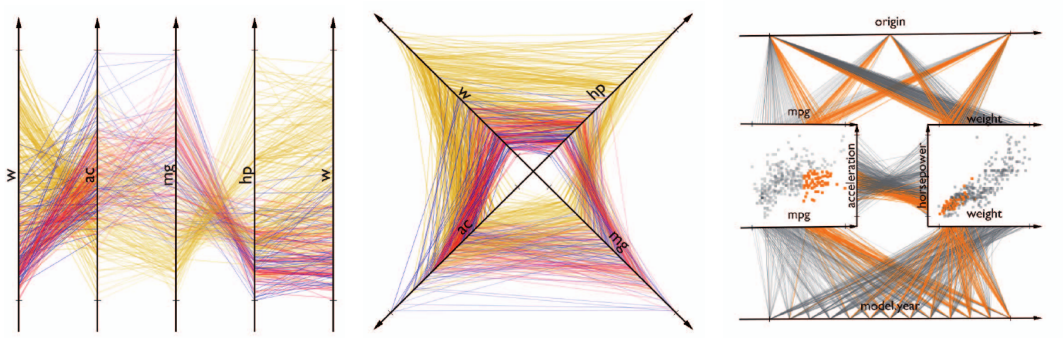


(a) (b)

Figure 3.2: (a) The *table lens* [145]. Focus areas are shown in full detail to make the symbolical representations legible. In context areas space-efficient visual encoding is employed. (b) A *scatterplot matrix* showing all combinations of 12 dimensions, including duplicates above the diagonal.

possible. The most obvious would be to use three-dimensional (3D) space instead of two-dimensional (2D) space. However, 3D introduces issues of occlusion, but most importantly 3D plots of data do not follow the rule that "the representation of numbers, as physically measured on the surface of the graphic itself, should be directly proportional to the numerical quantities represented" [186, p. 77]. Ignoring this rule can lead to perceptual errors. It is common to use icon-based techniques for the marks in scatterplots, assigning, for example, color or size to encode additional dimensions. Still, as the number of visual variables is limited and the perceptual effectiveness of combining many visual variables is doubtful, alternative approaches are required. One example is the *Grand Tour* [7], which selects a dense set of "projections" (scatterplots) to present to a user. Another very common approach is to show all possible combinations of dimensions and arrange them in matrix-form, yielding a *scatterplot matrix* [22, 23] as shown in Figure 3.2(b). As, for many dimensions, individual scatterplots are typically rather small, they are most commonly used in a multiple coordinated view [149] fashion. The matrix provides an overview and a duplicate of one scatterplot is enlarged to provide full interactivity. Sophisticated methods to navigate, query, and brush scatterplot matrices have been developed (see, for example, [40, 121, 193]).

A widely used geometric, multidimensional data visualization technique are *parallel coordinate plots* (PCP) [81, 82], shown in Figure 3.3(a). Related techniques are the *radar chart* (see Figure 3.3(b); also known as *star plots*) [23] and the *TimeWheel* [184]. The principle of those techniques is to use one axis for each dimension and arrange these axes on a plane – in the case of PCPs the axes are arranged sequentially, parallel to each other; the axes of radar charts have one shared center point and spread out radially; TimeWheel arranges the axes in a wheel with one axis in the center. For every record a polyline is



(a) (b) (c)

Figure 3.3: Axis-based visualization techniques. All plots show properties of cars such as acceleration (ac), weight (w) and horse-power (hp) [144]. (a) Parallel coordinate plot for five dimensions. (b) Radar chart for four dimensions. (c) Hybrid of scatterplots and parallel coordinates plot for eight dimensions. All images taken from [27].

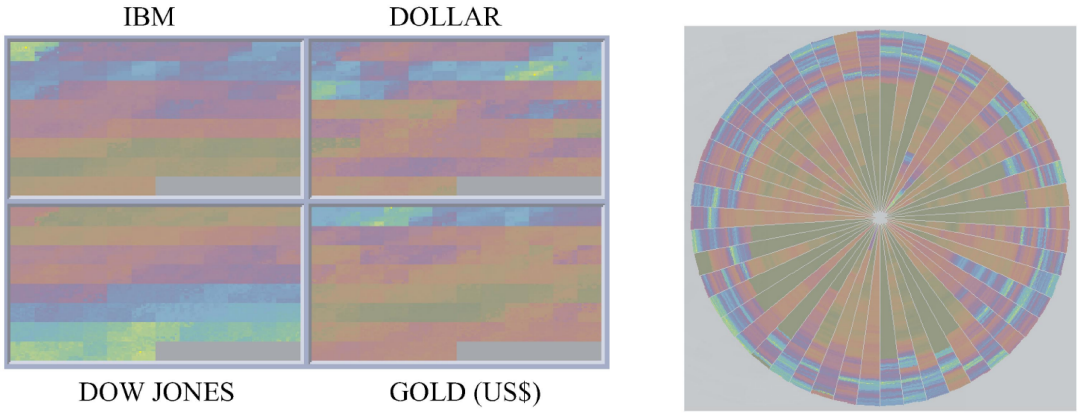
drawn between adjacent axes. The point of intersection of the polyline and the axes is determined by the magnitude of the record's value at the associated dimension.

Parallel coordinates have been researched extensively. Methods for clutter reduction such as sampling [34, 39] or clustering [49] have been developed. Advanced brushing techniques, for example, structure-based [50], composite, smooth, and angular brushing [64] enhance the interaction with parallel coordinates and its derivates. Studies on usability and perceptual issues found that parallel coordinates are effective and easily comprehensible even for novice users [171, 172]. Hybrids between parallel coordinates, scatterplots (see Figure 3.3(c)) and other visualization techniques such as histograms have been developed (e.g., [27, 101, 193, 211]).

Approaches to deal with an overwhelming number of dimensions, which is a problem for scatterplot matrices and to a lesser degree for parallel coordinates, are dimension reduction methods such as principal component analysis (PCA) [88]. However, we observed that users are often not interested in automatic reduction methods. In many cases, the input data is part of a well designed experiment, where users have a priori knowledge of the dimensions' semantics and may already have hypotheses about their relationships. Nevertheless, recent research has shown promising directions on how to reduce the number of dimensions, while allowing users to interactively specify which types of features they are interested in [85].

3.1.2 Pixel-based Techniques

Pixel-oriented techniques, shown in Figure 3.4, divide the screen into a set of windows, each window corresponding to one dimension. Within these windows, the data values of the dimension are arranged in equal order as small squares or pixel, with the magnitude of the value encoded using color [95]. The two main challenges of pixel-oriented techniques are the



(a) (b)

Figure 3.4: Pixel-based techniques. (a) Four dimensions with rectangular windows and semantic, recursive pixel arrangement. (b) 50 dimensions in circle segment windows and linear pixel arrangement. Both images taken from [95].

choice of an adequate color map and the choice of a suitable arrangement of the pixels. For the choice of color, Keim [95] recommends a multiple-hue color map, designed in a way that brightness increases monotonically. He argues for multi-hue color maps, over just varying brightness or saturation, because of an increased number of just-noticeable differences. However, multi-hue color maps bear considerable risk of misinterpreting data [18, 19, 90]. For choosing an arrangement of pixels, Keim argues for user-driven (semantic) recursive arrangement, as shown in Figure 3.4(a), over mathematically optimal arrangements such as *Peano-Hilbert* arrangements. The final challenge is to choose the shape of the windows. Solutions range from rectangular windows [97], as shown in Figure 3.4(a), to radial layouts, as shown in Figure 3.4(b), where the windows correspond to circle segments [3].

Heat maps [38], as they are commonly used for gene-expression data analysis, are a simple form of the above-discussed pixel-based techniques. A typical example is shown in Figure 3.5. The window for the dimensions is rectangular, the arrangement of pixels is linear, with a width of only one pixel and n-pixels height (or vice-versa). Heat maps most commonly employ a diverging red-black-green color map to encode the magnitude of the values. Recently, more perceptually justified color maps are being employed. This is owed to the fact that about 5% of the male population has a red-green deficiency (dichromacy) [53, p. 30] caused by a recessive trait on the X-Chromosome, of which males only have one. An alternative color-map is the red-white/gray/yellow-blue diverging color map [131]. The justification for employing diverging color maps is found in the properties of the data. The neutral color encodes “normal” regulation, while the others encode up- respectively down-regulation. Heat maps are covered in more detail in Section 3.6.

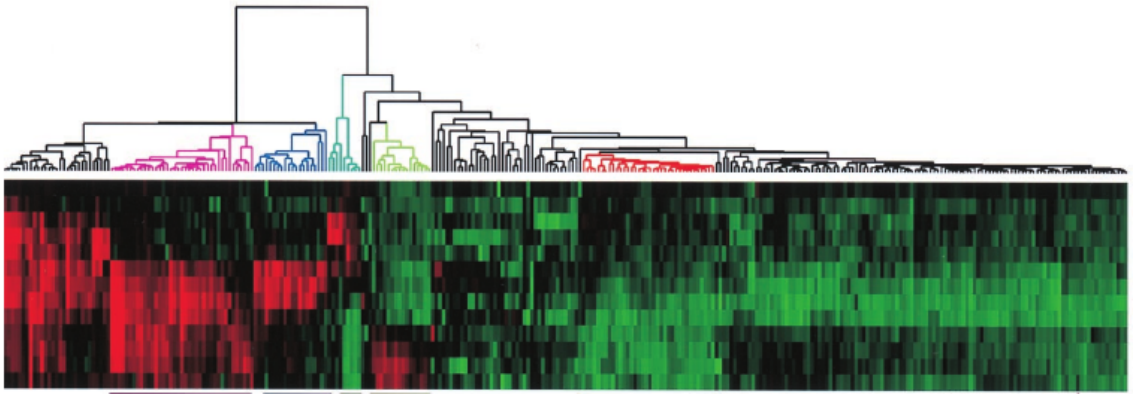


Figure 3.5: Clustered heatmap with dendrogram showing gene-expression data [38]. Green shows down-regulated, black neutral, and red up-regulated gene expression.

3.1.3 Hierarchical Techniques

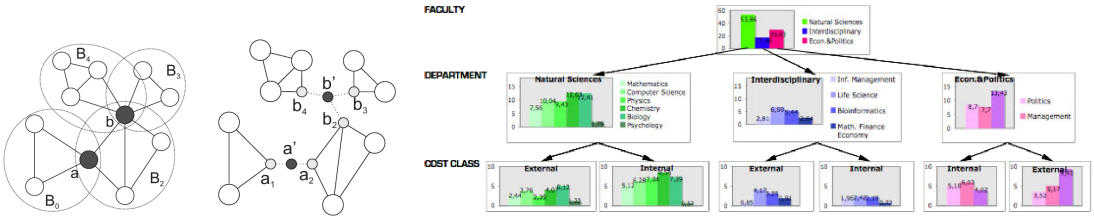
Feiner and Beshers [16, 42] propose a hierarchical approach to visualize multidimensional datasets by embedding virtual worlds within each other . Another approach is dimension stacking [110], where dimensions and their associated values are recursively stacked within each other. These hierarchical techniques are difficult to understand and are only usable with a limited number of dimensions [44]. Consequently they have not been adopted widely.

3.1.4 Summary and Context

In this section we discussed the different possibilities to visualize multidimensional data. As a large part of this thesis deals with multidimensional data, these techniques form the baseline for our own work. Our divide and conquer approach is a hierarchical technique: we show the structure of the data on one level and the actual data on a second, more detailed level. On both levels we draw from visualization techniques discussed here.

3.2 Divide and Conquer in Visualization

Divide and conquer strategies in visualization are employed when breaking down a visualization problem into sub-problems makes solving the original problem easier, more efficient or more effective. Inhomogeneous data lends itself well to divide and conquer approaches, as homogeneous parts of the data can be more easily analyzed, abstracted and displayed. Divide and conquer strategies have been investigated most frequently for graph data and its visualization, possibly because of the unfavorable runtime complexities of analysis and layout algorithms for general graphs. Graph layout algorithms benefit greatly from a subdivision of the data into smaller, more homogeneous subsets, which can be efficiently processed individually, and then compiled for an overall result. In addition to the gain in speed, this strategy can also generate more expressive representations, because the sub-layouts can be optimized.



(a) (b)

Figure 3.6: Divide Strategies. (a) An example of an improved graph-clustering based on a divide and conquer strategy [1]. The graph is split up at the articulation points, (dark grey on the left), the sub-parts are then clustered separately. (b) A decomposition tree illustrating the semantic divide process employed in *OLAP*. The plots show the finances of a University. The data is split based on two criteria of homogeneity (faculty and cost class) [122].

The following two sections give a short overview of existing approaches for the visualization of inhomogeneous graphs and tabular data by discussing different methods that are often used to perform the divide and the conquer steps.

3.2.1 Dividing Inhomogeneous Data

For large graphs, the subdivision of inhomogeneous data is performed purely in the data space, as it has to be performed before the mapping (i.e., creating the layout), which may be time-consuming. Graph theoretical methods are used to determine more coherent subgraphs within the inhomogeneous overall dataset. In most cases, these subdivision methods are hierarchical clustering algorithms or traversal strategies for identifying connected components; both are often used together. A possible way to combine these methods is to first perform a quick traversal to identify (bi-)connected components that are then further clustered hierarchically in a second step [1]. An illustration of the process is shown in Figure 3.6(a).

For multidimensional, tabular data, statistical subdivision methods are usually employed. In case of dividing records, the subdivision is based on the statistics via (hierarchical) clustering, or on the semantics, as is often observed for *OLAP* (online analytical processing) like partitioning of the data space into different value ranges. Examples for clustering algorithms are hierarchical algorithms [38], which yield a similarity tree, or partitional algorithms such as k-means [118] and affinity propagation [46]. OLAP is a technique for data mining in multidimensional datasets and relational databases primarily employed in business applications. Basic OLAP techniques include *slice and dice*, which is used to reduce the dimensionality by selection or projection, *roll-up*, which increases the level of aggregation, and *drill-down*, which decreases the level or aggregation or increases detail [24]. The purpose of OLAP is to not just perform equidistant partitioning, e.g., a person's age in sets of 10; instead, it brings in common knowledge and makes more meaningful partitions, such as being of legal age at 18 or retiring at 65. An example for an OLAP-based slice and dice operation is shown in Figure 3.6(b).

The same is true for subdividing dimension, which is also based on statistics through the aforementioned correspondence analysis or on grouping dimensions according to their semantics; a user would likely place zip codes and a person's age in different dimension groups, even if for some reason the statistics found a correlation between both.

The divide step is a crucial one, because it pre-determines many of the features a user will later see in a visualization of its results. A falsely parameterized algorithm may result in an utterly useless visualization that does a good job at communicating false results that are not actually representative of the data. Hence, different tools and frameworks have been devised to support the user during the divide step. For dividing the dimensions a hierarchical dimension management framework [210] can be used to construct subspaces, orders and filter dimensions. For the subdivision of the records, one option is the Hierarchical OLAP visualization [122], which supports the subdivision of the data space via OLAP and allows the user to interactively steer the subdivision process.

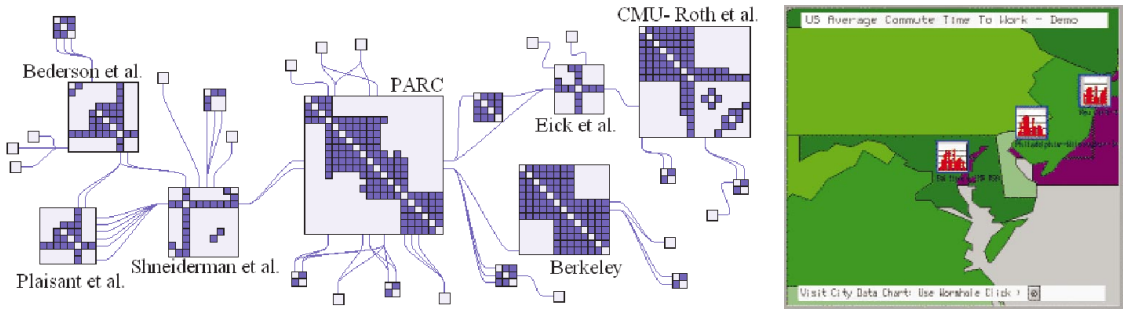
It is important to note that the created subdivisions do not necessarily need to be disjoint, even though often they are generated without overlaps, which makes the following conquer step easier.

3.2.2 Conquering Inhomogeneous Data

After the inhomogeneous data has been subdivided, the resulting groups are processed and visualized individually. Finally, the outcomes for all groups have to be fused together to form a visualization for the whole dataset again. The result can be a uniform visualization, in which all individual visualizations are of the same kind, or a multiform visualization, in which entirely different visualizations are merged together [148]. In the field of graph visualization, an example of a uniform visualization is the *TopoLayout* [6], which hierarchically combines different node-link layouts of subgraphs. For multiform visualizations there are examples of pairwise combinations of all three major graph representation types, i.e., matrix, node-link, and implicit layout: *NodeTrix* [71], shown in Figure 3.7(a), combines a matrix with a node-link layout; *Elastic Hierarchies* [213] combine a node-link with an implicit layout; and Rufiange et al. [153] combine a matrix with an implicit layout.

To assemble an overview of a subdivided tabular dataset, the individual visualizations of the subsets need to be patched together. Conceptually, there are two ways of doing that. The first possibility is a very rigid arrangement of the visualizations in a certain style that reveals relationships merely by thoughtful positioning of the individual views. Examples for this approach, however, are scarce. Two notable techniques applying this approach are portals, as used in the *DataSplash* system [207], and matrix-arrangements as used in *Multiform Matrices* [121]. Portals are locally embedded smaller visualizations in a larger base visualization. The relationship among different portals is communicated automatically through their position. In Data Splash, for example, the individual visualizations are put in the context of a map representation. An example is shown in Figure 3.7(b).

Multiform Matrices extend the basic concept of scatterplot matrices and add other visualizations to the matrix arrangement. In the example in Figure 3.8(a), the redundant cells of the scatterplot matrix are used to show geo-spatial attributes of the data. The association of the view to the part of the data it represents is clearly conveyed through the position in the matrix.



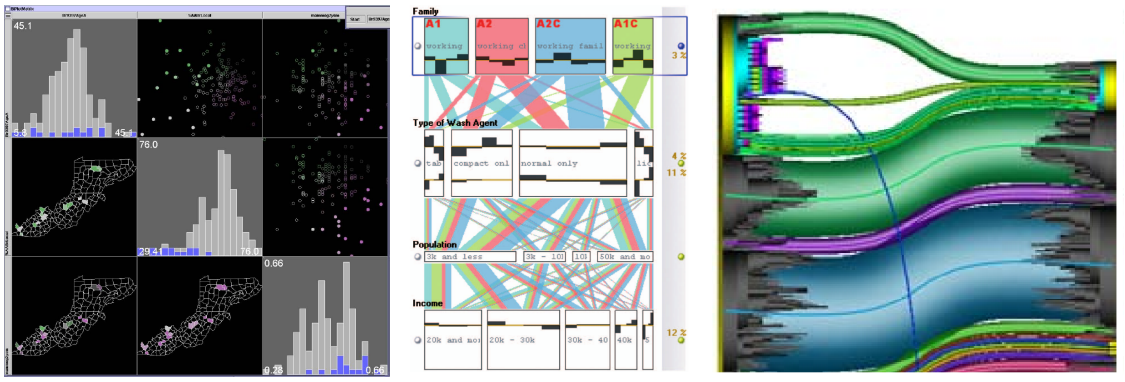
(a) (b)

Figure 3.7: Conquering strategies: graphs and portals. (a) *NodeTrix*, a divide and conquer strategy for graphs employing multi-form encoding [71]. The graph shows co-authorship of information visualization authors. Semantic grouping (based on research institutes) for highly connected parts is used. The dense groups are shown with a matrix-based graph layout, which is well suited for highly connected graphs. Loosely connected components of the graph are connected using a node-link layout. (b) *Data Splash*, a visualization using the portal technique. Portals with visualizations of tabular data are embedded in a map [207].

In theory, both of these techniques have the potential to employ multiform visualizations; however, the examples always show the same visualization in all added views.

The second possibility is to allow a more flexible arrangement of views and to use other visual attributes to encode their relationships. Assuming a fixed layout (i.e., the visual variable position is employed otherwise) **synchronous highlighting techniques** must be employed to achieve their association. Cockburn et al. [29] categorize highlighting as a cue-based focus-and-context technique. Synchronous highlighting is also often referred to as brushing [123] in multiple coordinated view systems, especially if multiple elements are simultaneously highlighted. Seo and Shneiderman [159] mention three basic techniques to encode relationships: color coding, drawing lines and blinking. However, in our opinion this is not general enough (color coding is but one option for an in-place technique), and also too specific, as we see no conceptional difference between blinking and other in-place techniques. We distinguish between three classes of highlighting techniques: those that employ **in-place encoding** such as color-highlighting or glyphs (for example, arrows), those that **modulate the surrounding** of a highlighted item, for example, blurring everything else, and those that employ **connectedness**, meaning that they actually connect the highlighted items using some geometry.

An example for a divide and conquer visualization technique using the **connectedness** approach are *Parallel Sets* [104], which connect boxes using parallelograms. The size of the boxes is proportional to the number of elements in the category. Inside the boxes Parallel Sets can embed histograms, thereby visualizing the distribution of the contained data, as can be seen in Figure 3.8(b). The focus of Parallel Sets, however, is not the display of embedded visualizations, but to encode the magnitude of relationships between different categories. Consequently, although in theory it would be possible to use this



(a) (b) (c)

Figure 3.8: Conquering strategies: position and connectedness. (a) Multiform matrices – an example for a technique that uses position to encode relationships between divided parts of the data (Figure modified from [121]). (b) and (c) both use connectedness to re-introduce the connections lost in the divide step. (b) Parallel Sets [104] connect histograms with parallelograms to show the relationships of multiple categories. (c) CodeFlows shows the evolution of source code using pipes. The structure of the code is shown using icicle plots [180].

technique as a multiform visualization with different views being connected, in practice it has so far only been used in a uniform manner with all views utilizing the same kind of representation. Parallel Sets are discussed in more detail for its categorical visualization properties in Section 3.4.

CodeFlow, by Telea and Auber [180], also employs connectedness for multiple-view association. CodeFlow is a system for comparing different versions of source code on the code level. While such data can be considered as graph data, it can also be seen as multidimensional data, where each revision is one dimension. Telea and Auber use icicle plots placed along vertical axes, where each axis represents the version of the software system under investigation. They then draw spline tubes between corresponding fragments in different versions. The tubes can either be shaded, as shown in Figure 3.8(c), or opaque in the middle and translucent at the borders, to allow a clear separation of the tubes. To draw the user's attention to changes, they use color for the bands that changed within the range of currently inspected versions, while others are rendered gray. Telea and Auber's application domain is vastly different from ours: source code evolves gradually and thereby makes bundling or similar measures obsolete. In addition, Code Flow does not provide drill-down methods that preserve the context to the whole dataset.

Another example is the, for its devision strategy previously discussed, decomposition tree [122] shown in Figure 3.6(b).

3.2.3 Summary and Context

To our knowledge, there is no visualization technique employing in-place techniques or the modulation of the surrounding to encode relationships between multiple fragments of datasets, with maybe the exception of window titles in traditional multiple-coordinated view (MCV) systems. By relaxing the requirement for related work from relating multiple fragments of a dataset to general encoding of relationships we find a whole body of important literature, which we will cover in the next section.

Our own approach for divide and conquer visualization of tabular data integrates and advances these ideas. We propose a flexible technique that combines both, thoughtful arrangement, and linking for multiform visualizations of subsetted inhomogeneous data.

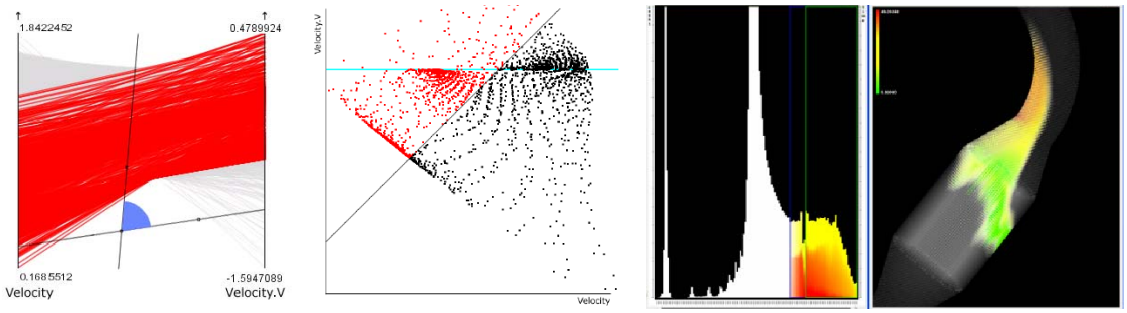
3.3 Expressing Relationships

In the previous section, we elaborated different possibilities to re-introduce connections lost due to the divide step of our divide and conquer approach. We found that expressing relationships between the spatially separated entities is an important concept. Expressing relationships includes the concept of synchronous highlighting (brushing). We distinguished three types of techniques to express relationships: **in-place** techniques, techniques that **modulate the surrounding** and techniques that employ **connectedness**. As those techniques are relevant for any kind of multidimensional data analysis, including the divide and conquer techniques presented in this thesis and the analysis of general heterogeneous data, we discuss those techniques with a broader scope.

3.3.1 In-Place Techniques

We define in-place relationship expression techniques as techniques that mark relationships between associated elements either by co-modulating their appearance (e.g., synchronous coloring, synchronous blinking), or by adding a glyph (e.g., a pointer or label), either in immediate proximity, or connected to the item.

Color is being almost universally employed for highlighting and brushing. Examples are shown in Figures 3.3 and 3.9(a). Van Long and Linsen [119] use colored brushing to show relationships between a cluster tree and the concrete values in a parallel coordinates browser. Graham and Kennedy [59] show multiple trees and visualize their relationships by interactive linking and brushing. Most MCV systems, like, for example, *Tableau* [175], or *VisPlore* [142], employ colored brushing to connect views such as parallel coordinates, histograms, or scatterplots, as can be seen in Figure 3.9(a). Reasons for the widespread adoption of color are its ease of implementation, its ability to concurrently highlight arbitrary numbers, and its preattentive properties [185, 205]. Preattentive entities are recognized immediately, independent of the number of distractors. Non-preattentive attributes require serial search, i.e., conscious (attentive) comparison of every item. Employing color for highlighting, however, also has several drawbacks. While color is ideally suited to encode many items of one class, color is ill suited to encode many classes, i.e., the selective properties of color are limited. Healy [66] found that more than seven colors lead to reduced performance in accurately and rapidly detecting the colors. Also, color may be



(a) (b)

Figure 3.9: In-place brushing techniques: color and saturation. (a) Brushing with color in a MCV system. The brushed elements are shown in red in both views (Figure modified from [64]). (b) Brushing with saturation. Brushed items are shown in color, the others unsaturated. The color encodes the additional dimension temperature (Figure modified from [35]).

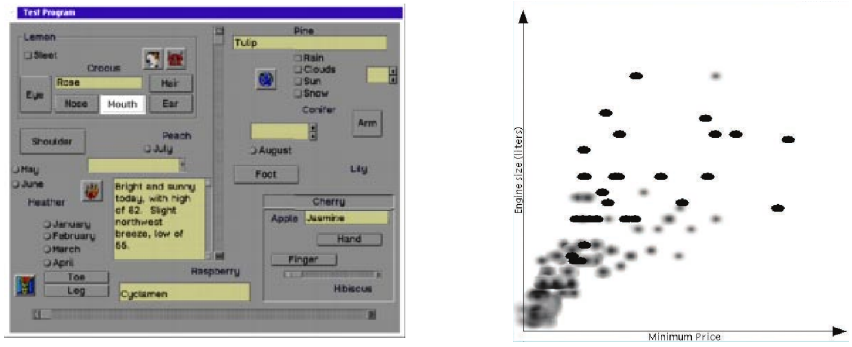
already employed to encode other attributes. SimVis [35], for example, uses color to encode other parameters, and falls back to (binary) saturation to highlight brushed areas, as can be seen in Figure 3.9(b). Equally, color can not be used for highlighting in pixel-based techniques.

Synchronous blinking can also be used as an in-place technique. While it is very preattentive, it is also considered disturbing by many users and can hardly be used for more than one or two items. Other in-place techniques, such as drawing frames, underlining, etc., are primarily employed in general software (e.g., for highlighting misspelled words) and seldom in visualization software. These techniques are typically combined with color.

Drawing glyphs or symbols and labeling can theoretically encode many relationships simultaneously. However, glyphs and labeling are even less selective than color, meaning that finding two related items requires serial search when enough distractors are present.

3.3.2 Modulating the Surrounding

Modulation of the surrounding is typically done by decreasing saturation [212], brightness [100, 212], or sharpness [105]. The latter two are shown in Figure 3.10. Zhai et al. [212] show that darkening and decreasing saturation are highly effective but negatively affect user satisfaction. Hoffmann et al. [72] reproduced the negative user rating for darkening, and found that darkening was more error-prone than colored highlighting or connectedness. Kosara et al. [106] found that blurring is also highly effective as a highlighting technique. We can generalize that the modulation of the surrounding is very effective, but not versatile and not scalable. In fact, it is not possible to express more than one relationship at once. Also, the implementation is sometimes not straight-forward (blurring may require shaders, for example). Combined with the low user satisfaction this may be the reason why these techniques are not widely used for highlighting (synchronous or individual), even though all these techniques are preattentive. A more promising ap-



(a) (b)

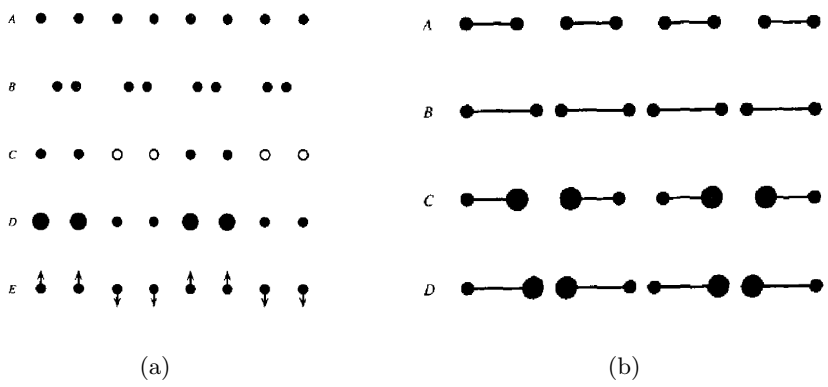
Figure 3.10: Highlighting by modulating the surrounding: darkening and blur. (a) Everything but the highlighted button is darkened [212]. (b) Everything but the brushed elements is blurred [106].

plication scenario for such techniques are maybe minimal, hardly perceivable alterations of images, to sub-consciously guide user-attention [191].

3.3.3 Connectedness

Connectedness (or *uniform connectedness* [139]) was shown to be a very strong grouping principle, even stronger than classic *Gestalt* principles [201] such as proximity, similarity (color, shape, size, brightness), or common fate [139, 215], as illustrated in Figure 3.11. It was also shown that connected elements are perceived preattentively, approximately at the same speed as proximity, but faster than similarity [62]. Ziemkiewicz and Kosara distinguish between three forms of connectedness, namely outline, connector and fill [215].

We distinguish between **general links**, as they are, for example, used in node-link diagrams (where the links are representation of the edges, which are part of the data structure), or in parallel coordinates plots (where the links encode the actual information) and visual links. We define **visual links** as "continuous shapes such as connection lines, curves, or surfaces that connect or surround multiple related pieces of information, thereby augmenting a base representation" [174]. In this context, a base representation is a image or visualization that is meaningful without the addition of visual links. The notion of base representation sets visual links apart from the general links, as they are used in node-link diagrams, where the meaning of the diagram is lost if the links are not present. There are two types of base representations. The first one is not aware of or does not adapt to visual links at all, i.e., visual links are superimposed on existing visualizations. The second type of base representations may leave empty space for the visual links, or may adjust its content for improved links routing. We present representatives of both classes in this thesis: the Matchmaker, VisBricks and StratomeX techniques use base representations of the second type, while the visual linking techniques presented in Chapter 8 are designed for base representations that do not adapt to the visual links.

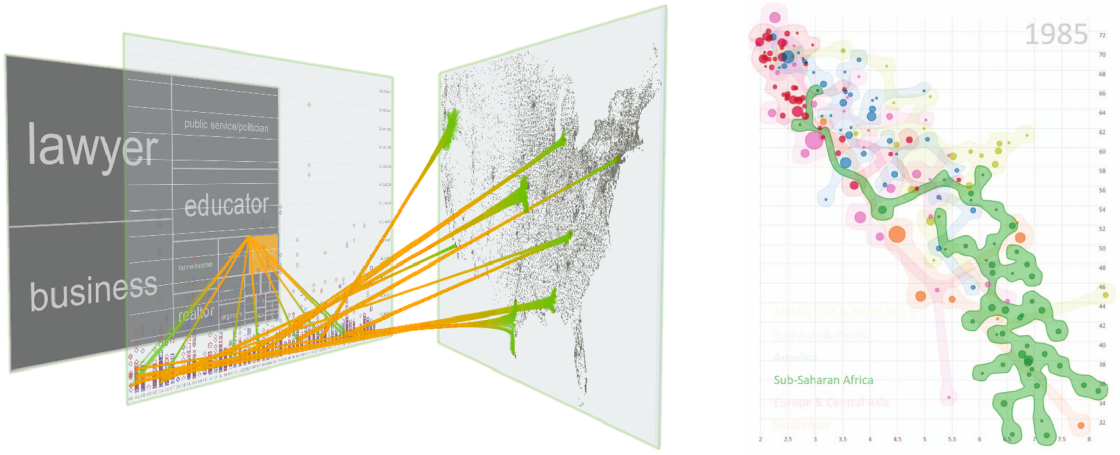


(a) (b)

Figure 3.11: Classic *Gestalt* grouping principles compared to connectedness. (a) Gestalt Grouping principles: (A) ungrouped, (B) proximity, (C) color, (D) size and (E) common fate. (Figure modified from [139]). (b) Demonstration that uniform connectedness overcomes classical gestalt grouping principles. Comparisons to (A) no Gestalt grouping, (B) proximity, (C) size and (D) proximity and size combined [139].

Linking entities, albeit being a strong grouping and highlighting principle, does not scale well. As the number of links increases, their paths become hard to follow. Bundling strategies have been developed to group and organize the links and make linking scale to larger numbers. Bundling strategies either utilize an underlying structure, such as a hierarchy [75, 77]; use a force-directed layout where links attract each other [76]; or formulate the problem as an optimization to minimize the required ink [52]. A related problem, which is more relevant for visual links than general links, is that clutter makes the underlying base representation hard to read. Of course, the above-mentioned bundling strategies can improve the situation, but they do not guarantee to minimize the impact on the base representation.

Semantic Substrates [169] are a technique to handle large graphs by employing a semantic grouping of nodes. Within those groups, the nodes are arranged based on node attributes, resulting in a view similar to scatter plots. The links within or between plots can be added by interactively refining queries. One could argue that this is one of the first works using interactive visual links between disjoint plots. Collins and Carpendale [30] have generalized the Visual Links concept to connect multiple, arbitrary visualizations. They arrange visualizations in a restricted 2.5D environment and connect them with links. An example, where a treemap, a scatterplot, and a map is connected is shown in Figure 3.12(a). An example for visual links using shapes instead of curves or lines is *Bubble Sets* [31], which shows a set relationship (or alternatively a brush) as a hull around a set of selected features. Figure 3.12(b) shows an example of geographic relationships overlaid on top of a scatterplot. Examples from the biological domain employing connectedness are HCE [159], Circos [107], and MizBee [126], which we will elaborate on in Section 3.6 when dealing with visualization for the biological sciences.



(a) (b)

Figure 3.12: Visual links connecting multiple views. (a) *VisLinks* generalizes the visual links concept to different visualizations [30]. (b) *Bubble Sets* uses surfaces as visual links. The example shows a scatterplot of fertility rates versus life expectancy for several countries. Visual links are used to connect countries of the same continent [31].

3.3.4 Summary and Context

Picking up the ideas of general visual links, we will show that visual links are a suitable method to integrate visualizations of general heterogeneous data, even across multiple applications. We will discuss several technical refinements such as the thoughtful arrangement of views in 2.5D space, as well as a routing algorithm taking base representations into account. Additionally, by comparing visual links to traditional highlighting techniques in a controlled experiment, we will show that visual links are an excellent method to express relationships between spatially disconnected entities.

3.4 Categorical Data

For some tasks the composition of a stratification and the comparison to other stratifications is more relevant than the actual data. Since subset membership can be treated as a categorical variable, visualization methods for comparing categorical data would be suitable representations for these cases. We also consider categorical data as a possible source format. Consequently we review literature on the visualization of categorical data.

The literature describes two principle approaches to categorical data visualization: conversion of the categorical visualization problem to a quantitative problem in data space, as well as explicit representation of categories in view space. A recent study [86] suggests that both variants have their place in visual data analysis, as each of them is suited best for specific visual analysis tasks. Explicit representation was found to work better for frequency-based tasks, addressing questions such as, “Which category is the most

common?", while the conversion method works better for comparison tasks, answering questions such as, "Which two categories are most similar?" The following will briefly describe the related work for both approaches.

3.4.1 Conversion into Quantitative Data

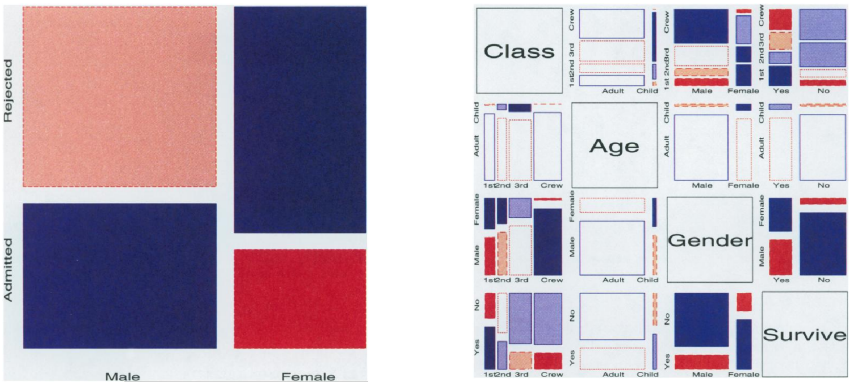
The motivation behind converting categories into quantitative data is to make the data processable by automatic methods, and displayable by visualization, with techniques originally devised for quantitative data. A simple example for a conversion is a linear mapping of the categories to an interval in quantitative space. The transformed data can then, for example, be displayed in a parallel coordinates plot [65]. However, such a trivial mapping often produces undesired results. For example, when completely unrelated categories are next to each other in the quantitative space, this can have adverse effects on the display and on computational methods. It is therefore preferable to use advanced mappings.

Categorical data can be either ordinal or nominal. In the ordinal case, the categories are inherently ordered, while nominal categories are not. If, for the aforementioned reasons, an order of the categories is desired it can be computed, e.g., by *correspondence analysis* [60, 151] or clustering-based approaches [17, 120]. The reasoning behind most of these approaches follows Friendly's mantra of *Effect Ordering*: "Sort the data by the effects to be observed" [48]. A second step then computes a spacing between the categories to convey the degree of similarity between the categories. An established method to achieve this is the *Optimal Scaling* approach [151], which is able to use the output of a correspondence analysis for deriving a spacing. An alternative method is proposed by Shen et al. [164], who map categorical to numerical data via a reference set. After this transformation, arbitrary techniques for quantitative data can be used, where parallel coordinates are a common choice. Depending on the type of mapping, some additional measures counteracting effects of showing categorical data in parallel coordinates can be taken. Havre et al. [65], for example, compare different clusters by introducing intermediate lines where they spread all polylines evenly, thus reducing the overplotting problem.

3.4.2 Explicit Representation

If categorical data is not transformed to quantitative data, there are two general ways of visualizing it. Similar to the approaches described for the conquer step of divide and conquer visualizations, these are to use relative positions or to use an explicit encoding of relationships.

An example for **relative positions** are slice and dice subdivisions of the drawing area, such as the *mosaic plots* [73], in which the size of the area representing a category is proportional to the number of data records in the dimension. An example of a simple mosaic plot is shown in Figure 3.13(a). Mosaic plots encode the relationships of two dimensions. Each dimension is associated with one axis of a Cartesian coordinate system. The intersection of two categories in the dimensions are depicted as blocks. The relative frequencies of the categories of one dimension are encoded by the length of the edge parallel to the dimension's axis. Mosaic plots can be easily read and understood. To overcome the limitation that only two dimensions can be shown at a time, arrangements such as



(a) (b)

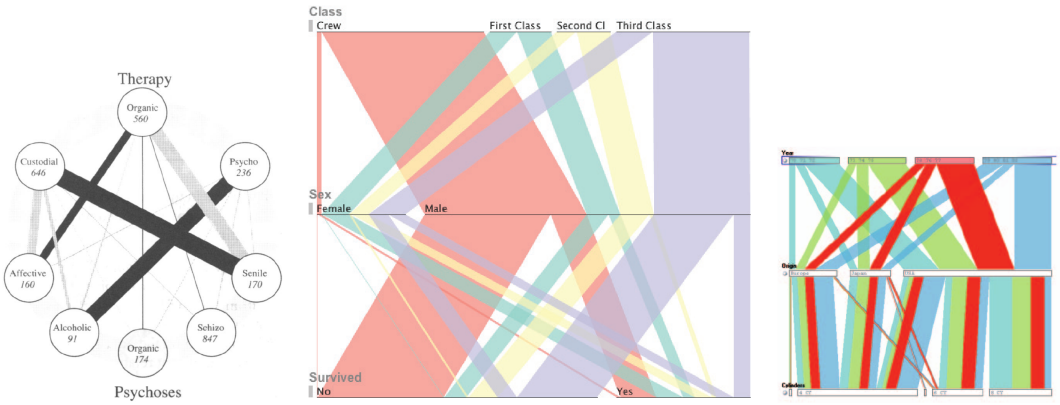
Figure 3.13: Category visualization using relative positions. (a) A simple *mosaic plot* displaying the admission rates at a university. (b) A *mosaic plot matrix* showing the relationships between 4 categorical dimensions for passengers of the *Titanic*. Both images taken from [47].

the *mosaic plot matrix* [47] were developed. Mosaic plot matrices place multiple bivariate mosaic plots in a scatterplot matrix-like arrangement, as shown in Figure 3.13(b). This shows the relationships of multiple dimensions simultaneously.

However, interactions between more than two dimensions can not be read from such a plot. Figure 3.13(b), for example, shows a dataset on passengers of the *Titanic*, with the dimensions *class*, *age*, *gender* and *survival*. In this plot it is easy to read that the majority of the passengers in 1st class survived, while most of the crew perished. Similarly, it is easy to see that most females survived. The question whether most female crew members survived cannot be answered.

While mosaic plots are not often used to compare stratifications or clusterings, there is an example of their use for comparing a clustering of records across different categories [74]. Visualization resembling the row/column structure of Mosaic Plots can also be found in the biological field, for example, to classify cancer subtypes [192, Fig.3]. Yet, it should be noted that the figure in this paper is not based on an interactive visualization, but specifically produced to present the findings.

The second class of categorical representations use **explicitly encoded relationships**, where ribbon-like links between the dimensions and categories are predominant. The ribbons typically connect different categories of dimensions, where their width encodes the frequency of the interaction between the connected categories. An early example is the *CobWeb* technique by Upton [189], shown in Figure 3.14(a). The two dimensions to be compared are arranged on two sides of a circular layout, with each category shown as a small circle. Ribbons are drawn between the categories of the different dimensions. Upton also gives examples with multiple dimensions shown at once. However, again, interactions between more than two dimensions can not be read from CobWeb diagrams, albeit this would be possible using brushing. Also, CobWeb diagrams, as displayed in Figure 3.14(a), do not visually encode the magnitude of the different states, but provide only numbers.



(a) (b) (c)

Figure 3.14: Category visualization using ribbons. (a) The *CobWeb* technique. The dimension's categories are grouped and placed in a circular layout. Ribbons of different width connect the categories of the different dimensions [189]. (b) Revised version of *Parallel Sets* in *standard mode*, showing the Titanic dataset [103] (c) Original version of Parallel Sets in *bundled mode*. In comparison to (b) the ribbons between the second and the third dimension are not split based on the categories of the first dimension [104].

Parallel Sets, by Bendix, Kosara, and Hauser [12, 103, 104], draw from a variety of techniques, such as parallel coordinates, mosaic plots and *Sankey diagrams* [147], resulting in a flexible visualization technique for categorical data. Parallel Sets assign each dimension to an axis, which are arranged similar to parallel coordinates, as can be seen in Figure 3.14(b). The categories are plotted as boxes with a height proportional to their frequency. Categories of neighboring dimensions are connected via parallelograms, the width of which encode the amount of shared records. Parallel Sets distinguish between a *standard mode*, in which the ribbons between two dimensions are always split based on the categories of the "focus" dimension, and a *bundled mode*, where the ribbons only consider the interactions between the two neighboring dimensions and which is shown in Figure 3.14(c). While the former makes it easier to understand the splitting of connections, the latter reduces clutter, which is especially an issue when more than three dimensions are to be considered. Parallel Sets clearly show the relationships between two dimensions, but can, in contrast to the techniques discussed previously, also show interactions between multiple dimensions. Taking up the example discussed before (to look for the portion of female crew members who survived the Titanic disaster) it is obvious that the information that most female crew members were saved, is easy to read from Figure 3.14(b). An similar approach is used in *CComViz* for the comparison of clustering algorithms [214].

In this thesis, we also consider categories (derived) from multiple, heterogeneous data sources. However, so far, only few studies have been reported on this topic. One of them is the D-Dupe software [92], which clusters multiple datasets and then matches up the results in a visualization to identify duplicates between both datasets.

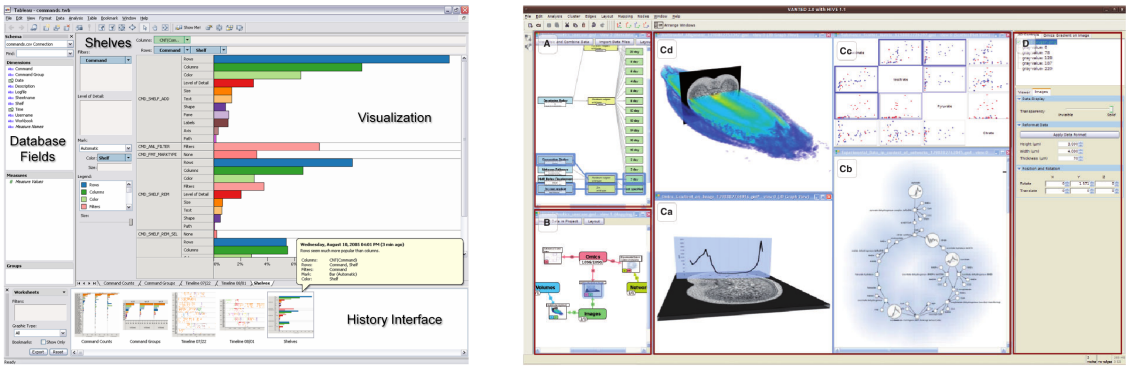
3.4.3 Summary and Context

We found that despite many promising general visualization approaches, the state of the art does not provide a technique for interactive visual subset comparison across dataset boundaries. Since we intend to combine visualizations of underlying data with the encoding of categories, data space techniques are not suitable for our tasks. We have decided against employing relative positions, since they do not integrate easily with embedded visualization due to unfavorable aspect ratios for smaller categories. Hence, we chose a ribbon-based technique, which scales better in this regard. For our ribbon-based technique, we prefer the parallel approach, where the ribbons are not split based on a focus dimension. The reason for this decision is the interactive nature of the proposed visualization techniques, where we expect analysts to frequently change their focus dimension. Also, we are interested in analyzing more than three dimensions at a time, at which point the splitting produces too much clutter.

3.5 Heterogeneous Data Visualization

We previously discussed techniques which can be used to visualize heterogeneous data, for example, the portal approach [207], or the brushing between multiple datasets [35]. In this section we give a brief overview on the state of the art in heterogeneous data visualization that goes beyond the aforementioned techniques. For a detailed analysis of the literature on heterogeneous data analysis refer to the thesis by Streit [176]. Heterogeneous data analysis is challenging because of two reasons: First, conducting a targeted analysis is difficult because of the overwhelming options analysts have. And second, the different datasets require different views and visualization techniques, which need to be integrated [183, p. 11]. The difficulty of an analysis is caused by the multitude of visualization techniques and the many datasets. Just assigning which type of view should show which dataset is by itself already challenging. Configuring visualization techniques and finding a reasonable sequence to explore the data is even more difficult. Consequently, analysts will greatly benefit if they are provided with orientation (e.g., with a map of the data) or even with guidance. Two approaches that provide orientation support are (a) to provide a history of previous steps, or (b) to provide a map of the data space. The former is a common approach in general visualization systems. Shrinivasan and van Wijk [170], for example, keep records of user actions that make it possible to revisit previous states. Their implementation also supports branches of the analysis path. Heer et al. [68] suggest a system to a similar end, shown in Figure 3.15(a), but uses thumbnails to record and revisit the history. Koop et al. [102] suggest a system that goes beyond orientation, by offering guidance for creating and configuring visualizations. Their approach is an extension to *VisTrails* [11], a provenance-based system that records analysis pipelines for re-use. Maps of the explorable data typically resemble relational database schemes. North et al. [136] extend this idea to views, as they envision *DataFaces*, interactive connections of visualization and data schemes, as future work.

There are several examples for tools supporting the integrated analysis of heterogeneous data. We distinguish between those that work with cross-referenced datasets, and those that handle general heterogeneous data. A prominent member of the first class is *Snap-*



(a) (b)

Figure 3.15: Heterogeneous data analysis. (a) An extension to *Tableau* provides a graphical history [68]. (b) The *HIVE* system combines heterogeneous data analysis with orientation support [150].

Together visualization, which integrates multiple visualization applications, which run as separate processes [137]. The integration is based on a mutually accessible relational database. An extension of the Snap-Together system allows users to interactively create both a data model and a set of suitable visualizations [135]. Lieberman et al. [116] present a system that extends the original network-data only semantic substrates [169] to handle cross-referenced data. The views in the system are, however, limited to scatterplots, connected with visual links. Analysis tools for general heterogeneous data sources are rare. A notable exception is *TimeLine* [5], which integrates multiple datasets, such as CT scans and data from relational databases.

3.5.1 Summary and Context

Heterogeneous data analysis is a complex field with many facets but little literature up to this date. With data being increasingly connected, and problems being of a larger scope, the topic will gain attention in the near future. Two recent publications from the biomedical domain [150, 194], published simultaneously with our contributions (detailed in Chapter 8) show the growing relevance. The work by Rohn et al. [150], shown in Figure 3.15(b), is not only a truly heterogeneous data analysis framework, but also provides orientation by means of a map of the data.

None of the systems discussed seamlessly integrates the analysis process with a map of the data. Neither does any of the systems combine guidance with heterogeneous data analysis. In Chapter 8 we show how to do both, and demonstrate its utility. Also, while some systems connect separate applications, none of them are truly independent, as all of them utilize a shared data storage. We believe that heterogeneous analysis scenarios are best addressed with specialized tools that are nevertheless tightly integrated. While we do not provide a solution for this problem, we show some possible directions in Chapter 8.

3.6 Visualization in Molecular Biology

In this section we present a selection of relevant visualization techniques for applications in molecular biology. Typically, the techniques are strongly related to general visualization techniques discussed up to this point. Here we go into detail on several domain-specific techniques. This section is structured by the type of data the discussed techniques address. If an application covers multiple data types, it is discussed where it is most relevant in the context of this thesis.

3.6.1 Expression Profile Data

Expression profile data is typically visualized using clustered heatmaps with dendrograms [38, 54, 159] (similarity trees based on the results of a hierarchical clustering algorithm), parallel coordinates [54, 101, 155] (or profile plots as they are referred to in biology publications) and scatterplots. All these visualization techniques were treated in Section 3.1. Microarray analysis and visualization platforms are widely available. Prominent examples are the *MultiExperiment Viewer* in *TM4* [156] or Mayday [10, 32, 54]. *Mayday* is a visualization tool for expression profile data focusing on the integration of meta-information to annotate expression data, and on the integration of analytic capabilities, for example, by providing interfaces to *R* [143] and *Weka* [61]. Mayday uses a plug-in architecture to make it easily extensible.

The key challenge when using heatmaps is to come up with a useful and biologically relevant ordering, which is usually based on clustering algorithms. Clustering of genes ideally creates groups of co-regulated genes, which indicate co-function [38]. It is also common to cluster samples or experiments, if no other semantic ordering (for example, ordering in a time-series experiment) is given [200]. Clusters of samples can indicate clinically relevant shared characteristics, such as tumor subtypes.

Heat maps are very popular in biomolecular visualization, with about 4000 publications to date containing heatmaps [200], and the original paper by Eisen et al. [38] being the third-most cited paper of the *Proceedings of the National Academy of Sciences* [204].

The *Hierarchical Cluster Explorer (HCE)* by Seo and Shneiderman [159] is a tool for interactively exploring hierarchically clustered heatmaps. Core features of HCE are a dynamic approach for partitioning the data into clusters by adjusting the "cut" of the dendrogram and a focus+context approach based on multiple views for heatmaps. An overview, containing all records, is supplemented by a detail view in a separate window, which only shows a selected sub-set. An enhancement for HCE introduces algorithmic ranking of projections [161], the result of which is displayed in a matrix. Based on this ranking, users can then explore the most likely relevant projections using statistical views such as scatterplots or histograms. HCE also supports comparing the effects of two different clustering algorithms on the same dataset. It renders two heatmaps on top of each other and draws straight lines between the related items. While Seo and Shneiderman state that this basic implementation was already very helpful for their users, they also notice that simply crisscrossing lines can cause confusion for the users. Furthermore, they only show this feature for very small datasets (less than 50 records and 6 dimensions).

The typical analysis of gene expression is conducted with samples where no spatial

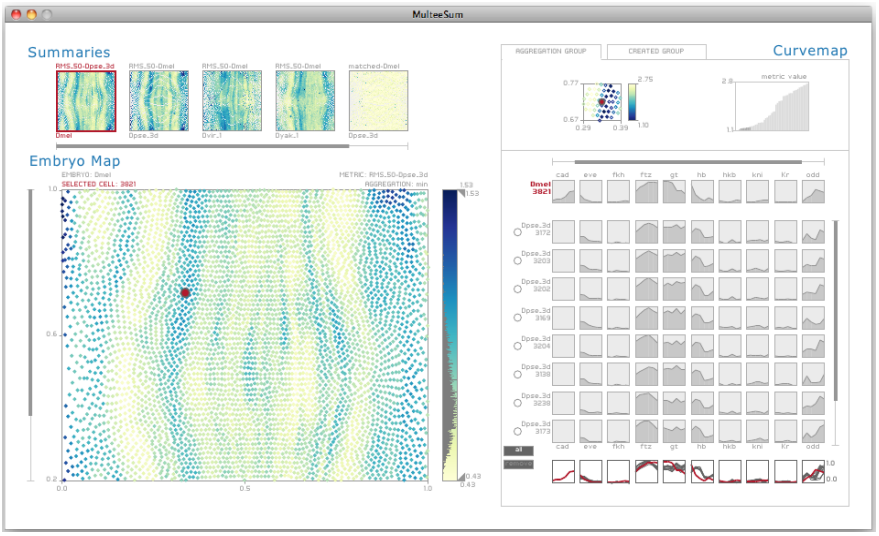


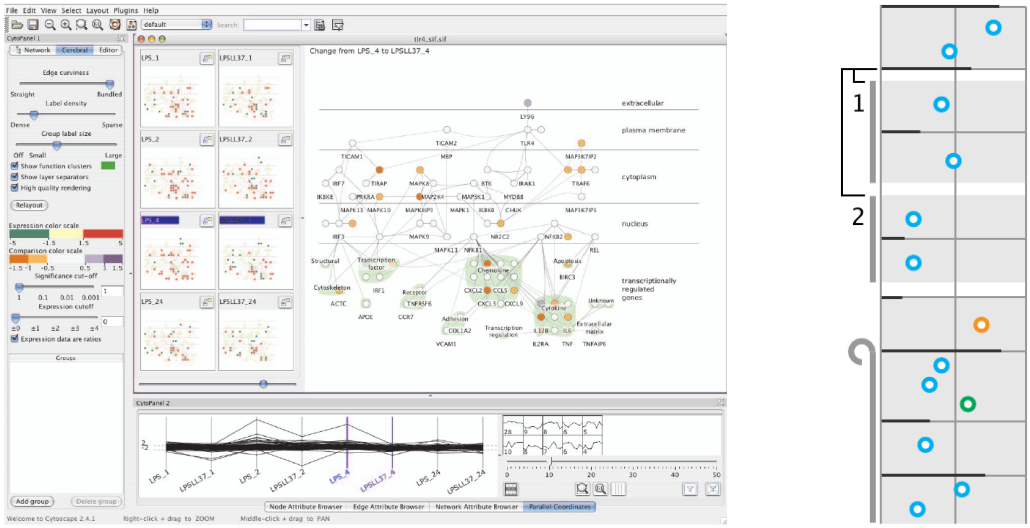
Figure 3.16: *MulteeSum*, a tool for visualizing spatially referenced gene expression data. The view on the left is a projected map of an embryo. The circles show the positions of cells where expression data is available, and their color encodes the magnitude of a summary expression score. The *Curvemap* view uses filled line-plots to show expression values for selected cells [127].

information is recorded. An alternative approach is used when the spatial location is relevant, for example, in the early stages of the development of an embryo, when the future function of cells is determined. Two notable tools in this context are *PointCloudXplore* [154, 155, 199] and *MulteeSum* [125]. The latter is shown in Figure 3.16. The scale and type of data differs from typical expression profiles in that the number of genes is significantly smaller, but measurements are taken for thousands of spatially registered cells and several time points. Both tools provide a map of the embryo with color coded points symbolizing the cells. PointCloudXplore and MulteeSum differ in the tasks they address. PointColudXplore uses parallel coordinates in 2D and 3D versions to show the expression data. MulteeSum contains the *Curvemap* view, which shows a matrix of small line-plots, where one axis of the matrix contains the genes, the other the selected cells. The line-plot encodes the expression level over time. Meyer et al. [127] originally developed the Curvemap view for *Pathline*, a tool for comparing gene regulation among species, which we will discuss in the context of pathway visualization.

For a comprehensive overview of expression profile analysis tools as well as of pathway visualization (covered in the next section) see the survey by Gehlenborg et al. [55].

3.6.2 Pathways and Protein Interaction Networks

Graphs are a common data form in biology. Examples are genealogies and phylogenetic trees. In molecular biology, two types of graphs are especially important: protein-protein interaction networks and pathways. Both have also been the subject of substantial visualization research.



(a) (b)

Figure 3.17: Mapping experimental data onto pathways. (a) *Cerebral*, a *Cytoscape* plugin, shows small multiples for experimental conditions, with data for one selected condition shown in the large graph view. The magnitude of the experimental values is color-coded onto the nodes. A parallel coordinates view for the experimental values complements the pathway views [9]. (b) Linearization of a pathway, where experimental values are encoded using circles and bars. This makes it possible to employ position, the most powerful visual variable, to encode the magnitude of the values [127].

Protein-protein interaction networks essentially capture the chemical interactions and bindings of proteins. The binding typically is part of their biological function. The possibilities of interactions are huge. In yeast, which is considered a simple organism, more than 20.000 interactions are estimated among the 5000 gene products [55]. Protein interaction networks are stored in graphs. When studying protein interaction, automated layouting of the network is often employed, due to the overwhelming size of the networks. *Cytoscape* [162] is one of the most frequently used tools to visualize this kind of data. Cytoscape owes its popularity largely to its plug-in mechanism, which makes it easy to extend. Many Cytoscape plug-ins exists for diverse visualization and analytical tasks. Some of them, like *Cerebral* by Barsky et al. [9], shown in Figure 3.17(a) are also interesting from a visualization perspective. Barsky et al. discuss a graph layout algorithm which takes biological properties into account.

In the context of this thesis, however, the way Cerebrals handles the overlay of multiple gene expression experiments on the nodes is most interesting. It does so by utilizing small multiples of one larger version of the graph, where the nodes color-code the magnitude of the expression. Cerebral also provides a parallel coordinates view to explore the expression data.

Pathways are biochemical processes which carry out a specific function in a cell (see Chapter 2 for details). Pathways typically are hand-curated and stored in publicly available databases. Prominent examples are *KEGG* [91] and *BioCarta**. The focus of analyzing pathways is different from protein interaction networks: Instead of exploring protein-interactions per se, users of curated pathways are typically interested in functional implications of a particular gene, in, for example, an experimental condition. It is therefore not surprising that plenty of methods, asides from the aforementioned small-multiple approach, have been developed to do this. An alternatives to the small multiple approach is mapping multiple experiments onto the node. Possible on-node encodings are multiple, color-coded glyphs [117, 130], animation of the color code [93], bar charts [89, 202], sometimes with error bars, or line plots [79]. Meyer et al. [127] take a completely different approach: they linearize the pathway layout and plot aggregate expression values using lines and points, as shown in Figure 3.17(b). This lays a stronger emphasis on the comparison of the numerical values, while abstracting the information of the network.

3.6.3 Genomes and Sequence Data

Genome visualization is not covered in this thesis. Some techniques used for genome visualization are nevertheless relevant for visualizing relationships in the context of our divide and conquer approach. Especially tools for comparative genomics data are of interest, as they typically use visual links to encode relationships. Two layouts are prevalent: linear layouts, for example, used in the *Ensembl synthenyview* [28], where the datasets to be compared are aligned side by side, and circular layouts, where segments of circles contain the different datasets. Meyer et al.'s *Mizbee* [126] and Krzywinski et al.'s *Circos* [107] are examples of the latter group, which are also interesting for their advanced visual encoding. They both use bundled curves respectively ribbons to show relationships and differences among genomes. Mizbee uses two circles of chromosomes, one for the genome of each species to be compared. The selected chromosome of the outer circle is copied to the inner ring, and curves are drawn between the location of conserved regions in this one chromosome and all other chromosomes in the target species. Consequently, only relationships of one source chromosome to the target's chromosomes are shown at a time. Additionally, an enlarged rectangular detail of the source chromosome and a detailed view comparing the source and a target chromosome is provided. Circos [107] can place several datasets in concentric rings and show position changes with curves connecting the rings. However, this method does not scale to many changes in position, which is why, alternatively, chromosomes from different samples can be arranged on a single circle. Circos is however a tool that produces only static plots. These and other approaches to genome visualization were summarized in a recent review article [133].

In contrast to MizBee, we intend to compare multiple datasets or dimension groups at a time. Also, Mizbee uses spatially separated detail views, where we think a tight integration with the overview is crucial to make the relationship between overview and detail more obvious. Using a multi-circular layout, as it is possible in Circos, for analyzing multiple datasets or dimension groups would result in heavy over-plotting.

*`http://www.biocarta.com/`

3.6.4 Summary and Context

Visualization plays a key role in molecular biology research, as the vast amount of techniques and publications clearly show. Many techniques developed for these tasks can also be applied to other fields. An example is the HCE, which has also been used to analyze demographic data [160]. Other tools have a very specific focus, but solve an important domain problem well (e.g., Pathline [127] or PointCloudXplore [155]). With the techniques described in this thesis, we aim to do both. The central concept and the fundamental techniques are of general value and are not limited to a particular use case. Some other contributions, such as the integration of pathways, are only relevant for the specific domain they are made for.

Chapter 4

Framework

Contents

In this chapter we describe Caleydo, a visualization framework for molecular biology developed at the Graz University of Technology. Caleydo has been developed since 2006, with the first public release of the Software in 2009. Caleydo is intended to address two needs: to give biologists a tool that they can actually use for an analysis, but also to be a platform for research and teaching in visualization.

All the visualization techniques presented in this thesis are either part of or employ Caleydo. In this context, we first discuss several basic visualization techniques provided by Caleydo. We continue with a discussion of Caleydo's data filtering capabilities in Section 4.2. The chapter is concluded with a brief discussion of several software design decisions relevant to the divide and conquer visualization approach and the handling of heterogeneous data.

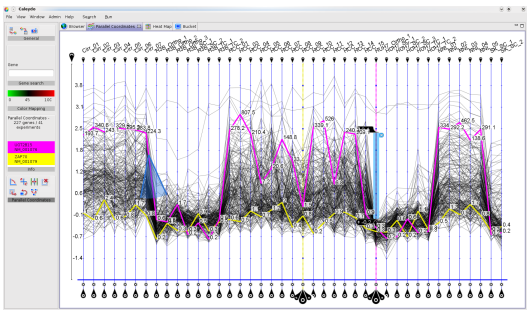
4.1 Fundamental Visualization Techniques

In this chapter we discuss several visualization techniques Caleydo provides that are not part of the divide and conquer concept. In many cases, however, these are the building blocks for the techniques realizing the concept.

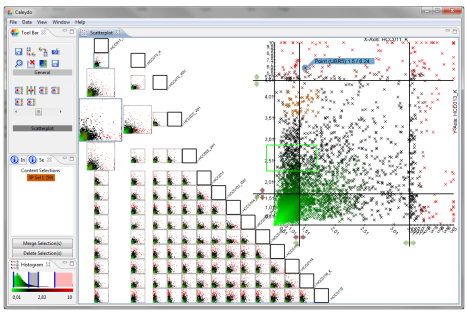
Caleydo provides three classes of visualization techniques for multidimensional data:

1. Those that encode the data directly.
2. Those that show a similarity of entries according to a hierarchy determined by a clustering algorithm.
3. Those that show abstract statistical properties of a dataset.

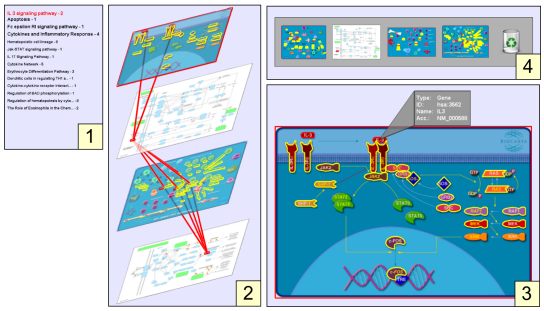
Representatives of the **first class** are implementations of parallel coordinates, shown in Figure 4.1(a), a scatterplot matrix, shown in Figure 4.1(b), and a heatmap. Both,



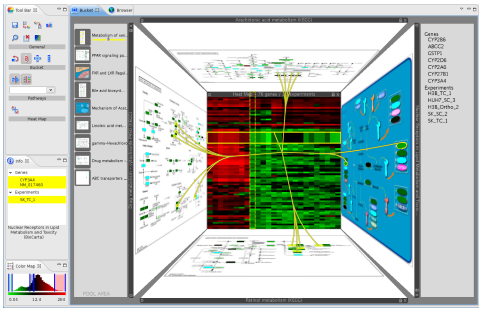
(a)



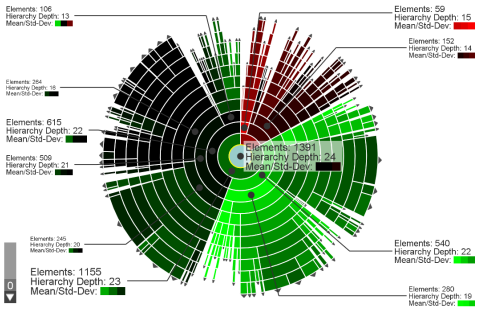
(b)



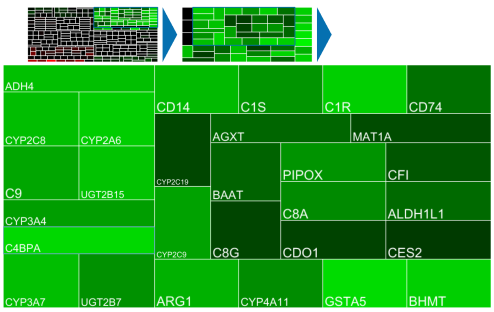
(c)



(d)



(e)



(f)

Figure 4.1: Various visualization techniques of Caleydo. (a) A parallel coordinates view. (b) A scatterplot matrix implementation. (c) The *Jukebox* for visualizing pathway interdependencies [177]. (d) The *Bucket* view for visualizing pathway and gene expression interdependencies. (e) A *sunburst* and (f) a *treemap* view for visualizing similarity relationships in hierarchically clustered data.

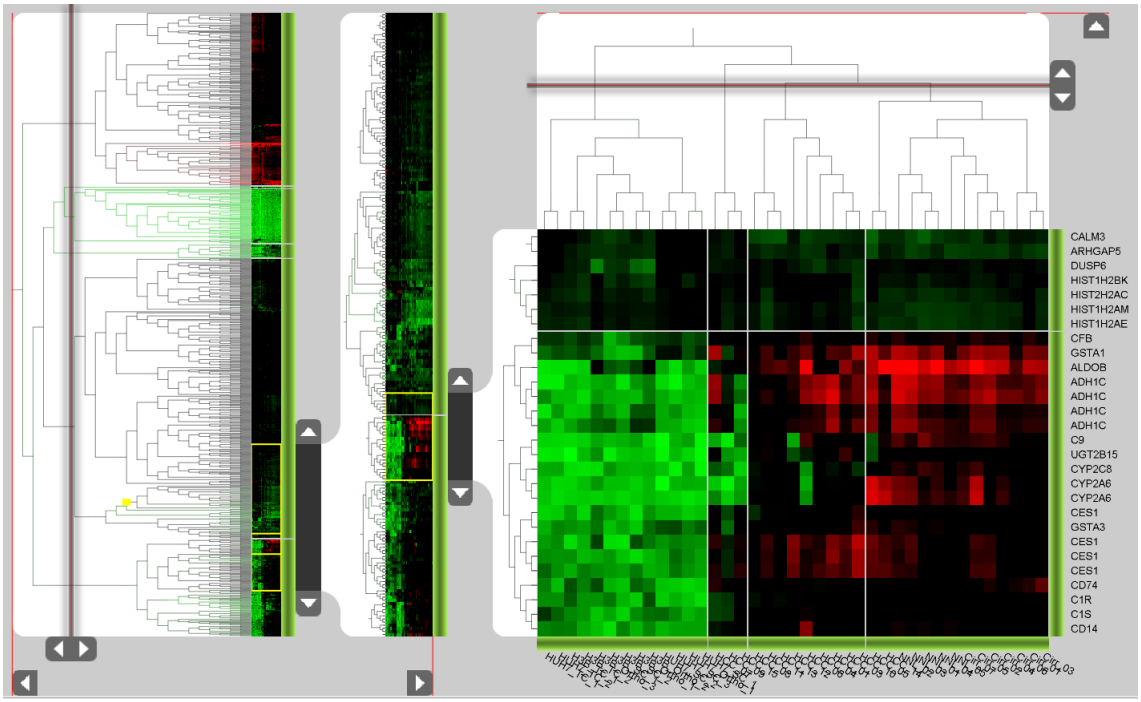


Figure 4.2: *Hierarchical heatmap.* A three level focus and context approach shows an overview heatmap at the left, an intermediate level of detail in the center, and a detailed heatmap at the right. The data shown in the detailed heatmaps is determined by the position of the sliding window. Dendrograms show the similarity relationships between entries.

parallel coordinates and scatterplot matrix provide fairly standard features and are therefore not discussed in detail here. The heatmap, however, has several improvements over traditional heatmaps worth mentioning. The hierarchical cluster explorer by Seo and Shneiderman [159] provides an overview heatmap, including a dendrogram, to show similarities between the entries, and a detailed heatmap which shows a selected sub-set of the overview. However, the relationships between the overview and the detailed heat map are not explicit, and transitions between different focus subsets are abrupt. To address this, we developed the *hierarchical heatmap*, shown in Figure 4.2. The hierarchical heatmap employs a direct visualization of the relationship between overview and detail, as well as smooth transitions between focus levels, which are known to aid user orientation [69]. Our approach is conceptually similar to the source code visualization developed by Ball and Eick [8]. Up to three heatmaps show different levels of detail for a multidimensional dataset. The leftmost shows the whole datasets, where global trends are recognizable. Of course, the quality of the overview depends on the quality of the clustering of the heatmap, as only clustered heatmaps can convey large-scale trends. A sliding focus window connects the leftmost to the second heatmap, which shows about 100-200 entries. Here, local trends are recognizable, but the size of individual entries is still too small to see details and print labels. Therefore, the second heatmap is connected with a third, again using a sliding focus window. In the third heatmap individual details, including labels, are clearly

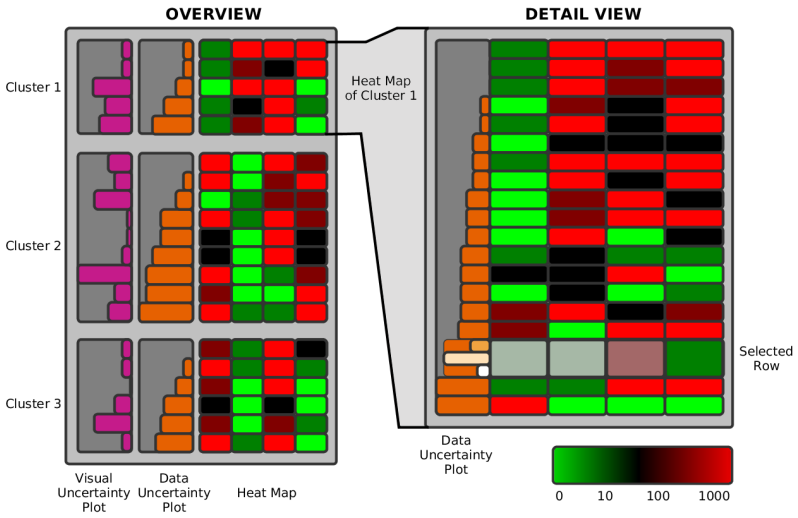


Figure 4.3: Concept for encoding uncertainty in the hierarchical heatmap. Visual uncertainty (due to overplotting) is shown as magenta bars on the far left. Data uncertainty is shown as orange bars next to the heatmaps. For selected records in the detailed heatmap, the different sources of uncertainty are shown individually. The uncertainty of a particular cell is encoded using transparency.

visible. If the scale of the data makes a three-level approach unnecessary, only two, or only a single heatmap is shown.

We also developed an extension of the heatmap to encode **uncertainty** in the data. The approach is illustrated in Figure 4.3. We distinguish two types of uncertainty: visual uncertainty, which is introduced during the rendering process of the visualization pipeline, and data uncertainty. **Visual uncertainty** deals with issues of over-plotting. We quantify how much information is lost due to the inability to map every data entry to at least one pixel and plot this as bars (magenta in Figure 4.3) next to the heatmaps. **Data uncertainty** can have multiple sources. We again use bars to plot an aggregate of multiple uncertainties. In the detailed heatmap, we plot the different types of row-wise uncertainty embedded in an overall bar and use transparency to encode the uncertainty of individual cells. For details regarding this technique we refer to the original paper [78].

Besides traditional multiple-coordinated view (MCV) setups Caleydo also provides the *Jukebox* [177] and the *Bucket* techniques [113], shown in Figures 4.1(c) and (d). Both techniques place views in a 2.5D scene, i.e., a 3D setup with restrictions on where elements (views) can be placed. While the Jukebox is limited to pathway data, the Bucket can be used with arbitrary visualization techniques. In the Bucket, a focus view is placed at the bottom of an open cube (hence "Bucket"). The walls contain contextual views, which take up only little screen-space due to the three-dimensional distortion. Both techniques connect related entities in the different views with visual links. The Bucket can be flattened so that details of the focus view can be explored.

The hierarchical heatmap in Figure 4.2 is combined with a member of the **second class of visualizations – those that show similarity relationships** – a dendrogram, for

both records and dimensions. The dendrogram has several functions: aside from encoding the similarity relationships between entries, the edges' colors also encode the average values of the associated entries. Additionally, the dendrogram is used to interactively determine the grouping of the entries with the cut-slider. Once a cut is made, the grouping is shown using green, shaded bars next to the heatmaps. To save space, the dendrograms can be collapsed to show only the tree above the cut. Caleydo also provides a sunburst [173] and a treemap [87] implementation, shown in Figures 4.1(e) and (f), to visualize the hierarchical relationships between the entries.

A histogram is the most prominent representative of the **third class – visualization techniques that show statistical properties**. The histograms in Caleydo serve two purposes: to show the distribution, and to work as the color legend for the heatmaps. The histograms can also be used to change or adjust the color mapping of a dataset. Other statistical views are, for example, aggregate heatmaps that show mean values and the standard deviation of the aggregate. Those views are, however, not used as stand-alone windows and are therefore introduced later.

4.2 Data Preprocessing, Filtering

In this section, we will briefly explain how Caleydo handles the first two processes of the visualization pipeline [36]: **data analysis** and **filtering**. The first process, data analysis, means data preprocessing – a term we prefer, as, in the wake of the field of visual analytics, analysis is thought of as a repeated process in the data exploration. Preprocessing tasks include smoothing, correcting errors or handling of missing values [36]. Caleydo does not include features for preprocessing, but assumes that the data is ready to be analyzed. The rationale for this is that, especially in molecular biology, the preprocessing is in most cases done by the software used for reading the data, or by established R-scripts [143]. While Caleydo does not handle missing or invalid values in the data on a preprocessing level, it reserves special mappings in all visualization techniques and handles it in all algorithms.

The second step of the visualization pipeline, **filtering**, is an essential process, since it enables users to remove the unimportant, or the out-of-focus data. Filtering is also an important part of Shneiderman's information seeking mantra - "overview first, zoom and filter, then details on demand" [168]. Caleydo provides various ways to filter data, which can be classified into algorithmic filters and visual filters. The former class includes filters based on significance (e.g., using t-tests) or fold changes, the latter is based on brushing techniques such as 1D-selections or angular brushes in parallel coordinates.

In most cases, filtering is employed sequentially, which corresponds to a logical AND combination of filters. However, other combinations such as OR and XOR are important as well. Logic combinations of similar operations have been published mainly for brushing, which is closely related to filtering. An early case study of a system that uses logically combined brushes is the *XmdvTool* by Martin and Ward [123]. An important use case for OR combination from the field of molecular biology is to filter expression data, which increases or decreases less than two-fold among experiments, effectively removing all data that remains largely unchanged over experiments.

Most visualization frameworks support filtering, some also track the process and make

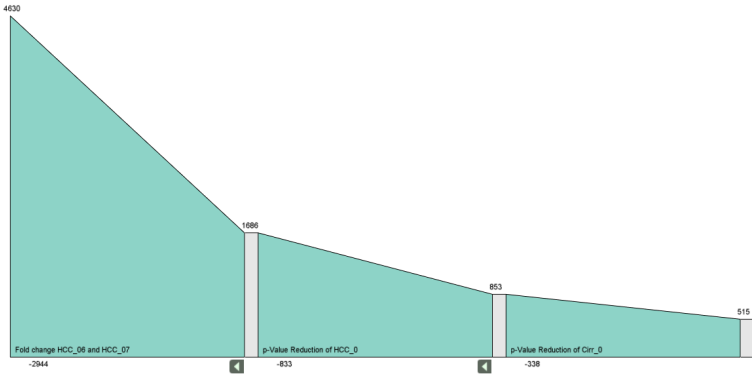


Figure 4.4: Visual representation of three AND-combined filters. Each parallelogram corresponds to a filter. The height of the left base of a parallelogram encodes the amount of data before, while its counterpart on the right shows the amount of data after the filter operation. As the output of one filter is the input of the next, a sequential arrangement suggests itself.

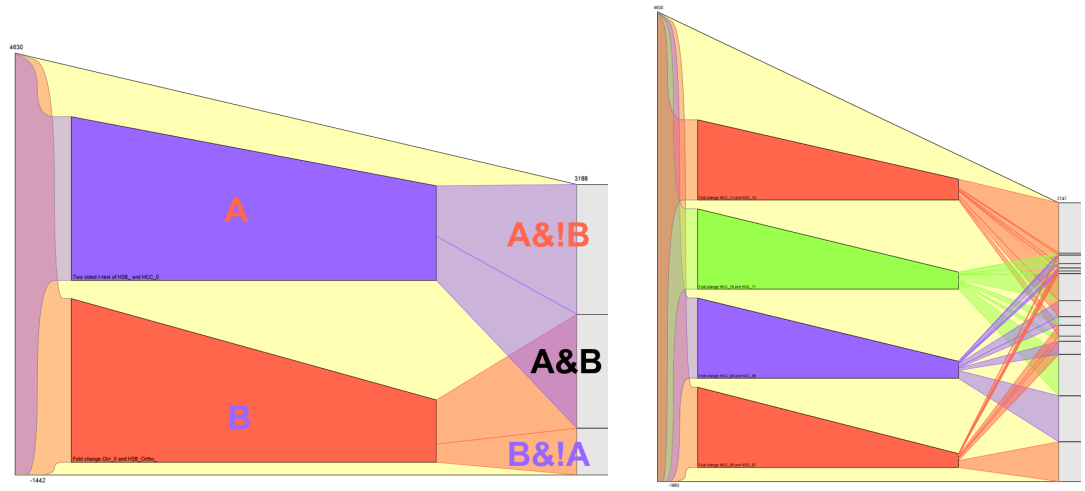
it easily reversible, but none visualize the effect of the filters on the data. This is surprising, since many visualization techniques do not convey the amount of data they show faithfully. If sampling is used, for example, it can be hard to grasp the amount of data shown or to detect the effect of a filter operation.

Filtering is also important as some algorithms or visualization techniques have preconditions as to the amount of data they can handle. Advanced clustering algorithms, for example, have considerable computational requirements. If online clustering, i.e., clustering that can be run in a time-frame where it is reasonable to expect a user to wait, is desired, the data size may not exceed certain thresholds.

Founded on those observations and arguments, we elicited a list of requirements for visually representing filters. A filter visualization technique should be able to (a) show compositions of multiple filters and to (b) show the consequences, the effects of each filter and of filter compositions. We also elicited further, minor requirements, which are omitted here for brevity but can be found in the original paper [57].

To address these requirements, we have introduce the *filter pipeline*. The visualization technique for sequences of filters (AND-combined) is shown in Figure 4.4. Inspired by Minard's work, the famous *Carte Figurative des pertes successives en hommes de l'armée française dans la campagne de Russie 1812-1813* [186], which shows the continuous decline in Napoleon's army during his Russian campaign, we chose to represent a sequence of filters as a sequence of trapezoids, where the length of the left base represents the amount of data before the filter operation, and the length of the right base represents the amount of data left after the filter operation. As in a sequence of filters, the output of one is the input of the next filter, a sequential arrangement is appropriate. The filters have labels showing the amount of data before and after the filter operation, as well as a label describing the filter.

For OR combinations of filters, we developed the technique shown in Figure 4.5(a). Smaller trapezoids, A and B in Figure 4.5(a), are inscribed into one larger trapezoid. The small trapezoids represent the individual filters, the large, enclosing one encodes the



(a) (b)

Figure 4.5: OR-Combined filter representation. The individual filters are nested inside an overall filter which represents the combined effect of all filters. The nested filters connect to a bar on the right where set-intersections of all filters show how much which filter contributes to the overall effect. The example in (a) shows two filters, A and B, which have three intersections on the right. (b) A more complex example with four nested filters.

overall effect. To show that each of the sub-filters operates on the input of the overall filter, they are connected to the left base of the overall filter by transparent ribbons. How much a filter contributes to the overall effect is encoded by a bar attached to the right base of the overall filter, to which the nested filters connect. The bar represents all set intersections of the filters as segments. In Figure 4.5(a), the segments are labeled *A&!B* for elements that only Filter *A* removes, *A&B* for elements both filters would remove, and *B&!A* for those that only *B* removes. As the number of intersections grows exponentially with the number of filters, and also the space for nested filters is limited, this method does not scale arbitrarily. We believe, however, that in typical use cases the number of OR-combined filters is limited. Figure 4.5(b) shows a case with four simultaneous OR combinations, which is still very usable.

The filter pipeline has several additional features, especially concerning the interaction with multiple filters. Examples are methods to change the order, create OR-Combinations with drag-and-drop, hide, and modify filters. For a detailed description, we again refer to the original publication [57]. We discuss an extension of the filter-pipeline considering non-binary brushes in a recent paper [78]. The basic idea is to filter based on uncertainty of the data, i.e., to remove highly uncertain parts. Instead of setting a binary threshold, we use an interval of "certain" data extended by a range of data that cannot be considered certain, but may still be valuable. These two levels of certainty are represented separately and only data not fulfilling the "very uncertain" criterion is removed.

4.3 Implementation and Software Design

Caleydo is written in Java and uses the Eclipse Rich Client Platform (RCP)* for graphical user interface (GUI) components. The software is based on plug-ins, where views and other other components are realized independent of the core. This makes the aforementioned dual use-case possible: some plug-ins are considered stable enough for public release, some are research prototypes, while others are student projects. The visualization techniques use OpenGL†, accessed through the wrapper Library Java OpenGL (JOGL)‡, for rendering. Caleydo includes features that are typically part of multiple coordinated view (MCV) systems, such as linked brushing, filtering, etc.

What distinguishes Caleydo from other MCV systems is its ability to flexibly partition and reconfigure multidimensional datasets, and its power to combine individual views inside a single OpenGL window, both of which are essential properties to achieve the goals outlined in the hypotheses. How this is realized is described in the next section. Following the details on the data structure, we describe how Caleydo resolves mappings among multiple datasets and how layouts for complex views are handled. While many other software design choices of Caleydo would be worth discussing, these explained here are the most fundamental for the presented divide and conquer, multiform, and heterogeneous data analysis approaches.

4.3.1 Data Structure

Figure 4.6 shows the relationships between all classes mentioned in this section in a class diagram. Caleydo loads a single dataset from a comma-separated or tab-delimited file. Parameters, such as how many lines to skip, or which delimiter to use, can either be set in a GUI or supplied in the extensible markup language (XML). The data is loaded into memory and stored as primitive arrays of varying data types in the `Table`, where one column in the dataset corresponds to one array. This corresponds to the *Data Column* design pattern [67]. Using primitive arrays has the advantage of a small memory footprint, but changing the order of elements, filtering, etc., is tedious and slow. To overcome this and to be able to have multiple simultaneous filters and orders we introduce `VirtualArray`s. `VirtualArray`s hold lists of indices to the primitive arrays in the `Table`. In other words, a `VirtualArray` contains "access rules" for the data. By simultaneously creating multiple `VirtualArray` instances, multiple orderings or subsets can be realized. The `VirtualArray` is an advanced data structure based on a list and backed by a hash-map to allow both, store a sequence, and enable constant-time index-of operations. The back-end hash-map is created lazily so that computational overhead is minimal. `VirtualArray`s work equally on dimensions and records. It should be noted that the `Table` abstracts the concept of rows and columns, meaning that a column and a row in a file can each be exposed as both, a dimension and a record by the `Table`.

Other properties of both, dimensions and records, can be introduced by dividing or clustering. These processes create groups of entries that belong together, which is

*`http://www.eclipse.org/rcp/`
†`http://www.opengl.org/`
‡`http://jogamp.org/jogl`

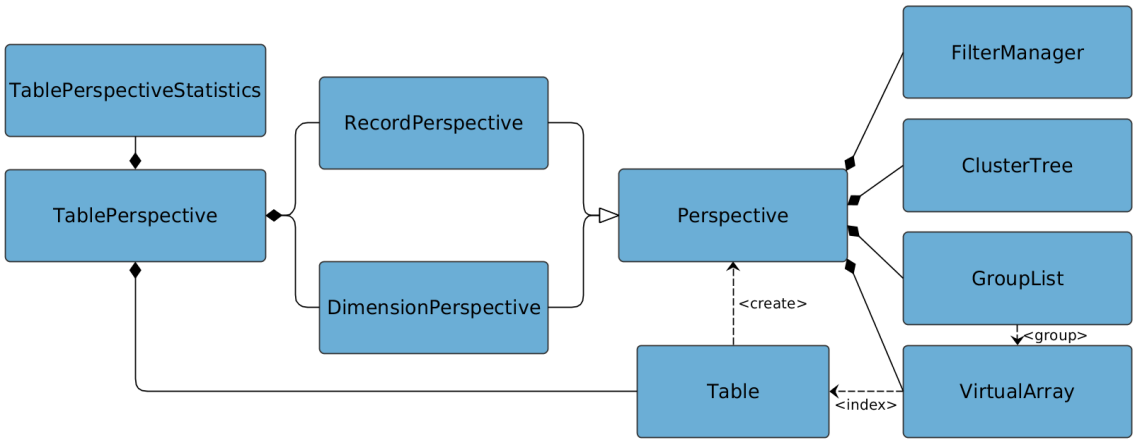


Figure 4.6: Simplified class diagram of the Caleydo data structure. Raw data is stored in `Table`, access rules (with respect to the order of elements, groupings, similarities) are held in `Perspective`. `Perspectives` are defined for either records or dimensions of a dataset. Only a combination defines a subset of the data. Such a combination is held by the `TablePerspective` class.

captured in the `GroupList` data structure (see Figure 4.6). It is the combination of `VirtualArrays`, which can be used to create subsets of datasets, and `GroupLists`, which partition `VirtualArrays` into groups, that we use to realize the divide step in our divide and conquer approach.

The `ClusterTree` is a concept related to the `GroupLists`: it records the similarity relationships of the entries as determined by a hierarchical clustering algorithm. `GroupLists` can be thought of as a cut trough the `ClusterTree`. Together with the `FilterManager`, which holds a history of changes to the `VirtualArray`, all those data structures are brought together in the `Perspective`. Combined they make up a "perspective", a point of view, on either the dimensions or the records of the data.

Figure 4.6 shows that `Perspective` is a super-class of `RecordPerspective` and `DimensionPerspective`. This is a measure to ensure type-safety and is in reality realized for all classes related to either dimensions or records (including the aforementioned `VirtualArray`, `GroupList`, etc.), but is omitted from Figure 4.6 for the sake of simplicity. All the back-end data structures use generics to avoid code-redundancy while all developers of plug-ins are only exposed to the concrete type-safe data types. Caleydo allows users to create perspectives either automatically, manually, or by importing them.

As one perspective encodes information about either dimensions or records, it requires a combination of one `RecordPerspective` and one `DimensionPerspective` to describe an actual subset of the data. This combination is realized in the `TablePerspective`, which holds a reference to both, and to the underlying `Table`. The `TablePerspective` can also be used to calculate statistics on the subsets, such as distributions, mean values, etc. To do this the `TablePerspective` uses the `TablePerspectiveStatistics` class.

Views in Caleydo access the data trough an instance of `TablePerspective` which holds references to the data, the access rules and the meta-information in one place.

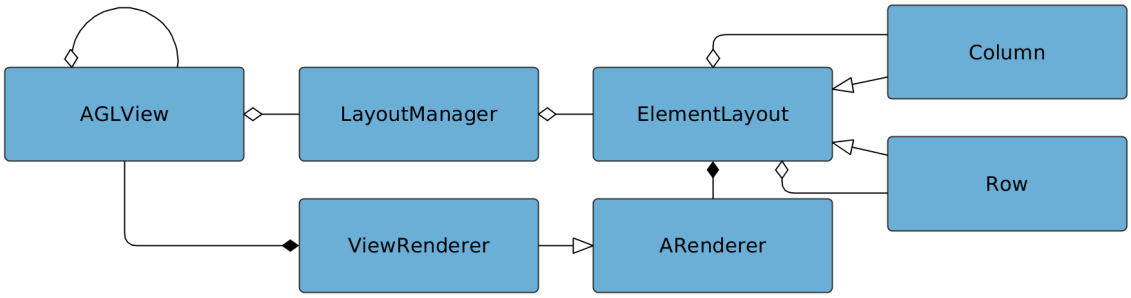


Figure 4.7: Simplified class diagram of the recursive layout management data structure in Caleydo. `AGLView` is the base class of all stand-alone views in Caleydo. It can, however, also be recursively nested, so that one view can contain other views. A `LayoutManager` simplifies structured layouts by providing `Rows` and `Columns` where `ElementLayouts` can be nested. An `ElementLayout` can have an `ARenderer`, which is a base class for rendering OpenGL objects. While an `ARenderer` is the lightweight counterpart to the `AGLView`, it can also render `AGLViews`.

4.3.2 ID Mapping

Molecular biology uses multiple competing annotations for genes and related gene products. Those relationships, as is evident by the definition of a gene in Chapter 2, are not trivial. Missing mappings and multi-mappings in all directions are possible. For the mapping of biomolecular entities, Caleydo relies on information extracted from the DAVID Bioinformatics Database [167]. As, however, real-time querying of the online resources for the desired amount of data is not realistic, Caleydo provides an advanced mapping data structure, which can capture all possible relationships. We integrate the information from DAVID with the identifiers (IDs) from the experimental data provided by a user. The ID mapping data structure is used not only for biomolecular data, but for all kinds of data. This ID mapping allows us to bridge between many heterogeneous datasets. Caleydo provides tools to dynamically convert data structures of one ID type to all other registered ID types, so that cross-dataset filtering or brushing is possible. This makes it possible to, for example, run a clustering on one dataset and apply the resulting grouping and ordering to a cross-referenced dataset.

4.3.3 Layout

We have discussed how the divide step is realized on a data structure level, and how relationships among multiple datasets can be resolved in Caleydo. This section discusses the basis for the visual conquer step.

Caleydo employs recursive nesting to achieve a close integration of multiple views. A GUI-window contains one top-level `AGLView`. The `AGLView` can render content and/or other, nested `AGLViews`, as indicated by the self-aggregation in Figure 4.7. `AGLView` is a heavy-weight base-class for views in Caleydo containing components for tasks such as event handling, picking, and data management. While `AGLViews` can be nested they are also used as stand-alone views in a traditional MCV system.

To simplify the layout-process for complex views or for combinations of views, Caleydo provides a `LayoutManager`, which holds nested `ElementLayouts`. As shown in Figure 4.7, two classes, `Row` and `Column`, are derived from `ElementLayout` and provide row-, respectively column-wise stacking of `ElementLayouts`. By assigning an absolute, relative or dynamic width and height for individual `ElementLayouts`, and by using the nesting in rows and columns, complex layouts can be realized. An `ElementLayout` may have an associated `ARenderer` which can draw OpenGL objects. `ARenderers` are the light-weight counterparts to `AGLViews`. Even so, they depend on a managing `AGLView` and cannot be rendered as stand-alone views. `ViewRenderer`, a sub-class of `ARenderer`, can be supplied with an `AGLView`, which makes it possible to include complex views as part of the layout. `ARenderers` typically render within the bounds of its parent `ElementLayout`, ensuring no overlap. However, it is legal for `ARenderers` to draw beyond their bounds, which makes features such as pop-up overlays easy to implement.

In complex scenarios, hybrid approaches of nested layouts and manually-positioned elements are common. As `ElementLayouts` are aware of their position relative to the top-level `AGLView`, they can be easily connected and integrated with other elements, such as visual links. While a `LayoutManager` significantly simplifies structured layouts, it is, however, not suitable to be used for highly flexible layouts such as node-link diagrams or 3D-layouts.

Chapter 5

Visualizing Relationships of Stratified Subsets

Contents

While a lot of research has been conducted on multidimensional data analysis, most approaches either visualize a dataset as a whole, or algorithmically extract the most relevant aspects using, for example, dimensionality reduction. In many cases, however, multi-dimensional datasets have an additional property that could be utilized to improve the analysis: Meta-data, either explicitly encoded, or just informally known to the user, makes it possible to stratify data into smaller batches, analyze and process it separately and then compare the batches with each other. When discussing analysis methods with our collaborators, we discovered that most of their data has such inherent groupings. For example, they want to compare time-series from different genotypes of a species, or from patients suffering from diverse forms of cancer. The meta-data defines semantically homogeneous groups and consequently makes the whole dataset inhomogeneous with respect to its semantics. This observation is reflected in Hypothesis I, which states "**dividing (stratifying) inhomogeneous, multi-dimensional datasets into homogeneous groups allows analytical algorithms to create better results, thereby making the subgroups more meaningful**." Hypothesis I also describes the challenge arising when dividing a dataset: the relationships among divided subsets of data are lost.

In this chapter, we describe the **Matchmaker** technique, which re-introduces the relationships lost by employing visual links. Figure 5.1 shows the detail mode of Matchmaker for two semantically stratified dimension groups. Overview columns with heatmaps are shown on the far left and right. Between them, several bricks show focus replicates of selected record groups. Relationships between the two columns are encoded by visual links

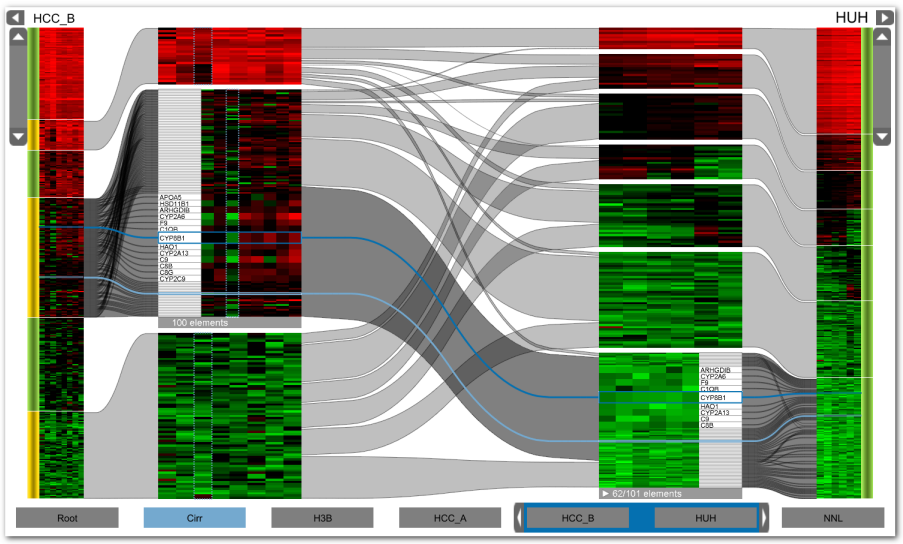
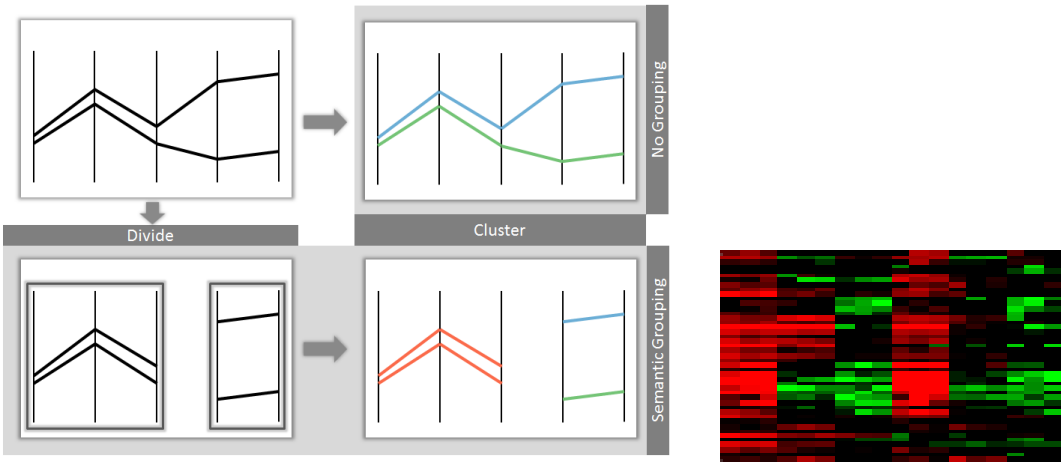


Figure 5.1: The Caleydo Matchmaker detail mode. Matchmaker allows users to stratify multidimensional datasets into homogeneous groups, cluster them separately and analyze the relationships between the resulting clusters.

connecting the bricks. In the remainder of this chapter we will explain the rationale for the design, introduce the visualization technique, and demonstrate Matchmaker's utility by describing two use-cases.

5.1 Motivation and Rationale

In biomolecular data analysis, clustering is used to group multidimensional, high-throughput data into meaningful subsets. For a biologist, the goal of using clustering is to assign a clear biological meaning to clusters. However, clustering, especially of many inhomogeneous dimensions, can conceal important relationships. Figure 5.2(a) illustrates one such case. The two records in the parallel coordinates plot will likely not end up in the same cluster if no grouping is introduced and all dimensions are clustered at the same time. In many cases this is desirable. However, if we know that the first three dimensions are from experimental conditions different from the last two, we can introduce a semantic grouping. This is illustrated in the lower branch of Figure 5.2(a). By clustering these dimension groups separately, the clustering algorithm is likely to assign the two records of the first group to one cluster, but those of the second group are likely to end up in different clusters. Of course, this example is a simplification. The real benefit of such a divide and conquer strategy is obvious when three or more groups of dimensions are created. An example for a complex case is shown in Figure 5.2(b). A typical goal of a biologist would be to find all records that increase over time in one group, and then explore how these behave in the others. This is not possible when all groups are clustered at the same time. If the groups were clustered separately, she could instead compare the source group to all others individually.



(a) (b)

Figure 5.2: Two situations where stratifications can be beneficial. (a) An example for records that are assigned to different clusters depending on whether the dimensions are clustered after they were divided or not. The original parallel coordinates plot shows two records in five dimensions. The top branch of the figure uses no division step; the records end up in separate clusters, as indicated by the different colors. The bottom branch groups the first three and the last two dimensions and clusters them separately. The records for the first group end up in the same cluster, while the records for the second end up in different clusters. The information, which segments of the polylines belong together, is lost. (b) Scrambled, inhomogeneous cluster of eighteen dimensions and six semantic subgroups that were not stratified. No clear biological function can be assigned.

A related problem is the need to compare the results of clustering algorithms. Different algorithms, parameters and similarity measures can have a profound impact on the result. Quality metrics for clustering algorithms are hard to find. Usually, the quality is assessed manually through interpretation by the user. An exception to this are *silhouette plots* [152], which visualize how well objects fit to the cluster they are assigned to. Silhouette plots are based on calculating a measure of how well an element fits to its cluster compared to the next-best candidate cluster. While silhouette plots can help judge the quality of individual clustering results, a method that clearly visualizes the differences between multiple algorithms and parameterizations could support the process of judging the quality of a clustering result significantly. A visualization of cluster stability among several algorithms was developed by Sharko et al. [163]. They use a *cluster stability matrix*, which shows the number of times two genes appear in the same cluster when running different algorithms. To visualize the stability of a cell in the matrix they use color-coding. Sharko et al. employ an indirect approach of calculating and visualizing a metric. We believe that a direct approach of showing relationships of clustering results is preferable. As previously mentioned, Seo and Shneiderman's *Hierarchical Cluster Explorer (HCE)* [159] contains a

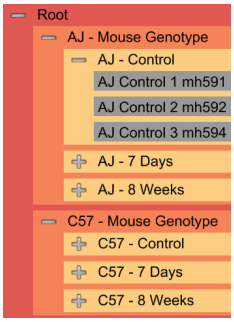


Figure 5.3: Interface for manual, hierarchical grouping. The hierarchy and order is shown in a nested tree representation. Groups from arbitrary depths in the hierarchy can be defined as columns in Matchmaker.

direct visualization of cluster relationships using visual links. However, their method is demonstrated only with a very limited number of entries (less than 50).

To address these challenges, we propose Matchmaker. Matchmaker realizes a comprehensive focus plus context strategy employing details-on-demand and drill-down capabilities for comparing multiple, separately clustered groups of dimensions. To optimize the visual quality of the connections between the groups we introduce an order-preserving curve bundling strategy, which minimizes crossings between clusters.

5.2 Dividing the Data

We treated the formal aspects of dividing multi-dimensional datasets in Section 1.2, where Figure 1.1 also illustrates the process. To briefly reiterate, we stratify a dataset into dimension groups $DG = \{dg_1, ..., dg_u | dg \in \mathcal{P}(D)\}$, and divide the resulting dimension groups individually into record groups $RG_i = \{rg_1, ..., rg_v\}$. We call the visual representation of a dimension group a *column*, and the visual representation of a record group a *brick*. In this chapter, we discuss how this division step can be achieved in Caleydo.

Caleydo provides three ways to stratify entries (records or dimensions): manual, automatic, or imported. For the **manual** approach, which is only feasible for a limited number of entries, we provide a designated interface to facilitate the grouping, which is shown in Figure 5.3. It supports hierarchical stratification into groups on different levels of a tree. The tree is visualized using an implicit tree layout. New levels or branches can be created, duplicated, removed, and resorted interactively. Every level of the tree can be used as a group in Matchmaker. Manual stratification is typically used for dimensions, since the magnitude of dimensions is usually smaller than those of records.

The **automatic** stratification capabilities of Caleydo are based on clustering algorithms. The Caleydo framework provides partitional (e.g., *k-means* and *affinity propagation* [46]) and hierarchical clustering algorithms (e.g., Eisen et al.'s *tree clustering algorithm* [38]) as well as interfaces to *Weka* [61] and *R* [143] to utilize external cluster implementations. As hierarchical clustering algorithms typically only provide similarity relationships among entries instead of designated clusters, the clusters have to be specified through interaction. This can, for reasonably-sized cases, be achieved by using the

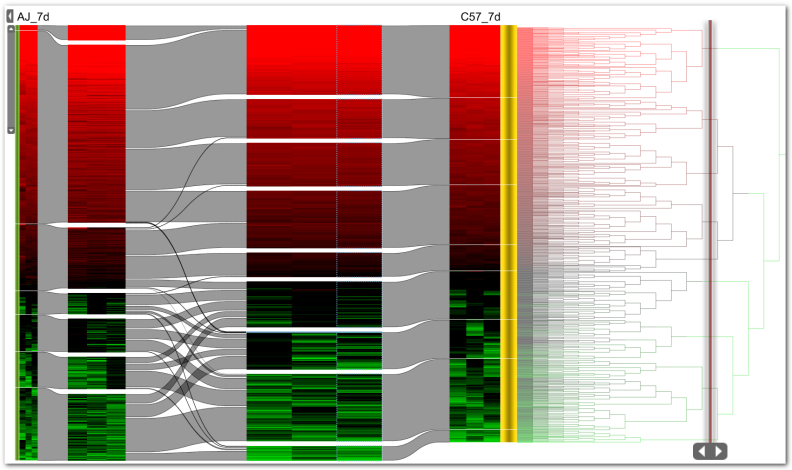


Figure 5.4: Dendrogram used for the dynamic adjustment of the hierarchy cut-off determining the granularity of the stratification. Changes in the cut-off level are immediately reflected in both the overview and the detail heatmaps.

aforementioned interface for manual stratification – the clustering algorithm provides the tree, which otherwise has to be created manually. For larger numbers of entries, a cut-off along a dendrogram can be used to determine the actual clusters. Caleydo uses a default value for the cut-off, which can be modified by dragging a slider to the desired level in a dendrogram in the Matchmaker detail view, as shown in Figure 5.4, or in the hierarchical heatmap as discussed in Chapter 4. Automatic stratification can be employed equally for dimensions and records.

Importing stratifications is relevant to integrate manually curated classifications of datasets, or stratifications created with external tools, such as bioinformatics pipelines. Caleydo uses tab-delimited or comma-separated files, where one column contains identifiers recognized by Caleydo's ID management, and the other column contains group-assignments.

5.3 The Matchmaker Visualization Technique

The visual conquer step in Matchmaker uses relative positions to show that bricks belong to the same dimension group (i.e., they are stacked on top of each other in a column); between the columns visual links are employed. Since clustering algorithms or other stratification approaches typically do not provide an order within a cluster, nor an ordering of clusters, the ordering of bricks (the visual equivalent of record groups), and the ordering of the records within the bricks can be chosen freely. We sort both bricks and records within the bricks according to their mean value and thereby introduce meaning to the position of the records. This allows us to use position to encode information, which is important, since it is the most powerful visual variable available [15]. Having introduced a specific ordering, we can use a parallel coordinates metaphor [81] to make the relationships among columns evident. We arrange the columns side by side, where each column corresponds to an axis

in a parallel coordinates plot. However, instead of using simple lines as axes, we show stacks of bricks containing heatmaps. Analogous to parallel coordinates, we connect the related records in the columns with visual links. This allows us to encode

- the magnitude and patterns of the values by using the heatmaps' color coding,
- the average magnitude of a record group relative to other record groups in the same dimension group via position,
- the average magnitude of a record relative to others in the same record group,
- the relationships among records and record groups across dimension groups using visual links.

As we aim to visualize amounts of data on a scale where a single pixel has to represent more than one value, we face the problem of level of detail (LOD) culling. Fortunately, the clustering automatically aggregates data, so that even if LOD culling occurs, the global trends are still visible. However, our requirements make it necessary to be able to explore the magnitude and the relationships of individual entries. Consequently, following Shneiderman's mantra – "overview first, zoom and filter, then details on demand" [168] – Matchmaker provides an overview, the ability to zoom into arbitrary parts while preserving the context, and interactive, embedded detail views for individual clusters. The detail mode is depicted in Figure 5.1. In both, overview and detail mode, relationships are shown using curves or ribbons. A naive approach for connecting records, however, results in visual clutter, rendering the visualization unusable. Therefore, we developed an edge bundling strategy suitable for our requirements.

5.3.1 Edge Bundling

The most primitive way to show the distribution of records among the dimension groups is to draw straight lines to connect the records, as illustrated in Figure 5.5(a). As discussed earlier, this method does not scale well. Even in small datasets, it is hard to identify trends. Figure 5.6(a) shows the connections between two heatmaps with about 400 records. While at the top the records remain mostly within the same record group, everywhere else crossings between record groups can be observed. It is very hard to see which bricks have stronger, and which have weaker relationships.

One could argue that straight lines work reasonably well in parallel coordinates plots, especially when some clutter reduction methods, such as using transparency, are employed. However, similar to when parallel coordinates are used to display categorical data, the nature of the combination of heat maps in bricks and the parallel coordinates coordinates-like arrangement of columns force an even distribution of axes-polyline intersections. This is not desirable, as it makes "visual clustering" due to coinciding position on an axis impossible.

One possibility to reduce the clutter in the plot would be to sort the records within clusters, since, as stated before, the order within a particular cluster has no a priori meaning. Sorting the data records in the clusters by taking their position in the compared dimension group into account can reduce the number of crossings significantly. However,

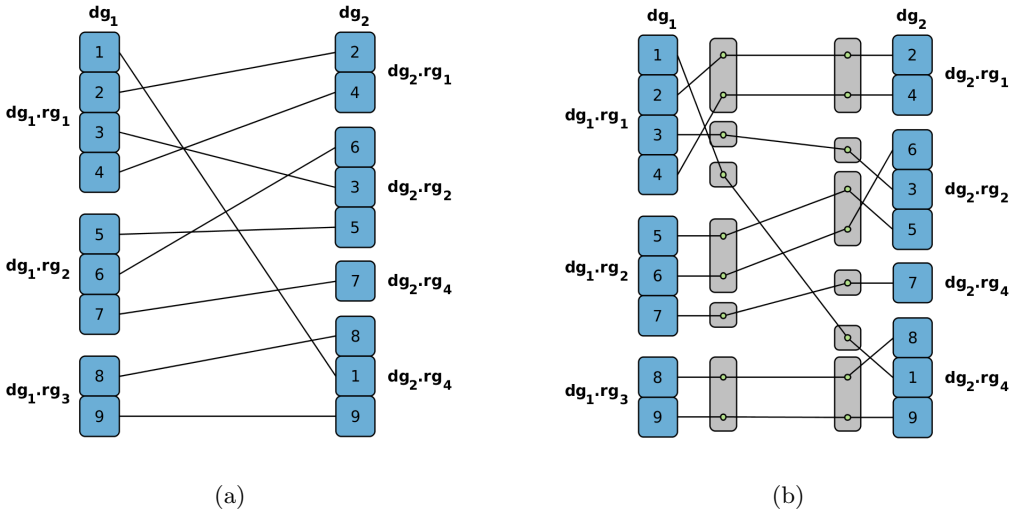


Figure 5.5: Illustration of visual linking between columns with and without bundling. (a) The naive approach using direct connections. (b) Our bundling strategy, where we introduce support points (green) through which the visual links are routed. Support points are sorted based on the destination record group of their associated record. Common destinations of support points are indicated by the enclosing gray boxes. This bundling technique minimizes crossings between the support points of the two columns at the cost of crossings between the support points and their associated records.

since we want to use position to encode the mean magnitude of records, and want to see the relationships among more than two groups simultaneously, this is not an option.

As a consequence, using methods that rely on sorting for crossing reduction, as for example Holten's method for hierarchical edge bundling [77] does, is not possible, even when a hierarchy behind the data is available (e.g., when a hierarchical clustering algorithm was used to produce the clusters). We therefore introduce a bundling strategy that:

- makes use of the grouping of records,
- makes use of the knowledge about the destination position of a record, and
- minimizes crossings of bundles among bricks.

The proposed bundling strategy is illustrated in Figure 5.5(b). For every record in every record group we introduce a support point, shown in green in Figure 5.5(b). Records within a record group can be connected to any of the support points associated with the record group, but never to a support point from another record group. The support points are ordered, so that the topmost support point of the source dimension group (left) is associated with the topmost record group in the target dimension group (right), for which the source record group in the source dimension group has a record. The common target record groups of support points are indicated by the surrounding gray boxes in Figure 5.5(b). In the example in Figure 5.5(b), the topmost source record group ($dg_1.rg_1$)

shares two records with the topmost destination record group ($dg_2.rg_1$). Consequently the two topmost support points of these record groups are connected. The connections from the record group to its support points are chosen so that the crossing between them is minimal. Then the next free support point is considered. If there is another equivalence among the record groups, the target record group's next point will be used. An example is the connection of r_4 in $dg_1.rg_1$ to its equivalent in $dg_2.rg_1$ in Figure 5.5(b). Otherwise, the next record group of the target dimension group is searched for equivalences. If there is one, the points are associated (as for example the connection of r_3 in $dg_1.rg_1$ to $dg_2.rg_2$). This process is repeated until all support points are connected.

As a result, all records from a source record group that connect to the same target record group are assigned to support points that are adjacent in both the source and the target cluster. Therefore, all connections between two record groups are parallel, minimizing the crossings between support points. This technique enables a user to easily identify trends as well as outliers. The main trends produce wide bands, while outliers produce thinner bands. A similar relative magnitude results in bands of small angles, while strong changes in average magnitude between dimension groups result in steep angles. So when, for example, significant changes between two conditions are of interest, then record groups connected by wide bands at steep angles are the feature to look for.

The bundling strategy introduces crossings between the clusters and their support points, making the precise association between records of two groups difficult in overviews of large datasets. However, this can be alleviate by either using interactive brushing, or by using the drill-down techniques provided.

Examples of different connection strategies are shown in Figure 5.6. Figure 5.6(a) uses straight lines and no bundling, while Figure 5.6(b) shows the result of the bundling strategy. The bundling makes the differences between the dimension groups easily recognizable. The exact nature of changes of the clusters is obvious in the bundled case.

We know from the *Gestalt laws* that continuous shapes are perceptually easier to follow compared to discontinuous shapes [201]. Consequently, a further visual improvement can be achieved by replacing the discrete lines with spline curves, as shown in 5.6(c). While this visual representation is already very clear, due to the many parallel curves it can be computationally expensive when used with large datasets and sometimes suffers from Moiré patterns. To address these issues, an abstraction of the individual connection lines by using ribbons is an option. The ribbons are shown in Figure 5.6(d). Matchmaker supports both, using individual curves as well as ribbons, and leaves it up to the user to choose. Ribbons have three advantages over individual curves: there are no Moiré patterns, they further reduce visual clutter and they improve rendering performance. This comes at the cost of hiding the associations of individual elements. To amend this, we employ a details-on-demand strategy: as soon as a user hovers the mouse pointer over a ribbon the contained curves are rendered.

5.3.2 Overview Mode

The overview mode shows the relationships between all chosen dimension groups simultaneously, so that the overall trends in the dataset become visible. Much like regular parallel coordinates implementations, the Matchmaker overview allows to rearrange columns to be

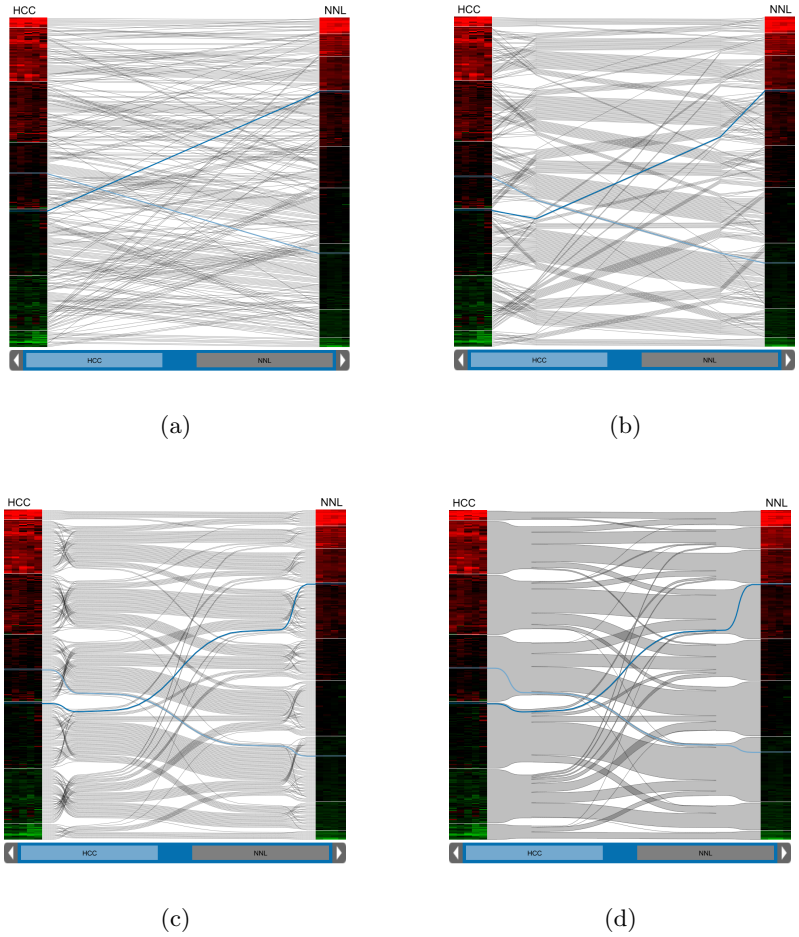


Figure 5.6: The relationships of two columns shown with different visual linking approaches. The straight line rendering in (a), where records are directly connected, produces a cluttered image, even for this relatively small dataset of about 400 records. In (b), we use straight lines, but apply the bundling strategy with added control points on a per-cluster basis (see Figure 5.5), resulting in a much clearer representation with identifiable cluster relations. In (c) the lines are replaced by spline curves for a more continuous picture making them easier to follow. The curves are abstracted to ribbons in (d).

able to compare arbitrary sets of columns, and supports interactive brushing to be able to follow a selection across multiple columns.

Figure 5.7 shows the overview using ribbons as visual links. One record group (orange) and two records (blue) are brushed. For the brushed records, curves are rendered on top of the ribbons. In the overview mode, the spacing between the support points of a brick is reduced, which results in bundled, narrower ribbons and leaves more whitespace. Experience has shown, that the increased whitespace makes it easier to distinguish adjacent ribbons. Matchmaker provides two brushing modes: either using highlight-on-hover,

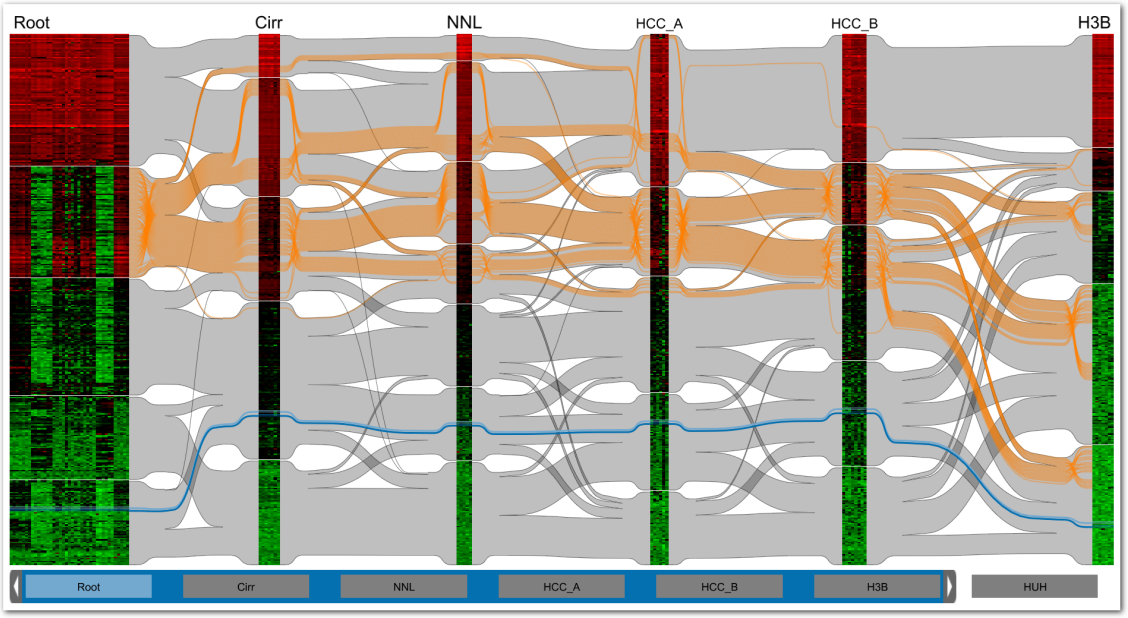


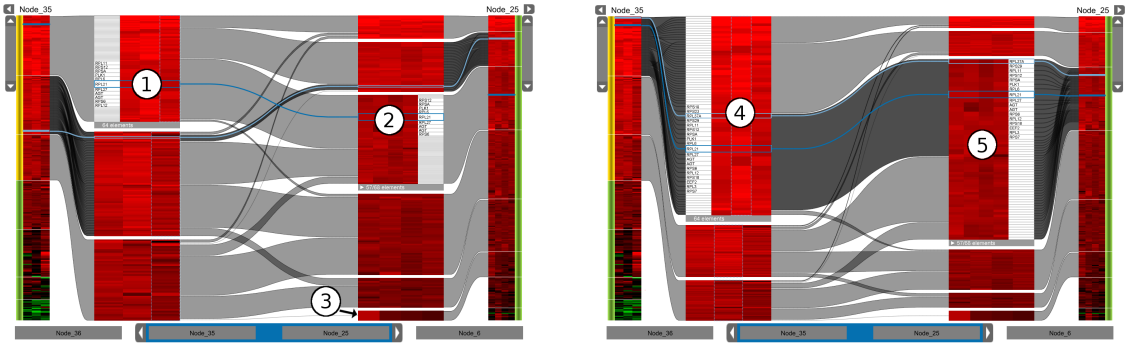
Figure 5.7: Matchmaker's overview displaying 39 different dimensions (78 in total) showing patient and cell line gene-expression data with 400 statistically filtered and clustered genes each. The dimensions are stratified into semantic groups, each group corresponds to a disease. *Cirr*, for example, stands for Cirrhosis. The left heatmap is the root group containing all experiments clustered together. Ribbons connect the columns showing the relationships of the bricks. While the genes in the *Cirr* group are stratified similarly to *Root*, many differences are evident between *HCC_B* and *H3B*. The orange brush highlights all genes selected in the second cluster of the Root group, showing how it spreads over the columns.

volatile brushing or persistent brushing on click. There are three possible scopes of a brush: individual elements, ribbons that connect exactly two bricks, or whole bricks. The brushing is reflected in all of Caleydo's views.

The interaction with the columns is facilitated through a bar at the bottom. The bar always reflects the order of groups, while the blue slider indicates which groups are visible. Interactive rearranging is achieved by dragging the columns' label in the bar to the desired position. Individual re-clustering of a dimension group, for example with different parameters, removing, or duplicating a column, can be trigged using a context menu on the column's bar entry.

In some cases, only a subset of the columns are of interest. Dragging the slider in the bar at the bottom of the overview to include only the desired columns hides the other columns, but their label remains visible in the bar. The bar always indicates which other groups are available, even when they are not visible.

While the overview is able to convey the main trends in the data, for a deeper understanding of the dataset, a drill-down to the level of individual records is necessary. To make this possible, Matchmaker uses a detail mode, which is activated when only two columns are visible.



(a) (b)

Figure 5.8: Different states of the detail mode. (a) The detail view displaying three selected bricks in detail heatmaps and one selected record. Orthogonal stretching is used to be able to show the selected record's label, as can be seen at (1) and (2). Hiding of records in target groups can be observed at (3). Out of the larger bricks to its right, only one element is shown. The other elements are hidden because they do not occur in one of the selected bricks. (b) The detail view showing the same data with orthogonal stretching applied for the selected bricks at (4) and (5). They have more than twice the height compared to their counterparts at (1) and (2) in (a).

5.3.3 Detail Mode

The transition from overview to detail mode is seamless: it can be achieved by either setting the slider in the bar at the bottom to include only two columns, or, for rapid transitions, by using a mouse-wheel action while the mouse cursor rests between two columns. The latter triggers an animated transition removing all other columns, thus making the rapid changes of the layout transparent to the user.

In the detail mode, several GUI elements are added: a shaded bar, located at the outer sides of the heatmaps, allows users to pick individual bricks for detailed inspection. Bars of selected bricks are golden, while others are green. Furthermore, we provide a slider next to the cluster bar, which makes it easier to select multiple adjacent bricks simultaneously. Finally, buttons at the top corners allow the user to slide-in dendrograms, as shown in Figure 5.4, which can be used to refine the granularity of the stratification.

Most importantly, selecting a brick from the columns in detail mode triggers the creation of a focus replicate for the selected (source) bricks and of all target bricks. Target bricks are those bricks in the target column that share at least one record with the selected source brick. Multiple selected bricks are possible. Figures 5.1 and 5.8 show examples with several focus replicates.

Figure 5.8(a) shows a default spacing, where every heatmap has a height proportional to the number of elements it contains. When a record is selected, the heatmaps in the bricks use orthogonal stretching [157] to show the selected record and those in its vicinity in detail, including labels. This enlargement of focus regions is somewhat similar to the

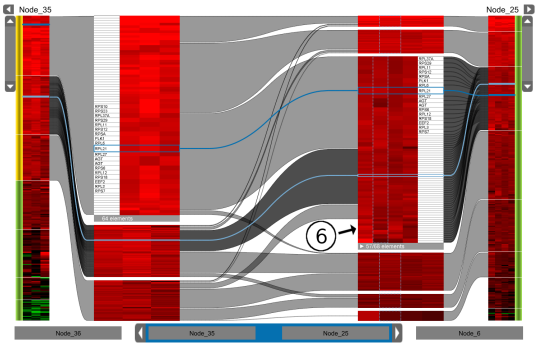


Figure 5.9: The detail mode showing elements that are hidden by default. At (6), the otherwise hidden elements are shown on user request.

orthogonal stretching for areas of interest in *TreeJuxtaposer* [132]. The stretching of records can be observed at (1) and (2) in Figure 5.8(a). Optionally, orthogonal stretching can also be employed for whole focus bricks, enabling a more detailed analysis. An example is shown at (4) and (5) in Figure 5.8(b). When comparing these bricks to their equivalents at (1) and (2) in Figure 5.8(b), it is evident that they are much larger and many more records have labels, as more space is available.

While the replicate of the source brick contains all records, the target bricks' replicates show only the data they share with the source brick; the other records are hidden. Hidden records are indicated by the label in the gray tool-bar below the focus bricks, which is shown when a record is selected or the mouse is hovering over the heatmap. Hiding non-referenced records allows us to show the relevant, referenced records at maximum size. An example where many records are hidden can be seen at (3) in Figure 5.8(a). Here, the target focus brick contains an outlier: only a single record is shown, all other records of the record group are hidden. In some cases, hiding records might not be desirable, therefore hiding can be turned off. An example is shown at (6) in Figure 5.9. The previously hidden records at the bottom of the large heatmap on the right are not connected to records on the left, as there are no corresponding records visible. Showing or hiding can be triggered by clicking the button in the tool bar.

While individual records are rescaled to fit within the current size of the heatmaps, we chose to define a minimum size for a detail heatmap. This ensures that all heatmaps in the detail view are usable and not reduced to only a couple of pixels. If the number of heatmaps is too large to be shown simultaneously, some heatmaps at the bottom will be culled, since they are out of the view frustum. They can be brought back into focus by reducing the number of selected heatmaps.

5.4 Scalability and Implementation

The proposed methods and the underlying implementation perform well for datasets with up to 100 dimensions and up to 2000 data records on standard hardware (e.g., an Intel Core Duo CPU with an NVIDIA GTX 8800 GPU and a 22 inch screen with a resolution of 1680x1050). By default, the Matchmaker view can present up to 10 groups of which

6 can be rendered simultaneously. This was found to be a good compromise between the desire to show more data and the desire to avoid visual clutter. To accommodate unconventional displays, this can be changed in the settings. How many data records Matchmaker can handle largely depends on the number of clusters and the similarity of the groups. Given the described hardware configuration, experiments showed that for about 10 clusters, the technique can handle up to 3000 data records with acceptable visual clutter. However, a larger number of clusters or very different datasets result in a growing number of crossings. Our order-preserving bundling technique produces a readable overview for up to 20 clusters for datasets with less than 2000 records. By using the detail mode for the cluster inspection, the user can analyze many more clusters.

The images in this chapter show a published gene expression dataset [94], except for Figures 5.4, 5.10 and 5.11, which visualize the dataset discussed in Section 5.5. The dataset contains gene expression experiments from patients with different types of cancer and related diseases. The type of disease was used to semantically group the experiments for the comparisons. The color coding for all heatmaps is on a logarithmic scale. While the figures in this chapter show the red-black-green color map prevalent for heatmap visualizations, we also provide perceptually grounded alternatives suitable for users with dichromacy. All other colors for both, the visualization technique as well as the figures, are taken from ColorBrewer [20].

5.5 Case Studies

In the following, we present two case studies of analysis conducted using the Matchmaker technique. The first describes a real-world analysis of a biologist, while the second shows how Matchmaker can be used to assess the behavior and suitability of different clustering algorithms.

5.5.1 Analysis of Gene Expression Data in Steatohepatitis

Our collaborators from the Medical University of Graz study why patients differ in their susceptibility to develop steatohepatitis, which is characterized by inflammation and fattiness of the liver. Steatohepatitis is a precursory disease to cirrhosis. These differences are observed even when exposed to the same amount of steatohepatitis-inducing conditions like alcohol abuse, diabetes or obesity. The reason for this difference in susceptibility to steatohepatitis inducing agents has to be genetic, and the purpose of our partner's experiments are to define genetic regions or modifier genes, which are differentially expressed in these two groups and are responsible for the different reaction to the same causative agent [108].

They use a mouse model of steatohepatitis induction, where animals develop steatohepatitis features, like ballooning of *hepatocytes* (break-down of the cell's skeleton) and *Mallory-Denk-Body* formation (aggregates of misfolded proteins), after being fed with rodent chow supplemented with *DDC* (3,5-diethoxycarbonyl-1,4-dihydrocollidine) for 8 weeks [63]. Our collaborators identified two mouse strains (genotypes), *A/J* (AJ) and *C57Bl6/J* (C57), which show distinct phenotypes upon DDC feeding. By histological analysis of liver tissue, it is possible to determine that AJ mice develop steatohepatitic

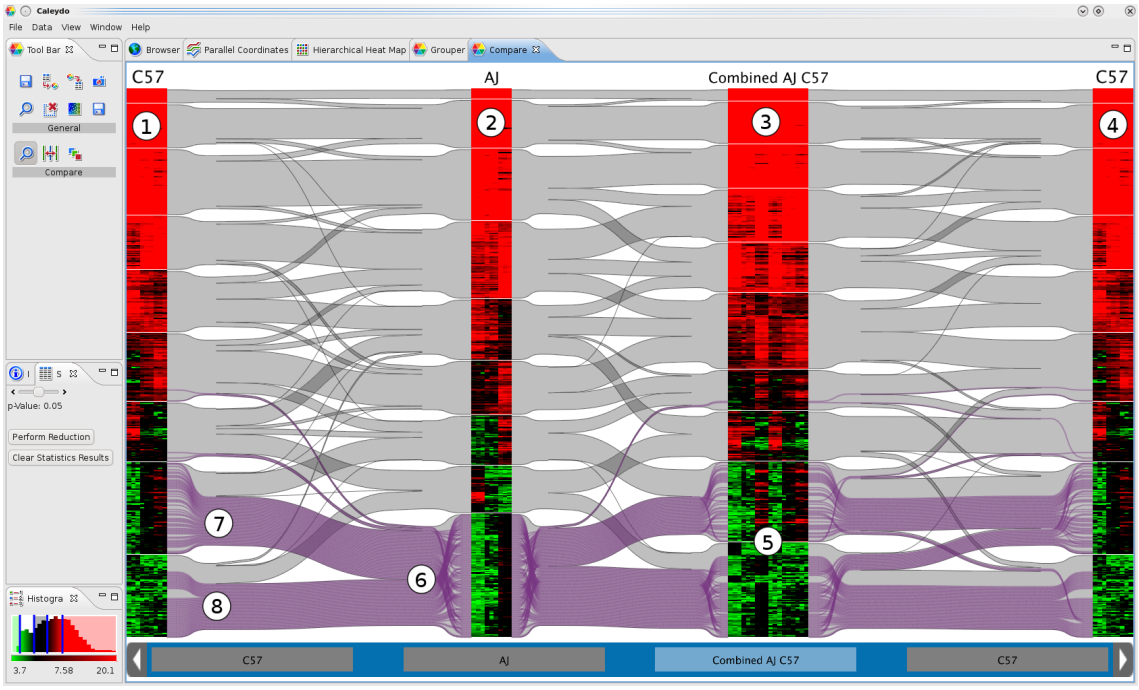


Figure 5.10: Screenshot of the Caleydo Matchmaker in overview mode taken during an analysis session by a biologist. We see four columns (1-4). The first two and the last, C57 and AJ, are homogeneous with respect to semantics, as the samples they group belong to the same genotype. Each column consists of 9 experiments: reference, 7 days of intoxication and 8 weeks of intoxication from left to right, with 3 replicates per category. The third column, showing a combination of C57 and AJ, contains all experiments from the first two groups. The fourth group (4) is a copy of the first to enable better comparisons between C57 and the combined column. The combined column contains inhomogeneous clusters (5). Clustering the homogeneous columns yields more consistent results, allowing a biologist to assign meaning to a cluster. The biologist brushed the bottom brick in AJ (6), identifying that the genes in this cluster are split into two clusters in C57, one being similarly regulated over time to AJ (7), the other (8) containing genes not-deregulated (equally regulated) in C57, while up-regulated (going up over time) in AJ (6). Since this difference may be important, he chose to explore this cluster in detail.

features, whereas C57 mice do not. To determine which genes are differentially deregulated in the two mouse strains, they performed an experiment where three groups of animals in each strain were fed with DDC for 8 weeks, 7 days or not at all (reference). Gene expression data was obtained from the liver tissue of these animals using whole-genome microarrays with 33,000 probes by Applied Biosystems Inc*. The analysis involves finding genes deregulated (i.e., those changing expression over time) due to DDC feeding in AJ animals, the responder strain, but not deregulated in the C57 animals, and vice versa. This analysis is difficult to perform with traditional tools, which do not treat the groups

*http://www.appliedbiosystems.com/

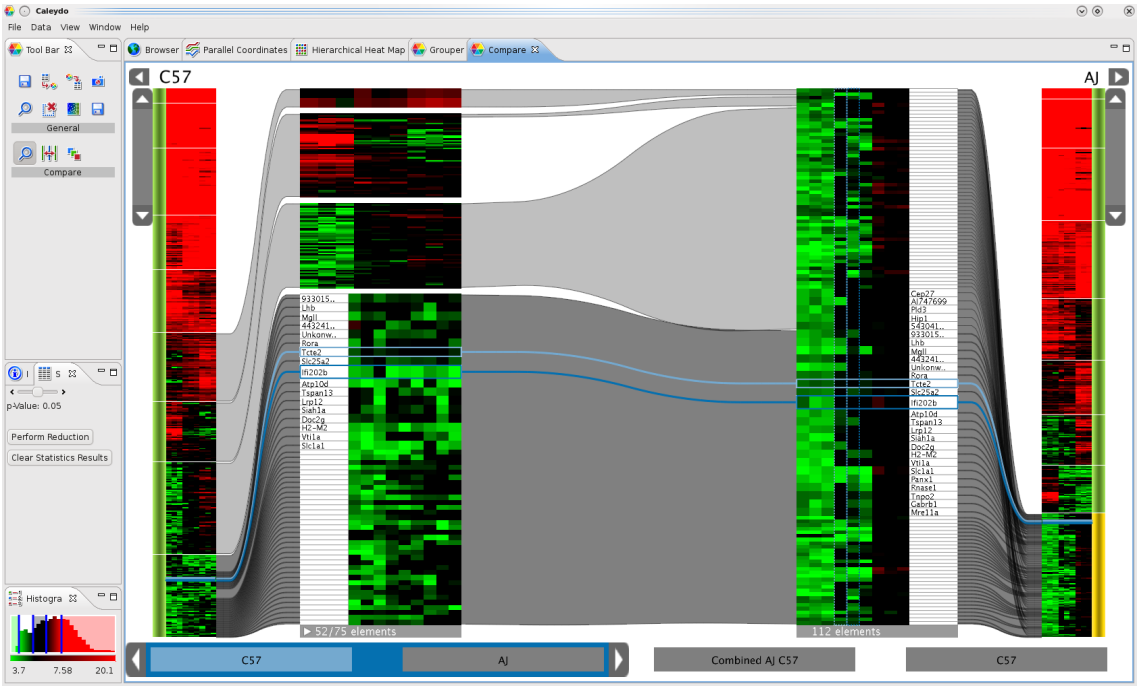


Figure 5.11: Screenshot of the detail mode during the analysis. We see the deregulated cluster in AJ on the right, and the not-deregulated cluster for C57 containing the same genes on the left. By exploring the genes and using Caleydo's built-in features to find contextual information on genes, he was able to hypothesize that these genes are involved in apoptosis and thus alter the phenotype of the liver tissue by removal of cells damaged by oxidative stress.

individually.

Using the Caleydo framework and the Matchmaker visualization technique, they were able to perform cluster analyses on the DDC feeding experiment in each mouse strain separately. Figures 5.10 and 5.11 show two screenshots taken during an analysis session by a biologist. Figure 5.10 shows the regulation over time for the different mouse strains in the columns, with reference (no intoxication), 7 days and 8 weeks, with 3 experiment replicates each from left to right. At (1) on the left is the C57 strain, next to it the AJ strain (2), then a group where both are combined (3) and a duplicate of the C57 strain (4). In the overview, we see that the bottom two clusters of the combined column are very inhomogeneous (5). When following the highlights, it becomes obvious that when the clustering is done on a single strain, the genes present in the highlighted cluster in AJ are being split up into two clusters in C57 at (7) and (8) in Figure 5.10. One of those clusters in C57 contains genes not-deregulated (equal over time) in C57 (8). The expert noted that these genes might be important in the different reaction of C57 to DDC intoxication. He then continued to analyze this cluster in more detail, as shown in Figure 5.11. While browsing the list of genes in this cluster, he found several genes involved in the regulation of apoptosis (programmed cell death), which might cause cellular turnover in the liver and alter the phenotype by removing cells damaged by oxidative stress. The removal of these damaged cells, which are prone to ballooning and have Mallory-Denk bodies by apoptosis,

could be a reason why these features of steatohepatitis are absent in C57.

The expert stated that for him the key advantage of clustering distinct groups (AJ and C57) separately is that he can quickly assign a biological meaning to a cluster (for example "up-regulated in AJ"). Matchmaker then enabled him to follow these genes in the other strain and see how they behave there. This is more difficult when the groups are clustered together, as the clustering algorithm tries to find a best match over both groups and thus makes the clusters inhomogeneous.

5.5.2 Comparison of Clustering Algorithms

Usually, data analysis tools provide a wide range of clustering possibilities to the user. There are several types of clustering algorithms, for example, partitional versus hierarchical, divisive versus agglomerative, unsupervised versus supervised; and other influential

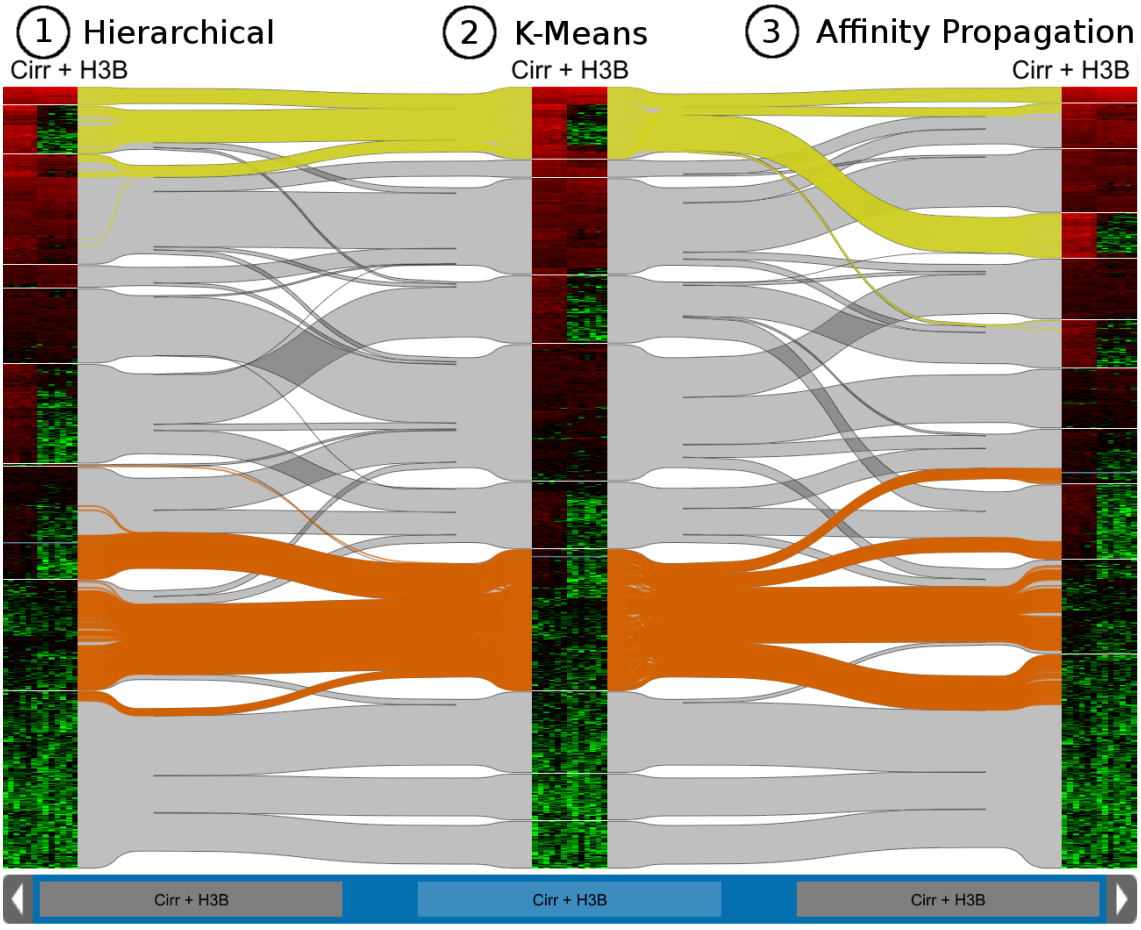


Figure 5.12: A comparison of three clustering algorithms run with 1800 records: (1) hierarchical clustering, (2) k-means and (3) affinity propagation. The yellow and orange brushes show that the k-means algorithm assigns obviously different records to one cluster, while the other two algorithms work as desired.

factors such as the choice of a distance measure or parameters. However, users are often not aware of the consequences of these factors, and cannot anticipate the results. Due to the flexible arrangement of dimension groups in Matchmaker, the user can load the same data (sub)sets multiple times, showing each as a column. The dimension groups can be clustered separately with either the same algorithm and varying parameters, or completely different algorithms. Matchmaker thereby enables a user to understand the impact of the cluster algorithms and its parameters applied to a concrete dataset. Consequently, the user can decide which clustering algorithm fits the data best. Figure 5.12 shows the clustering algorithm comparison scenario using gene expression data. Experiments, i.e., dimensions, of two cell lines are grouped together and clustered multiple times using different algorithms: hierarchical clustering on the left, k-means clustering in the middle and affinity propagation on the right. All algorithms are parameterized, so that they produce a similar number of clusters. The same distance measure – the Euclidean distance – was used in all cases. The brushed bricks in Figure 5.12 clearly show that the k-means algorithm assigned differently expressed genes to the same cluster, while affinity propagation and the hierarchical clustering algorithm created separate, homogeneous clusters. Also, at the bottom of the columns, k-means splits the group of genes, which both, the tree clustering algorithm, and affinity propagation assigned to one cluster, into three separate clusters, with no clear evidence of difference between the records. This leads to the conclusion that the k-means algorithm is not a good choice for this data, while the two other algorithms achieve comprehensible – but still different – results.

5.5.3 Discussion

When observing our users during the case studies, we noticed that the process of data preparation (choosing and generating groups, running clustering algorithms on the groups) needs to be improved. While this was not the focus of our research, it is crucial for an adoption by end-users that this process is made intuitive.

For the Matchmaker interface itself, feedback on ease of use was positive throughout. Nevertheless, we noticed significant differences of how easily users understand the benefits of the methods for the two use cases. When comparing clustering algorithms, the meaning of the groups and their relationships are immediately obvious – one group corresponds to one clustering algorithm and all groups show the same data. However, for biomolecular analysis, where meaningful sub-spaces of the data need to be created in order to benefit from the Matchmaker technique, a more thorough introduction was necessary. We believe that this is due to the unconventional arrangement of the heatmaps. However, after our collaborators were instructed that clusters are now largely homogeneous, allowing them to easily identify how clusters change between groups, they greatly appreciated the benefits for their applications.

5.6 Conclusion and Future Work

In this chapter, we have presented Matchmaker, a visualization technique addressing the division hypothesis (Hypothesis I). We have shown how the data can be stratified into homogeneous groups on the one hand, and how the lost relationships can be re-introduced

on the other hand. The separation of dimensions into homogeneous subgroups makes a meaningful automatic partitioning more likely and avoids obscure, scrambled clusters. The explicit representation of the relationships among records and record groups of different dimension groups by using visual links demonstrates that the breakup due to the stratification can be remedied without loss of information or clarity. Consequently, Matchmaker allows users to find patterns in the data, which otherwise would be obscured. We have also shown that Matchmaker can be used to compare the effects of different clustering algorithms.

We have validated our claims through two case studies. The first shows that Matchmaker is a valuable tool for biomolecular data analysis. The overview allows users to easily identify possibly interesting patterns, which can be explored in detail using the drill-down techniques presented. The case study on cluster algorithm comparison demonstrates how the technique can be used to evaluate the quality and properties of clustering algorithms, their parameters or both. We believe that this can be very helpful in choosing the right clustering algorithm for a wide audience.

We therefore can conclude that Hypothesis I, the division hypothesis, is fully supported. Since the original publication [114] of the ideas discussed in this chapter, a number of related techniques have been published. An example are two articles by Turkay et al. [187, 188], who build on our ideas for cluster comparison and extend them to consider cluster quality and structural changes of temporal clusters. Another example is a paper by Dinkla et al. [33], which uses a similar approach to cluster comparison, but extends it by including hierarchies. A recent example of a hybrid continuous/categorical data analysis scenario using a related visual metaphor is presented by Misue et al. [129]. These examples show that the Matchmaker technique was well-received in the visualization community.

Matchmaker is limited to show heatmaps inside the bricks. While this is reasonable for many use cases, a more general approach, enabling users to choose a visual representation, can have several benefits. We will discuss those extensions in the next chapter.

Chapter 6

Multiform Visualization of Stratified Subsets

Contents

In the previous chapter we addressed Hypothesis I by showing that stratifying a dataset into homogeneous groups is beneficial for an analysis, since the resulting subsets can be more easily interpreted. We also demonstrated that the loss of relationships, caused by the division of the data, can be remedied by employing visual links, so that the resulting visualization makes it easier to find patterns.

In this chapter we discuss how the Matchmaker concept can be extended: instead of showing only heatmaps at a predefined scale, we introduce multiform [148] bricks. **Multiform representations** can show the same dataset in one of multiple forms, employing different visualization techniques, levels of detail, or levels of abstraction. We will show that the ability to use multiple, alternative visualizations for the data subsets is an additional argument for employing divide and conquer visualization strategies. This allows the visualization techniques to be tailored to the degree of homogeneity of a subset, the task of a user, and to the size of a dataset, thereby addressing Hypothesis II, the multiform hypothesis.

We introduce the **VisBricks** visualization technique that aims to provide such multiform representations in a highly configurable framework that is able to incorporate any existing visualization technique as a building block. Together with a rich set of interactions and visual cues that help to merge, split, rearrange, and reconfigure the bricks, this flexible new representation supports many exploration and comparison tasks that otherwise would be difficult to accomplish. A visual impression of an implementation of the VisBricks approach is given in Figure 6.1. We take up the column-wise arrangement of

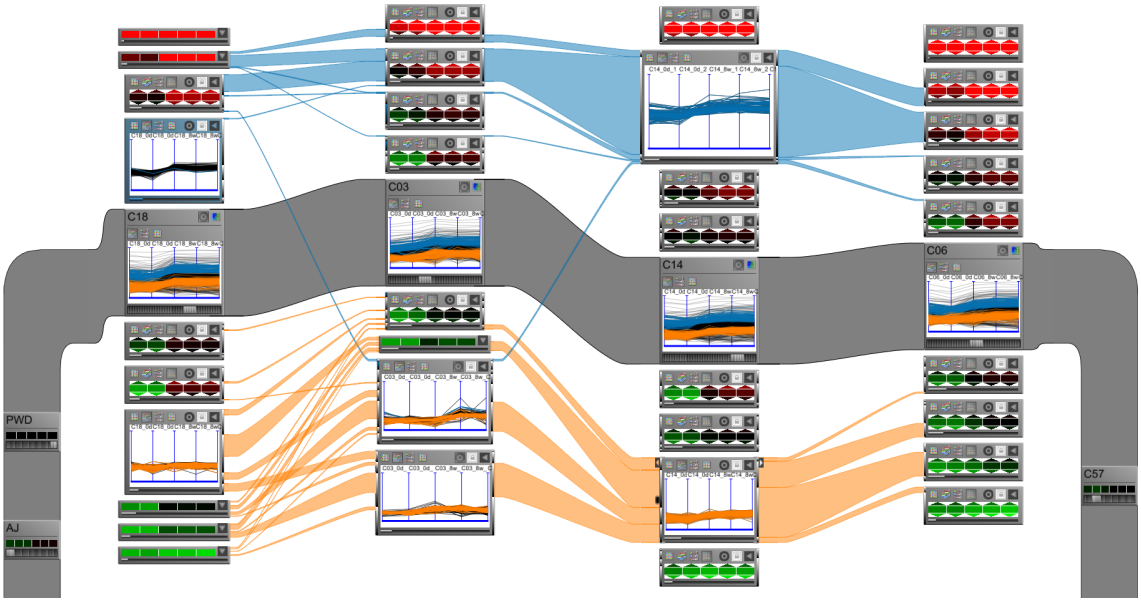


Figure 6.1: The VisBricks multiform visualization technique. Four dimension groups with different numbers of record groups each. The gray arch connects the overview bricks of the dimension groups. The record groups themselves are shown in stacked bricks above and below the arch depending on whether their average values are higher or lower than the overall average of the dimension group. Colored ribbons indicate how data items are distributed across record groups of multiple columns.

Matchmaker, but instead of heatmaps we provide several alternative visualization techniques, ranging from very abstract but space-efficient views to full-featured visualization techniques such as parallel coordinates. The arrangement of both, columns and bricks, as well as the sizes of the bricks can be adapted to the user's needs. We introduce several new classes of bricks, among them *header bricks*, which represent the data of a whole column. Header bricks are placed inside an *arch*, the bricks representing the record groups are placed above or below the header bricks, depending on a sorting strategy. We take up the molecular biology use case discussed in Chapter 5 and show how VisBricks can improve the analysis process. The results are promising and also indicate directions for future research.

6.1 The VisBricks Approach

For large datasets, it has proven efficient to follow Keim's *Visual Analytics mantra*: "Analyse First, Show the Important, Zoom, Filter and Analyse Further, Details on Demand" [99]. VisBricks embraces this paradigm and strives to support it on all levels by providing meaningful **preprocessing and overviews** to show the important features even for inhomogeneous data; a rich set of interactions to enable **zooming, filtering and further analysis**; and drill down methods to explore even large datasets down to the

details of the individual record. The core paradigm of VisBricks is to apply multiform visualization to stratified datasets: the homogeneous subsets can thereby be efficiently abstracted. VisBricks fully support the inhomogeneity of the data and the diversity of tasks at each level of the mantra through their multiform approach. Using multiform bricks permits users to tailor the visual representation of each subset of the data according to its characteristics, the task that is to be performed, and the level of detail required.

In this section we explain the conceptual foundations of the VisBricks technique, beginning with the overview, continuing with interaction aspects that enable zooming and filtering, and finally providing details about how the data is presented on a fundamental level.

6.1.1 Preprocessing and Overview

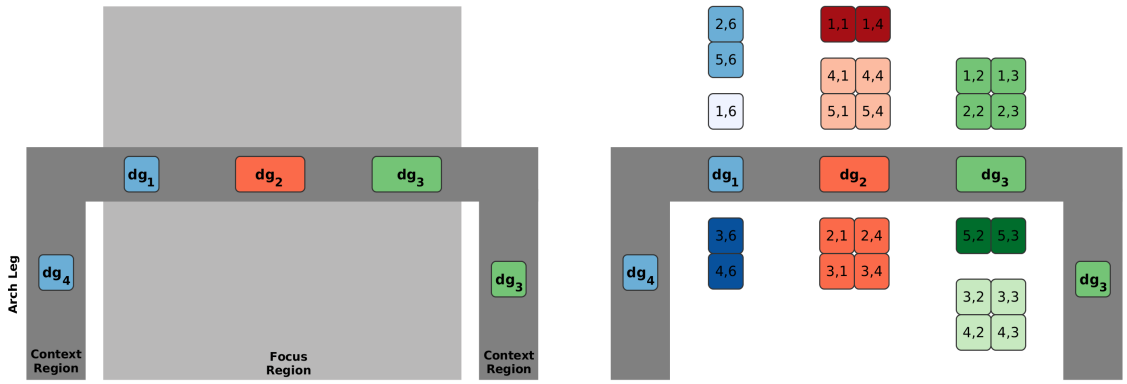
Abstraction is a key technique that enables an overview with limited visual or computational resources. There are several ways to achieve abstraction. Oliveira and Levkowitz [44] list dimension reduction, sub-setting (e.g., random sampling [34]), aggregation [41], and segmentation (e.g., cluster analysis [38]). While the former two provide abstraction by themselves, the latter only enables more meaningful sampling or aggregation.

An inherent property of homogeneous data is its suitability for abstraction. With homogeneous data, it is easy to choose a visual encoding that represents the data well. Inhomogeneous data, however, does not lend itself to reasonable abstractions. It is difficult or even impossible to find representative encodings for a very inhomogeneous dataset. Consider the following example: for a perfectly homogeneous multidimensional dataset, where every data item has the value 1, a single bar with height 1 is a suitable abstraction. A dataset of the same size but with very inhomogeneous values can not be abstracted as efficiently: if one would use a single bar all the diversity in the dataset would be lost.

VisBricks use the same basic process as Matchmaker. Bricks represent homogeneous subsets of the data, generated by vertical and horizontal stratification, that are aligned vertically and horizontally in dedicated drawing areas. The bricks are placed in the context of the whole dataset by using position and visual links.

We distinguish between two types of bricks: bricks representing and abstracting a whole dimension group, which we call *header bricks*, and bricks reflecting the subdivision of records within the dimension group, which are called *cluster bricks*, as the subdivision is often achieved using automatic clustering algorithms. The most important property of a brick is that it can encode its data in any number of ways and that it lets the user choose the technique while providing sensible defaults.

Populating the Arch with Bricks Header bricks are dynamically added to the arch in VisBricks. Figure 6.2(a) shows an illustration in which several dimension groups were created and can now be found in the arch. The dimension groups placed in the arch correspond to the example given in Figure 1.1 in Chapter 1. The arch has three regions: the center, where dimension groups that are currently in the focus of the investigation are placed, and two legs, one on each side, where dimension groups are moved when they are not in focus.



(a) (b)

Figure 6.2: Basic VisBricks concept. (a) The arch containing the header bricks. The layout is separated into a focus region, including the horizontal portion of the arch and the space above and below it, and the context regions in the legs of the arch. Dimension groups in the columns are intended to be homogeneous with respect to a user-defined homogeneity measure, i.e., homogeneous with respect to their semantics, characteristics, or statistics. (b) Cluster bricks were added above and below the arch for the dimension groups.

The dimension groups can then be further stratified into record groups, adding cluster bricks, as illustrated in Figure 6.2(b). Notice that this step is optional, as some dimension groups may not require further subdivision or may not be suitable for it. For other dimension groups, however, only this additional division makes it possible to create meaningful abstractions for the major trends in the dataset. As can be seen in Figure 6.2(b), cluster bricks are shown above or below their respective header bricks, but only for dimension groups in the focus region.

The achievable homogeneity of the cluster bricks depends on many factors. First, a sensible trade-off has to be found between the number of clusters and their degree of homogeneity. While our VisBricks implementation takes several measures to avoid clipping data, the number of clusters and therefore the number of cluster bricks has the greatest impact on the VisBricks' scalability. Second, the achievable degree of homogeneity for a given number of clusters depends on many factors, such as the choice of clustering algorithm, its parametrization, and the suitability of the dataset. After this division is accomplished, we are able to choose suitable visualizations for each brick to encode and abstract the now homogeneous subset of data.

Encoding Relationships between Cluster Bricks When exploring tabular data in a spreadsheet, sorting is a common strategy to find related records. Generally speaking, all visualization techniques that use rows or columns to identify records can make use of sorting. Techniques that encode relationships in a record differently, e.g., parallel coordinates, cannot employ sorting for that purpose.

When sorting by a single row in tabular arrangements, the other values in a record are

re-positioned accordingly. Sorting of multiple rows at the same time, however, breaks the ties between the values in the records. Sorting by more than one dimension simultaneously is equally desirable but much harder to achieve, as meaningful comparisons between tuples of values are more difficult to obtain. Consequently, few techniques are able to achieve such sorting. One notable exception is the table-based visualization for bipartite graphs [158], in which the disjoint sets of the graph are visualized in tables and sorting can be performed for each of the sets independently and also simultaneously. Because of the nature of the underlying data (a bipartite graph), no special care has to be taken to keep the association between the records intact.

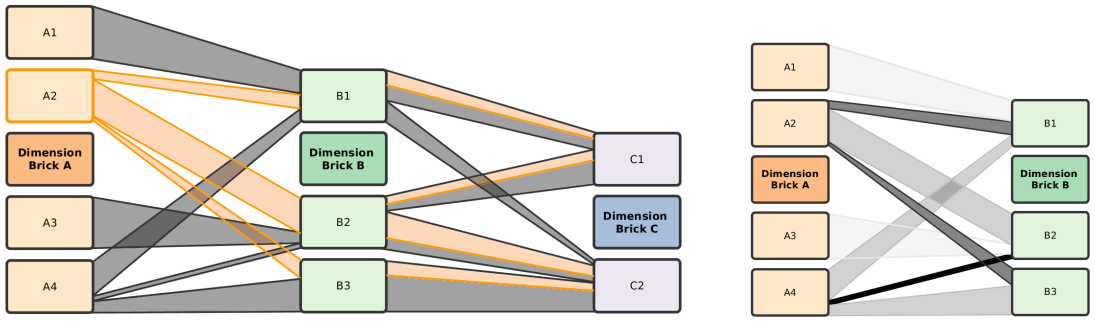
The Matchmaker technique employs sorting based on averages of clusters. VisBricks adopts this general idea and enhances it by additionally encoding the relationship of every brick to the average of the whole dimension group. The vertical position of a cluster brick is determined by two factors: the ranking according to the sorting strategy used and the relative value compared with the average of the whole dimension group. Because of the placement relative to the whole dimension group's average, the header brick and the arch divide the cluster bricks into those above the average and those below it. Thus, it is clear how each cluster brick compares to the other cluster bricks within a dimension group, as well as to the overall average.

Sorting strategies for numerical data would, for example, place the cluster brick with the highest average at the top and the brick with the lowest average at the bottom, whereas categorical data could be sorted by frequency. If no meaningful sorting strategy can be defined for a certain type of data, the bricks could be sorted to minimize crossings and distributed evenly above and below the header brick.

By partitioning and sorting the data records separately in the different dimension groups, the association between individual values of a record across dimension groups is no longer obvious, as the strict horizontal and vertical alignment of the data matrix has been broken up. Hence, the following conquer steps re-introduces this essential information in the overall layout of the bricks by encoding the relationships through visual links.

Encoding Relationships between Dimension Groups To provide a meaningful overview, the relationships between the dimension groups must be made explicit, thus realizing the conquer step. We achieve this by using both traditional, color-based **linking and brushing** as well as **interactive visual links**.

VisBricks employs ribbons for conveying which portion of the data contained in each brick is shared among bricks in neighboring dimension groups. When the bricks are brushed, the ribbons are not only shown for the relationships to the neighboring dimension groups, but also split into multiple threads connecting all related bricks in all dimension groups (see Figure 6.3(a)). In contrast to Matchmaker, VisBricks does not connect individual records, but shows proportional relations among the bricks. This is due to the abstract nature of some visualization techniques that can be employed in bricks, where records are not necessarily associated with a position along the height of the brick and therefore cannot be connected. The width of the ribbons encodes the magnitude of the relationship. This strategy is similar to the one used in *Parallel Sets* [104]. In contrast to Parallel Sets, the ribbons are not color-coded by default because the number of clusters can easily exceed the number of distinguishable colors, which has been shown to be fairly



(a) Selection propagation

(b) Interactive trends filter

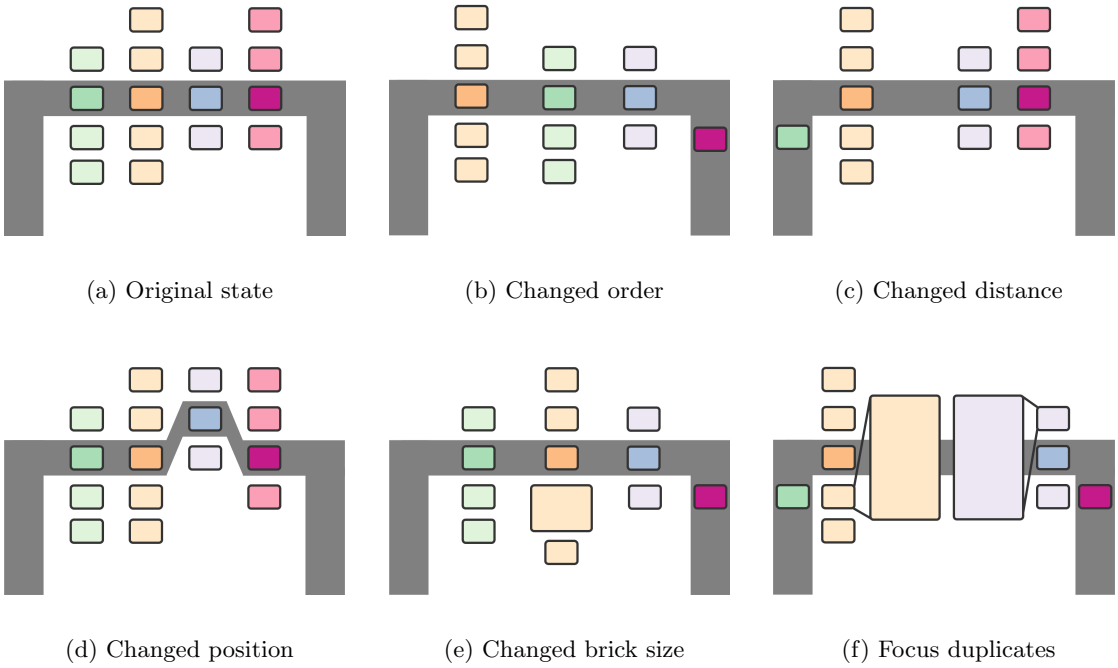
Figure 6.3: Ribbons connect bricks between adjacent dimension groups, thus indicating how many elements are shared among them. In (a) brick A2 is selected. The selection is propagated to all connected bricks. (b) The result of an interactive filter that focuses on outliers. The wider the connection band is, the lighter it is drawn.

limited [66]. Also, color might already encode other attributes within the brick's visualizations and is also employed for brushing the ribbons. Thus, VisBricks initially shows all ribbons in semi-transparent gray. It is possible to brush whole bricks, or only the ribbon connecting two bricks, thereby focusing on the subset of data shared by the two bricks connected. The brushing is propagated to all bricks where the elements are highlighted accordingly. In addition, the corresponding portion of the ribbons is colored, as can be seen in Figure 6.3(a). The brushing of bricks or ribbons can also be reflected in the views contained in the bricks. VisBricks support multiple simultaneous brushes, assigning a different color to each brush. In cases with many clusters, it is sensible to show ribbons only for brushed bricks, therefore not-brushed ribbons can be turned off.

Whereas wide ribbons show major trends among the dimension groups, thin ribbons indicate outliers. Initially, the showing of both outliers and major trends is a good option to convey an overview. However, in many tasks either only outliers or only major trends are relevant. We therefore propose a technique that allows users to interactively specify whether they are currently interested in the main trends, outliers, or anything in between. Because "outlier" or "main trend" are no absolute concepts, we chose to decrease the opacity for bands further from the current focus. Figure 6.3(b) shows an example in which the focus lies on outliers.

6.1.2 Zoom, Filter and Analyze Further

The provision of overviews is essential in making it possible to understand a dataset. However, to extract knowledge, it is necessary to drill down, either via interactive zooming and filtering or via a re-parameterization of the analysis, e.g., by refining the clusters. While the latter is not a matter of the visualization itself, the interactive zooming and filtering are performed directly on the visualization and should thus be supported by it. VisBricks provides five interaction patterns for manipulating the bricks and their layout.



(a) Original state (b) Changed order (c) Changed distance

(d) Changed position (e) Changed brick size (f) Focus duplicates

Figure 6.4: Interaction patterns in VisBricks. (a) shows the original state, whereas (b)-(f) show the consequences of the five different interaction patterns.

1. **Changing the order of columns**
   Coherent with Matchmaker, columns can be moved in and out of the focus region; the latter provides more space for those in focus. Additionally, the horizontal order can be modified, allowing a more detailed side-by-side comparison of different columns. When dimension groups are brought into focus manually, others can be forced out of focus and into the context region if more space is required than available. Figure 6.4(b) illustrates an example in which the order of columns in the focus region was changed and one dimension group was moved to the right leg.

2. **Changing the distance between columns**
   It can be desirable to change the spacing between columns. Increased space will be useful if the relationships between two neighboring columns are under investigation. In this case, the increased space reduces the clutter produced by the ribbons. A reduction of space is typically achieved automatically when the space is increased elsewhere. In Figure 6.4(c), the orange column was moved to the left, which pushed the green dimension group out of focus into the leg.

3. **Changing the vertical position of columns**
   By changing the vertical position of the columns, cluster bricks, which are close to or even beyond the border of the screen, can be moved into the center, and comparisons between two bricks of neighboring columns are facilitated. As shown in Figure 6.4(d), the arch is bent, if necessary, to guarantee that it always encloses the header brick.

4. **Changing the size of a brick**
   Each brick can be resized so that the contained visualization is alloted more space, as shown in Figure 6.4(e). When the space for a brick is increased, other bricks are moved upwards or downwards, and other columns are moved to the side. Again, dimension groups are moved to the legs, if necessary.

5. **Creating a focus duplicate of a brick**
   When a full-sized visualization is more suitable for a given task, VisBricks provide the means to allow a brick to temporarily claim additional space for an enlarged focus mode. However, this focus mode is not simply an enlarged version of a brick, which would be achievable using only the resize functionality. Instead, the focus mode provides means (a) to compare single bricks in detail to another column, (b) to compare this brick in detail to a brick of another columns, and (c) to prevent the other bricks of the same columns from being clipped. In contrast to Matchmaker, the focus mode is chosen for a single brick of interest, which is then duplicated and placed next to its dimension group. By choosing the side of the columns on which the brick is to appear, the target of the comparison is implied. When the detail brick is visible, its connections to the neighboring columns appear. A user can now analyze the relationships and choose a brick from the compared columns for detailed analysis. Figure 6.4(f) illustrates the state in which a second brick is enlarged. For some visualization techniques, the available horizontal space may not be sufficient. In such cases, the legs of the brick are moved out of the view, to increase the space for the focus bricks. Having only two focus bricks instead of multiple focus bricks as in Matchmaker, guarantees that the focus bricks are enlarged sufficiently to allow interaction with arbitrary visualization techniques.

Considering these interaction techniques it becomes apparent that a drill-down from the overview, which only shows the important data in abstracted views, to detailed views of individual homogeneous subsets is fully supported by VisBricks. Additional considerations regarding the detailed visual analysis of individual data properties are discussed in the following section.

6.1.3 Exploring Details

The detailed analysis in VisBricks is based on the multiform property of the bricks. Although we previously mentioned that multiple visualization techniques can be used within a brick, up to this point we have mainly treated bricks as a medium to present abstractions. However, bricks are more powerful.

The defining property of bricks is their ability to display the information grouped within them using diverse visualization techniques. We have distinguished between header bricks, which summarize the entire data in a dimension group, and cluster bricks, which show data that is homogeneous in terms of statistics. Both require very different visualizations, as the header bricks give an overview of the grouped dimensions, whereas the cluster bricks show the records grouped inside them. In general, it is not immediately obvious which visualization is sensible for which brick. The suitability of a technique depends on two criteria:

1. **Data characteristics criterion:** Is a technique suitable to visualize the data for the given data characteristics?

2. **Scalability criterion:** Is a technique suitable to visualize the given amount of data in the allocated space?

Data Characteristics Criterion For bricks that are homogeneous with respect to their data characteristics (see Section 1.2 for the definition of data characteristics), it is easy to assign suitable visualizations. The availability of a concrete visualization technique as a representation choice for such a brick requires only the knowledge that the technique can visualize data of the desired characteristics. An example is a parallel coordinates view, which is suitable for bounded numerical, unbounded numerical, and, to some extent, exclusive categorical but not for inclusive categorical.

However, when dimension groups are not homogeneous with respect to their characteristics, but only with respect to their semantics, it is not as simple to assign suitable visualizations. In this case, we seek out the "least common representation" that is sufficiently generic to be able to show all of the data types within such a mixed dimension group. To achieve this, we order the data types according to their *strictness* for the data characteristics. For the four data types, we consider bound numerical to be the strictest characteristic, followed by unbound numerical, exclusive categorical, and, finally, inclusive categorical as the most *relaxed* type. This ordering is based on the observation that data belonging to a stricter class can often also be visualized with a technique suitable for a more relaxed data type. What distinguishes visualization techniques for stricter classes from those for more relaxed classes are the assumptions about certain properties of the data that do not hold for more relaxed types. An example is a technique for bounded numerical values that assigns each record a hue of 1 for the upper bound and 0 for the lower bound. If this technique is used with a hybrid dimension group, in which one dimension contains unbound values, their color coding will become meaningless.

Visualization techniques for more relaxed data types have to allow their records to take on a wider variety of states, making the individual record more expressive, but also harder to abstract. This does not mean that a technique for a more relaxed characteristic is not suitable for a stricter characteristic; rather, it means that such a judgment cannot be derived automatically.

Note that it is not reasonable to employ a technique that is suitable for more relaxed characteristics to all stricter ones. Usually, more relaxed techniques are not able to fulfill the scalability criterion as well as stricter techniques do.

Scalability Criterion VisBricks heavily relies on the abstraction technique of segmentation into homogeneous groups at the top level, and in fact we employ a multi-level approach: bricks are required to provide at least one abstraction method for every data characteristic. Hence, each visualization technique can make use of the provided abstraction methods as needed. Dix and Ellis note that multi-level abstraction solutions are common; for example, a sampled dataset can be used as the input for aggregation techniques [34].

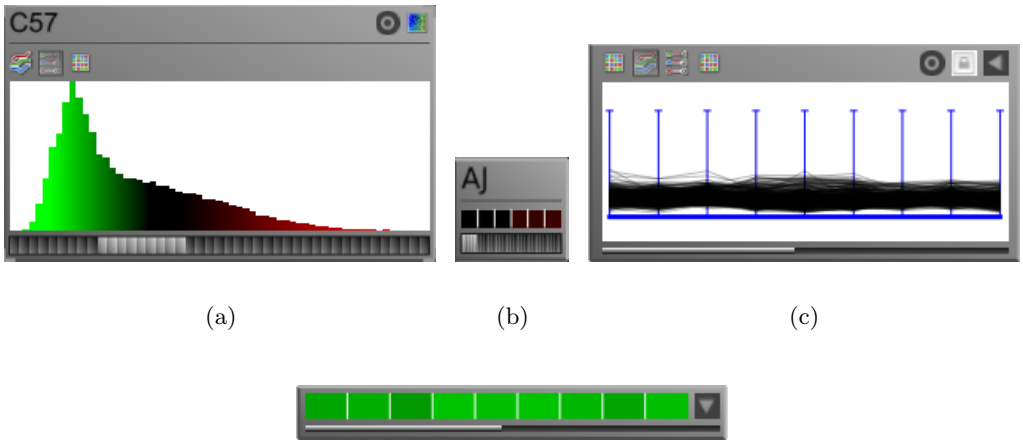


Figure 6.5: Classes of bricks used in VisBricks. (a) Regular header brick, summarizing a dimension group. (b) Compact header brick, used in the arch legs. (c) Regular cluster brick, showing one homogeneous cluster. (d) Compact cluster brick, used for overviews. The bricks in (a) and (b), respectively in (c) and (d) show the same data.

We distinguish between four classes of bricks, where each has different requirements considering the scalability criterion:

1. **Regular header bricks**
   Header bricks represent all records in a dimension group. As the number of records can be large, techniques that rely on the scaling of width or height with the number of records are not suitable for header bricks. Consequently, aggregation methods, such as histograms, or methods using sub-setting and natural aggregation, such as parallel coordinates, are suitable. In contrast, methods that require additional space for every record, e.g., clustered heatmaps [38, 204] or tables, are not suitable. A Regular header brick is shown in Figure 6.5(a).

2. **Compact header bricks**
   When a dimension group is moved to the legs of the arch, its cluster bricks are hidden, and the header brick is reduced to a static size, optionally showing a high-level aggregation of the data. Figure 6.5(b) shows an example for numerical data, in which the whole dimension group is aggregated into one single line of a heatmap. Although this abstraction is very crude, it may show a major trend in the data.

3. **Regular cluster bricks**
   Regular cluster bricks have the most freedom of all bricks. They may use any visualization technique suitable for the data, including those that require the scaling of width and height with the number of records. For example, Figure 6.5(c) shows a cluster brick containing a parallel coordinates view. Basically any imaginable visual-

ization technique that is able to provide an overview of a number of multidimensional records can be used inside a regular cluster brick.

4. **Compact cluster bricks**
   For each data characteristic, VisBricks requires one technique that represents a cluster at minimal height. This technique is used by default when otherwise the bricks would not fit in the view. Although this technique cannot completely avoid clipping, it significantly increases scalability. The actual height is not specified, because, for example, efficient visual abstractions of bricks that are inhomogeneous with respect to their characteristics are much more difficult to achieve than those for numerical data. Compact cluster bricks have a reduced set of user interface elements, which help to keep the size minimal. Figure 6.5(d) shows an example for numerical data, in which a heatmap line, similar to the abstraction used in the compact header brick, shows an aggregation of the cluster. Under the assumption that the records in the brick are in fact homogeneous, this abstraction is a valid representation for the cluster.

In addition to these four fundamental modes, views are also notified of the actual size of a brick. This makes it possible to prevent users from switching to visualization techniques that require more space than the brick currently has available. Also, the level of detail of visualizations can be adapted. The parallel coordinates, for example, add captions when a certain size threshold is surpassed and user interface elements when the view is enlarged further. This is especially relevant for focus duplicates of bricks.

With these scalable bricks at hand, users can interact with the data, drill down into record groups, explore the details of relationships between record groups and dimension groups, and even see the actual values of every single record in the data. In Section 6.4, we will present the results achievable with a prototype implementation. However, first we will discuss some design choices and scalability issues.

6.2 Design Choices

In addition to the main paradigms discussed up to this point, there are some additional considerations to improve the usability of bricks.

One piece of information that is lost when abstracting homogeneous groups of dimensions and records is the scale of the group. A homogeneous brick containing only a few elements is, for example, assigned the same space as another brick containing half the dataset. It is therefore necessary to encode the relative size of the groups in terms of the number of dimensions for the dimension group and the number of records for the cluster bricks. To encode the number of dimensions we use a row of squares with one square for each dimension; the squares will be filled if this dimension is part of the dimension group, as shown in Figures 6.5(a) and (b). We encode the number of records in the cluster bricks with a bar, as shown in Figures 6.5(c) and (d).

Also, the bricks need to contain user interface elements to, for example, display the name of a dimension group or allow switching between visualization techniques. Many

approaches are conceivable. For our prototype, we chose a mixture between static and pop-up buttons, which can be seen in Figure 6.5.

6.3 Scalability

VisBricks scales to a large number of records and dimensions. The primary limiting factor for the number of records is the computational limitation of the clustering algorithms. A secondary limitation is the available resolution: On a screen with 1680×1050 pixels, VisBricks can handle up to 30 clusters in one dimension group, thereby surpassing the Matchmaker technique. The cluttering of connections associated with a high number of clusters among many dimension groups can be improved by rendering ribbons only when brushed, or by using the trend filter. VisBricks can accommodate about ten to fifteen dimension groups, up to eight of which may be in the focus region.

6.4 Case Study

We evaluate VisBricks with data from the same analysis scenario as described in Chapter 5. Our partners from the Medical University of Graz want to find the genetic factors involved in steatohepatitis. To be able to monitor the expression of genes, they developed a mouse model, where genotypes of mice respond differently to intoxication with DDC. The use case discussed in Chapter 5 included the two major mouse strains, The AJ and C57, of which AJ develops a phenotype with steatohepatitic features while C57 does not. However, our partners also record measurements from PWD, another non-responder strain, and mice genotypes where one chromosome of the C57 genotype was substituted with the homologous chromosome from AJ, resulting in consomic mice. This can help to isolate the chromosomes which have an effect on the phenotype. In total, they record gene expression data for 7 different genotypes of mice, with data being collected without intoxication (reference), after seven days, and after eight weeks, with three biological replicates for

	AJ	**C57**	**PWD**	**C03**	**C06**	**C18**	**C14**
Responder Strain	Y	N	N	?	?	?	?
Consomic Mouse	N	N	N	Y	Y	Y	Y
Collected Data							
Reference	Y	Y	Y	Y	Y	Y	Y
7-Day	Y	Y	N	N	N	N	N
8-Week	Y	Y	Y	Y	Y	Y	Y

Table 6.1: Experimental setup of the steatohepatitis mouse model. For each of the mouse strains in the columns, our partners collected data without intoxication (reference) and after eight weeks of intoxication. 7-day intoxication data was only collected for the most important genotypes. Typically, each condition was conducted with three biological replicated each, with the exception of the consomic mice, where only two replicates were used.

Figure 6.6: The VisBricks overview containing seven columns stratified by mouse genotypes. Two of the columns are clustered, and their cluster bricks are shown. Subtle differences in the histograms in the dimension bricks are evident. The PWD genotype is placed in the right arch leg to make space for the other columns.

each condition. See Table 6.1 for details on the experimental setup. But not only the scope of the data presented here goes well beyond the previous use case: the VisBricks approach supports the full range of visual analysis, from a comprehensive overview of the topology of the entire dataset that integrates diverse computational and visual options, seamlessly down to the individual data record.

Following the visual analytics mantra, the computational analysis constitutes the first step. In this dataset, there are multiple levels of semantic inhomogeneities, i.e., measurements taken at the different points in time or from the different genotypes of mice. Sensible groupings of the data depend on the research question. Thus, if changes over time comprise the main focus, grouping based on similar points in time would be the best choice. However, because the differences in genotype are central to the research question, grouping based on genotypes makes most sense. To remove noisy and uncertain data the analyst filters the data using statistical methods and also removes values that are constant within a threshold across all conditions. The filtered dataset shown in Figures 6.6-6.9 has 37 dimensions, each containing the measurements of 1 sample, grouped by the 7 different genotypes, with 766 expression values per dimension.

The analyst is interested in differences between the AJ genotype and the consomic genotypes (C03, C06, C18, C14, in summary referred to as C*), as well as the non-responder strains (C57 and PWD). An example of a relevant observation is a gene that remains at the same regulatory level in the AJ mice but is upregulated as time progresses

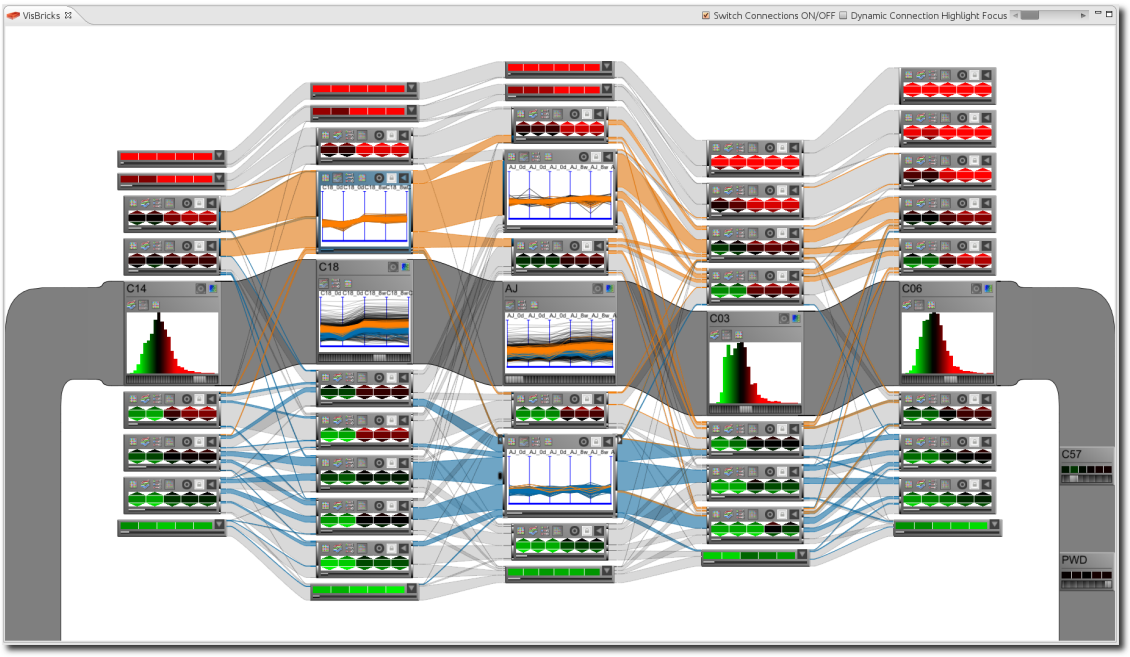


Figure 6.7: The seven mouse genotypes where all dimension groups are clustered and two cluster bricks are brushed. Notice the connection from the top-left brick, showing the parallel coordinates brushed in orange, to the lower brick in the center, where the parallel coordinates are brushed in blue. The brick in the center column contains outliers of the orange brick, which indicates genes of interest. While the existence of the outliers are easy to see because of the ribbons, the actual values of the outliers can be observed due to the colored brushing.

in the C* mice. Such a gene might be involved in preventing steatohepatitis in the non-responder mice.

Figure 6.6 shows the layout of the header bricks, one for each of the seven genotypes, as an overview of the dataset. Two dimension groups have already been clustered, and their corresponding cluster bricks are shown. The histograms in the header bricks show the summarized distribution of the values in the dimension groups, from low expression (at the left in green) to over-expression at the right in red. Subtle differences between the dimension groups are noticeable.

The analyst then proceeds by clustering the remaining dimension groups to uncover their statistical inhomogeneities. As the dimensions within the columns are sorted by time (early experiments are on the left, whereas the final measurements are on the right), there is a strong tendency of increased expression from left to right in the appearing cluster bricks in all columns.

The clustering groups together those genes with similar expression patterns. Such groups are often also functionally similar [38], making the clusters semantically meaningful. Looking for differences between a gene's expression in the AJ and the C* mice, the analyst is searching for two clusters that share elements (i.e., they are connected with a ribbon) but also show a different behavior for the genotypes. As the mice are treated exactly the

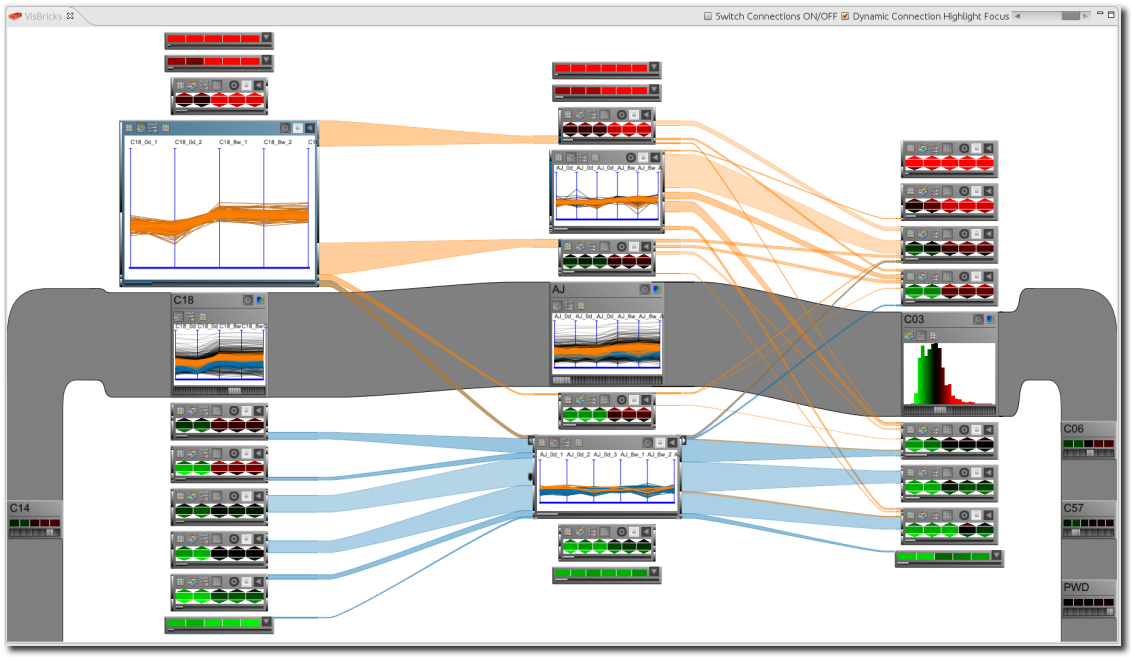


Figure 6.8: The bricks identified to have an interesting relationship are enlarged as part of a drill-down operation. The contextual information is reduced - some columns are put in the arch legs and only ribbons for selected bricks are shown.

same, such a difference is likely to stem from the difference in genotype and might thus be linked to the causes of steatohepatitis.

The analyst begins a more detailed analysis by filtering. He moves some columns to the arch legs to take a close look at the differences of the columns of interest. To see some of the more interesting bricks in detail, the analyst switches them to the parallel coordinates view. Other, less interesting cluster bricks, in which values remain nearly constant over time, are switched to the compact mode. The many broad ribbons between closely related cluster bricks show that much of the data is largely consistent across the dimension groups, indicating that those genes behave similarly in the different genotypes. However, there are connections between rather distant cluster bricks, hinting at possible outliers. Using interactive, colored brushing, the analyst explores the relationships of selected cluster bricks in more detail. The brushing highlights the ribbons and the actual data in the parallel coordinates. When brushing the cluster brick that shows the parallel coordinates in the second column (orange brushing in Figure 6.7), the analyst notices one brick in the neighboring column that is far away and very dissimilar. However, it still shares a few records with the brushed brick. The analyst switches the brick's aggregative view containing the outliers to a parallel coordinates view, where the outliers are immediately obvious. To explore the outliers in more detail, the analyst increases the size of the bricks and chooses to show only ribbons for outliers of brushed bricks, as can be seen in Figure 6.8. The shared records seem interesting and deserve closer investigation. Therefore the analyst creates a focus duplicate, as shown in Figure 6.9, where the genes are explored in

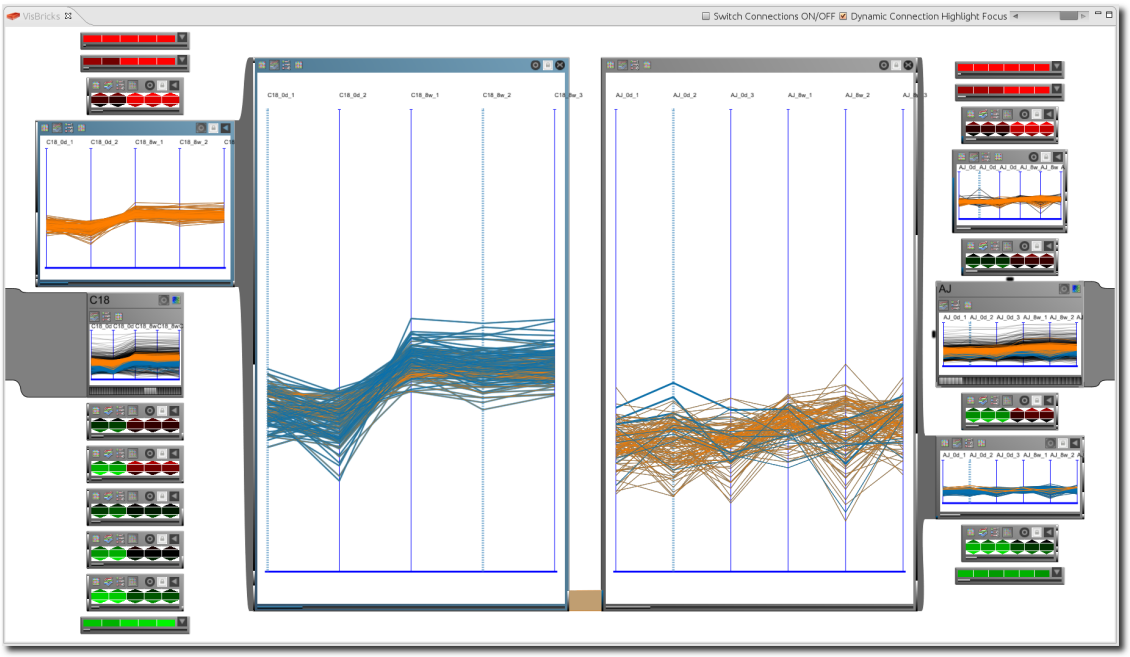


Figure 6.9: The bricks of interest with focus duplicates, enabling detailed analysis.

detail using two parallel coordinates views. The genes found may indeed play a role in steatohepatitis. The analyst continues the investigation by reviewing the literature on the found genes in online databases.

6.4.1 Discussion

The feedback from our partners was very positive; they were able to conduct an analysis only after a brief training period, in which the novel spatial arrangement and the meaning of the ribbons were explained. Our partners appreciated the interactivity of the system and its ability to focus on several different parts of the data at the same time. They noted that this was very hard to achieve in their previous workflow using earlier versions of Caleydo, other state-of-the-art microarray analysis tools, or statically generated R-plots. An interesting suggestion made was to integrate other, non-tabular data sources, such as pathways, into VisBricks as well.

6.5 Conclusion

We have shown that the VisBricks concept can handle large and inhomogeneous data spaces by employing it in a real-life, complex analysis scenario. The main advantage of VisBricks compared to traditional approaches is its ability to handle all types of inhomogeneities within data, both in the dimensions and in the records. This is achieved by treating each homogeneous sub-part of the data with the best available computational and visual tools. By using abstractions in the bricks, VisBricks is very scalable in terms of

the magnitude of records and dimensions. At the same time, the division into bricks and the rich set of interaction patterns allow users to employ multi-level approaches, where each brick contains an abstraction suitable to show the data at the desired level of detail. Consequently, Hypothesis II, the multiform hypothesis, if fully supported.

The VisBricks concept is sufficiently powerful to describe previous visualization approaches in terms of bricks, groups, and the relationships among them. One example for categorical data is Parallel Sets [104]. Each brick can represent a category, and Parallel Sets' "composed dimensions" can be interpreted as dimension groups. Parallel Sets optionally show histograms inside the categories, which is also possible in bricks. As VisBricks is a generalization of Matchmaker, the functionality of Matchmaker is covered in VisBricks.

At the very core of the VisBricks strategy are two concepts: to **show relationships of two disjoint units of data** and the establishment of **multiform visualization for those parts**. These concepts are in no way limited to individual datasets; they may also be applied to multiple, cross-referenced datasets. How this can be realized is the topic of the next chapter.

Chapter 7

Visualizing Relationships of Cross-Referenced Datasets

Contents

Up to this point, we have discussed strategies to visualize the relationships of multiple homogeneous subsets of single datasets. In this chapter, we extend the approaches introduced in the previous two chapters to accommodate multiple, cross-referenced datasets. We defined cross-referenced datasets as "datasets that share a type of identifier, or have types of identifiers that can be mapped to each other". A typical example for cross referenced datasets are those stored in relational databases, where a key defines the cross-references between the tables. Cross-referenced datasets are also very common in biology, where a common identifier, for example a gene name or a patient ID, encode structured relationships. Cross referenced datasets were discussed in detail in Section 1.2. While conceptually, each cross-referenced dataset can be considered as a dimension group and could consequently be shown in VisBricks, several issues are more likely to arise when working with multiple datasets:

1. **Dependent datasets** – Creating record-stratifications using clustering algorithms is desirable for most datasets. However, for some datasets it might be more interesting to explore how its data behaves based on the stratification of another dataset. A classical example is meta-data: What does the meta-data stored in a dataset reveal for the stratified groups of a primary dataset?

2. **Unequal scale** – It is possible, that the datasets are of different scale, or that only subsets of two datasets are cross-referenced.

3. **Analysis setup complexity** – When working with multiple views, many datasets and multiple, alternative stratifications of each of these datasets, the selection of which dataset to show in which view with which stratification is a challenging task by itself.

4. **ID mapping** – Identifiers of cross-referenced datasets may need to be converted so that relationships can be resolved.

5. **Column-row relationships** – Cross-referenced datasets may be linked across columns or rows in the source files. As only relationships between either dimensions or rows can be analyzed, a transpositon may be necessary.

With the exception of the last two, all of these issues could also be relevant when only a single datasets is used. They are, however, likely to occur more often in multi-dataset scenarios. While the last two points are a purely technical challenge, the solutions of which were discussed in Chapter 4, the other issues need to be addressed by the visualization technique. To do so, we propose two techniques. The first is **StratomeX**, an evolution of VisBricks, that is targeted at visualizing the relationships of cross-referenced, stratified, multidimensional datasets. StratomeX introduces dependent columns to deal with dependent datasets, provides columns targeted at the analysis of individual categories and can handle datasets of unequal scale. To deal with the challenge of the complexity of setting up an analysis (choosing datasets, choosing stratifications of the datasets, and assigning them to views) that arises when working with large numbers of datasets, we propose the Data-View Integrator. The **Data-View Integrator** is a meta visualization that shows relationships between datasets and allows investigators to interactively assign stratifications and datasets to views.

While the visualization techniques presented in this chapter are valid for any kind of cross-referenced numerical or categorical datasets, StratomeX was developed for a specific, but highly relevant challenge: the classification of cancer subtypes in large-scale, heterogeneous genomics data. We present a task analysis elicited in semi-structured interviews with investigators, and show how StratomeX can be used to address these tasks.

Our approach is validated in case studies with investigators, who are domain experts. We report on our findings from one of these case studies, in which data from The Cancer Genome Atlas (TCGA)* for glioblastoma multiforme (GBM) [181] was used to characterize subtypes. Investigators were able to quickly reproduce known results from the literature, and to gain further insights into the data.

We begin by introducing the application scenario and discussing the involved tasks. We then give details on the visualization techniques, before describing the evaluation of our approach in the section on the case studies.

7.1 Application: Cancer Subtype Analysis

The discovery, refinement and characterization of cancer subtypes is the basis for targeted treatment and has implications for patient outcomes and patient well-being. Lately, much

*`http://cancergenome.nih.gov`

of the research on cancer subtypes is being performed with data from large-scale projects such as TCGA, which are generating comprehensive genomic and clinical datasets for thousands of patients. Recent studies [138, 192] have shown that an integrated analysis of different molecular data types generated by the TCGA project can indeed be used to discover subtypes and suggest molecular alterations relevant for therapeutic approaches.

Interactive visualization tools are crucial to fully exploit the potential of these large and heterogeneous datasets for subtype characterization. Such tools can greatly increase the efficiency of investigators, who currently are relying mainly on ad-hoc scripts and static plots. StratomeX is intended to become the tool of choice for investigators facing the challenges of working with large-scale genomic datasets for subtype classification. Before going into detail on the tasks investigators have to conduct during an analysis, we give a brief background on cancer and why the classification of subtypes is important.

7.1.1 Background on Cancer and Cancer Subtype Analysis

Cancer is a family of complex diseases that are caused by the accumulation of molecular alterations. These alterations are either genomic and affect the DNA sequence or epigenomic and affect other inheritable characteristics, such as methylation patterns of the DNA. Molecular alterations can lead to abnormal cell growth that results in tumor formation, invasion of nearby tissue, and often in growth of metastases in distant parts of the body.

Traditionally, cancers have been classified and named after the tissue or cell type where they originate, such as "breast ductal carcinoma" or "lung squamous cell carcinoma". However, cancers that originate from the same tissue or cell type are often not homogeneous with respect to their histology or the underlying genomic and epigenomic alterations, which gives rise to the notion of **cancer subtypes**. Cancer subtypes are highly relevant for patient treatment and prognosis, since the efficacy of cancer drugs can vary greatly among cancer subtypes, and patients with different subtypes often have very different survival chances. In recent years, the identification and characterization of subtypes has increasingly focused on genome-wide molecular data, which is now becoming available also for large numbers of patients through the efforts of consortia such as TCGA.

Our collaborators from the *Broad Institute of MIT and Harvard* are analyzing data from TCGA, which is a large-scale study designed to identify and catalog the molecular changes that are recurrent in large cohorts of cancer patients and therefore implied to drive tumor formation. TCGA aims to collect samples from at least 500 patients for each of over 20 different cancer types for a total of more than 10,000 patients. Several dozens of *clinical parameters* are collected for each patient, and all samples are subjected to extensive molecular profiling. The data generated for each sample includes genome-wide *gene mutation status*, *copy number alterations*, *mRNA gene expression* levels, *DNA methylation* levels, and *microRNA expression* levels (refer to Chapter 2 for a discussion of these data types).

TCGA data generation centers are using either microarray or next-generation sequencing technologies to generate aforementioned data types. The consortium maintains *Firehose*[†], a data analysis pipeline that is used to automatically preprocess the data and to

[†]http://gdac.broadinstitute.org

run a range of bioinformatics analyses. The analyses are jointly performed for all samples from patients with a particular cancer type and include various clustering algorithms for mRNA, microRNA, and methylation data, as well as identification of mutated genes and copy number changes.

Investigators who are working on cancer subtype identification and characterization use three types of results from the analysis pipeline:

1. Quantitative data matrices, such as gene expression matrices with measurements for all genes in all patient samples.

2. Clusterings on these matrices that stratify patients into mutually exclusive subsets.

3. Categorical data matrices for structural variation data. Examples are the copy number status containing the ordinal categories homozygously and heterozygously deleted, normal, lowly and highly amplified. Another example is mutation status data with the nominal categories mutated and not mutated. These datasets are collected for each gene in each patient. Entries for individual genes in these matrices can be used to stratify the patients.

In addition to the output from the data analysis pipeline, investigators include quantitative **clinical parameters**, such as time until a patient's death, in their analyses. They may also include patient stratifications in their analyses that were computed outside the main data analysis pipeline. Furthermore, **pathways** are used to investigate the role gene products play in molecular interactions.

By comparing different stratifications, it is not only possible to pinpoint the most sensible subset across different datasets as a "candidate subtype", but also to investigate the functional effects of these possible subtypes within pathway visualizations, as well as their effect on clinical parameters such as survival times.

7.1.2 Tasks

To understand the requirements of our collaborators for subtype analysis, we conducted a series of semi-structured interviews and evaluated recent publications that report findings of subtype analyses on TCGA data, for example [192] and [138], to complement the requirements elicited from the interviews.

Our working definition of a (candidate) subtype is a subset of patients obtained from one or more stratifications and we use the terms subset and subtype interchangeably. Technically the subsets or subtypes correspond to record groups.

The exploratory analysis can be roughly divided into two phases. In Phase 1, the investigators try to find stratifications of patients that are derived from multiple data types, for example an mRNA gene expression clustering that correlates with the mutation status of a particular gene. In Phase 2, they evaluate these subsets with respect to their functional and clinical implications. Tasks from Phase 1 and Phase 2 are addressed in an iterative fashion. More specifically, in Phase 1, investigators need to:

- Select combinations of stratifications and datasets from different data types for visualization.

- Evaluate how well two or more stratifications support each other.
- View and explore mRNA and microRNA expression or DNA methylation stratifications. If different patient subsets exhibit distinct patterns, this will be an indicator that there might be supporting evidence that these stratifications are indeed candidate subtypes.
- Refine stratifications by combining information from two data types, for instance by splitting a gene expression cluster based on the mutation status of a gene.

In Phase 2, investigators focus on the following tasks:

- Review the effect of a stratification on clinical outcomes, such as patient survival or tumor recurrence. If there are notable differences among subtypes, there might be clinical relevance.
- Determine whether the subtypes have a functional impact by viewing stratified molecular profiling data in the context of biological pathways. As an example, investigators are interested in pathways that are generally activated but deactivated in some subtypes.

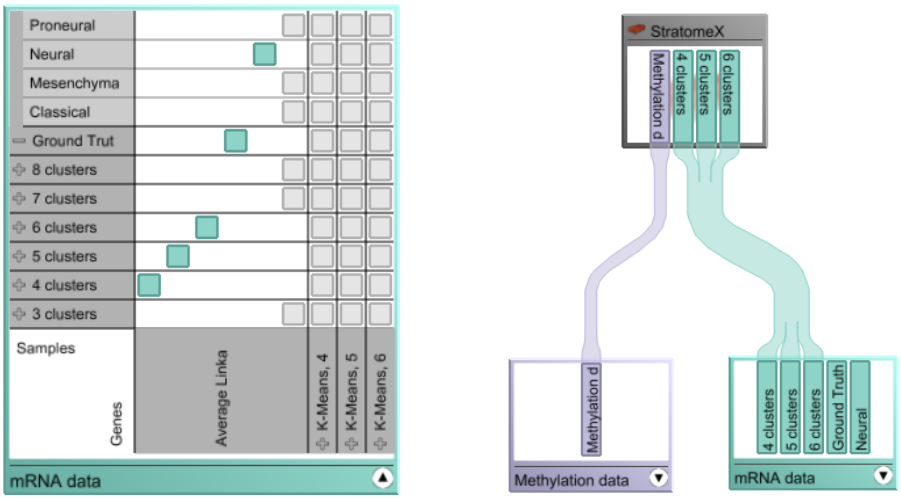
In addition, investigators will also perform quality control tasks, for example, by comparing different clusterings (same algorithm but different parameters; different algorithms) for a particular data type to evaluate how stable the clusters are.

7.2 The Data-View Integrator

Dealing with many different datasets, each with several stratifications that can be displayed in several views is challenging and should consequently be supported by a visualization tool. To this end we propose the **Data-View Integrator**, a meta visualization that serves two purposes. First, it orients the user by providing an overview of the datasets and the relationships among them. Second, it allows the user to dynamically configure combinations of stratifications and assign them to the views in which they can be analyzed.

Showing relationships between cross-referenced datasets is a quite common approach, especially in the context of databases, where database schemes are widely used. Integrating the viewing modalities in such a diagram is, in contrast, not widely supported, but not unheard of. North et al. envision *DataFaces*, interactive connections of visualization and data schemas, as future work [136]. This approach was recently realized in *Stack'n'flip* [179], which is also described in Chapter 8, as well as in *HIVE* [150]. We take up this idea and extend it to accommodate multiple stratifications.

By default, the Data-View Integrator shows a representation of the data model as a graph, where nodes correspond to individual datasets and edges represent shared identifiers among the datasets. In the subtype characterization application scenario, a unique patient ID serves as the primary key for referencing patients across all datasets. In addition, datasets such as mRNA and methylation data both contain patient IDs as rows and genes as columns and are therefore linked twice in the model. The nodes representing



(a) (b)

Figure 7.1: The two modes of the dataset nodes in the Data-View Integrator. (a) In the detail mode, the patient stratifications and gene clusterings are displayed as a matrix of possible combinations. By selecting one of the gray matrix cells, the user can interactively create a combination (cyan). (b) A view node connected to two dataset nodes which are in compact mode, listing only the existing combinations.

the datasets can be visualized in two modes. The compact overview mode shows only a caption for the dataset. The detail mode, shown in Figure 7.1(a), also shows the associated stratifications. In this example, multiple clustering results are loaded for both patient samples and genes, in addition to an external patient stratification labeled "ground truth".

As stratifications themselves are one-dimensional, views can only show combinations of record- and patient stratifications. Possible combinations are shown in a matrix layout when a node is in detail mode. By selecting a matrix cell, the user can indicate that she is interested in this combination, which is then highlighted and shown in a separate matrix column. The separate column results in an unambiguous horizontal position of a combination, which can be used to connect the combination.

In addition to datasets, views are represented in the graph. The user can directly assign which stratification combination she wants to explore in a view by using drag-and-drop. While some views can only show data from one dataset at a time and can consequently only be associated with a single combination, more sophisticated views, which support an integrated analysis across multiple datasets, can be assigned to multiple stratification combinations. Figure 7.1(b) shows a simple example where the dataset node is in compact mode, showing only selected combinations. Figure 7.2 shows a more complex scenario with multiple datasets and stratifications, as well as with multiple views, as they would be used for cancer subtype analysis.

The Data-View Integrator has two modes for the graph layout: it either utilizes the bipartite property of the graph and places the dataset nodes at the bottom and the view

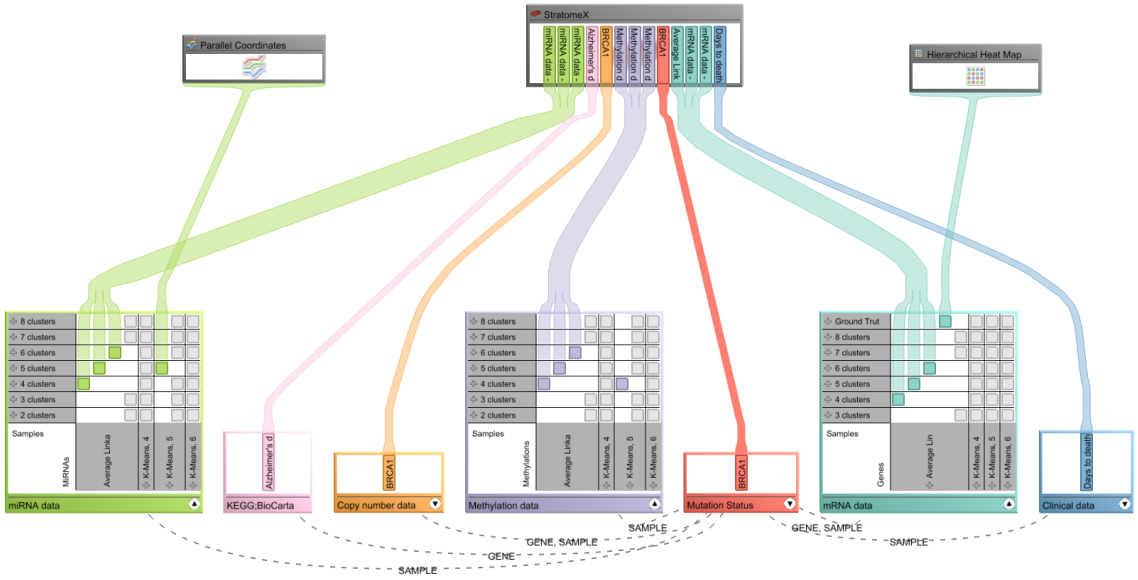


Figure 7.2: The Data-View Integrator showing the relationships between datasets as well as their association to views. Data sets and stratifications are shown at the bottom with the views placed above. Relationships between a selected dataset and all others are shown. Note that some views can show only one stratification, while others, like StratomeX, can show multiple.

nodes at the top, or alternatively arranges the nodes in a force-directed layout. While the bipartite variant is beneficial for data-view combinations, exploring dataset relationships is better supported by the force-directed layout.

7.3 StratomeX – A Subtype Visualization Technique

StratomeX is an evolution of VisBricks so it employs its visual encoding strategies. As shown in Figure 7.3, stratifications of datasets are arranged as columns side-by-side. The columns are split up into disjoint record groups representing either candidate subtypes, clusters, or categories – depending on the data type and stratification loaded. The multiform property of bricks means that the data can be encoded using various visualization techniques such as heatmaps, parallel coordinates plots, or histograms, which can be switched on demand. When using heatmaps, the height of a brick encodes the number of patients it contains, while other bricks are of static height. Ribbons connect the bricks of neighboring columns, encoding how many patients they share. This is illustrated in Figure 7.3. It should be noted that for this application scenario, the patients are the records (i.e., the patients are vertically grouped) and the genes, metylations, etc. are the dimensions. This is in contrast to all cases and figures up to this point in this thesis, where genes were used as records and the dimensions corresponded to samples, experiments or patients.

As different datasets can contain disjoint sets of patients, the height of the bricks cannot be used to compare absolute values, even when visualization techniques that have a height proportional to the amount of data they encode are used. We have chosen

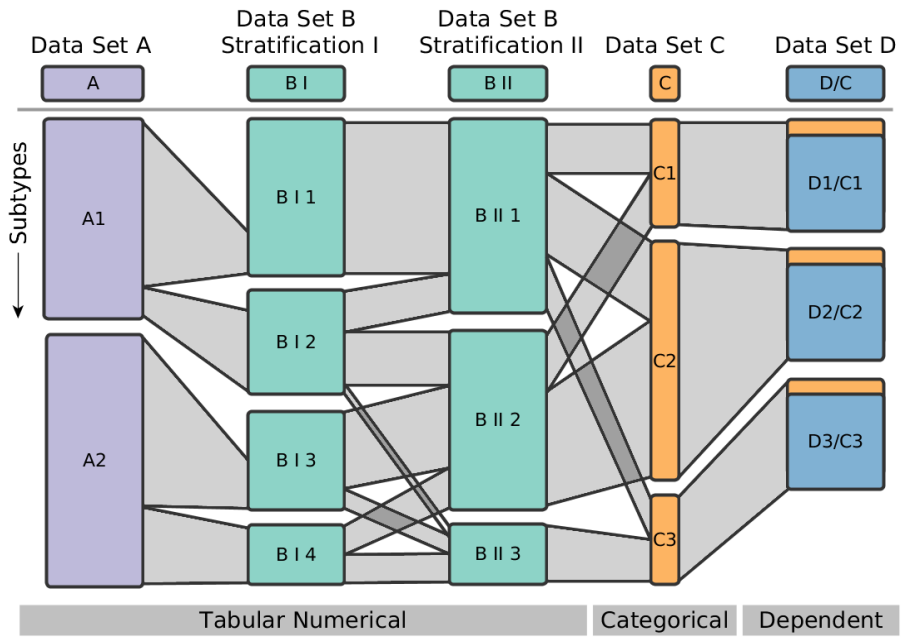


Figure 7.3: Schematic comparison of five columns. The first three columns show stratifications of tabular, numerical datasets, where the second and third show the same dataset only with different stratifications. The fourth, orange column represents a categorization. The rightmost column illustrates the concept of dependent subsets, where the groups are based on the stratification of another column. The ribbons between the subsets indicate how many patients are shared between them. For instance, all patients of brick `BI1` are contained in brick `BII1`. `BII1`, however, also contains patients from the second brick in the first column.

relative heights, since investigators are primarily interested in the relative relationships; additionally, relative heights optimally utilize the available space. This will be valid if the dataset constitutes a representative subset of the population. As long as two neighboring columns contain the same patients, the outer edges of the ribbons connecting them will be parallel. For disjoint sets of patients, however, the height at the beginning of a ribbon may not be the same as at its end, as shown between the first and second column in Figure 7.3. In this example, Data Type B contains more patients than Data Type A, leaving parts of the sides of the bricks unconnected.

7.3.1 Column Classes

One aspect that distinguishes StratomeX from VisBricks is that it can deal with multiple heterogeneous datasets. While VisBricks does not distinguish between types of columns, we introduce two new classes of columns for StratomeX, which are needed for multi-dataset analysis in general and for the cancer subtype analysis tasks in particular.

Table columns – The most fundamental class of columns, the only one available in VisBricks, uses the stratifications to create record groups of multidimensional datasets. For the subtype identification task, the heatmap representation is best suited and therefore

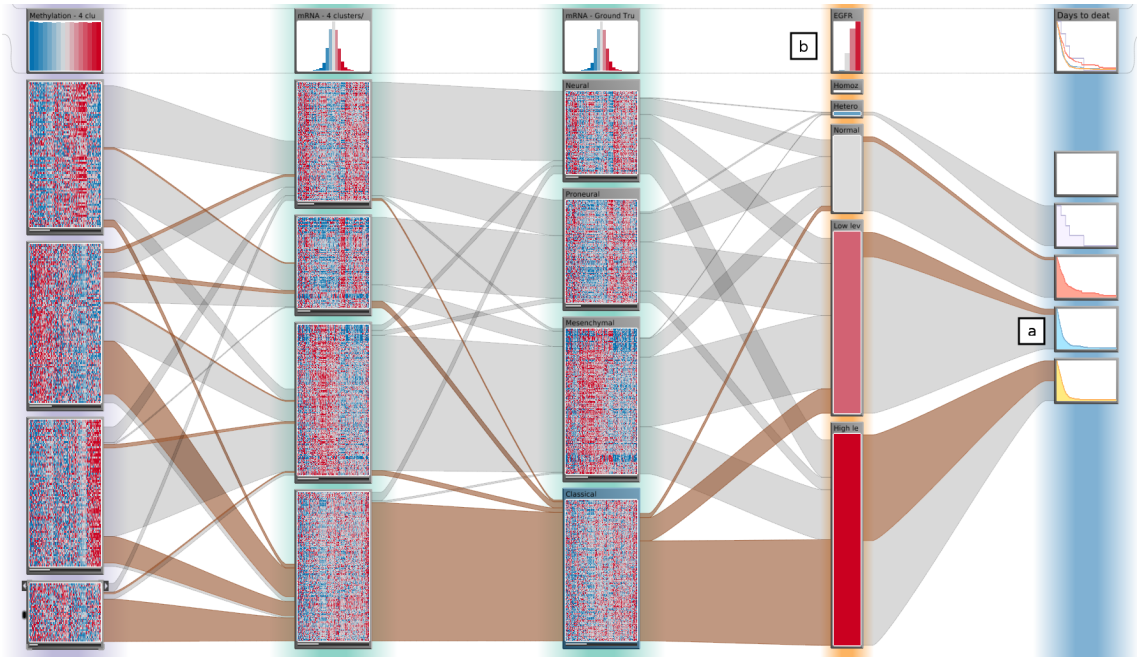


Figure 7.4: StratomeX configured as illustrated in Figure 7.3. The heatmaps in the bricks allow the investigator to judge the homogeneity of the subset, the header bricks at the top show the name of the column and an overview of the data of the dimension group. In the fourth column we see a stratification based on the categories for copy number variation of *EGFR*. The rightmost column shows *Kaplan-Meier* plots for "days to death" as dependent bricks for the copy number-based stratification. Note that patients with amplifications of *EGFR* have far worse outcomes compared to patients with no copy-number alterations.

chosen as the default. The stratification into subsets is in most cases not fixed, often alternative stratifications exist. This can make manual refinement of the stratifications necessary. The plausibility of a particular stratification is judged by investigators using the embedded visualizations and the relationships to other stratifications in StratomeX. Figure 7.3 shows a stratification for one dataset in the first column, and two stratifications for another dataset in the second and third column.

Categorical columns – Categorical columns represent an unambiguous stratification of patients based on a single attribute. An example for a single attribute as a category is the mutation status of one particular gene of interest, which can be *mutated* or *not mutated*. Categorical columns contain no visualization of the underlying data other than a color assigned to every category, but have permanently visible labels showing the name of the category. If, in contrast, multiple categories should be shown in one brick, table columns will be required instead. Using only categorical columns and categorical data in StratomeX would result in a visualization technique similar to Parallel Sets [104].

Dependent columns – In many cases it is of great interest to explore the effect of a stratification of one dataset on another one. StratomeX allows the user to do this by introducing dependent columns. The dependent columns use the same stratification as

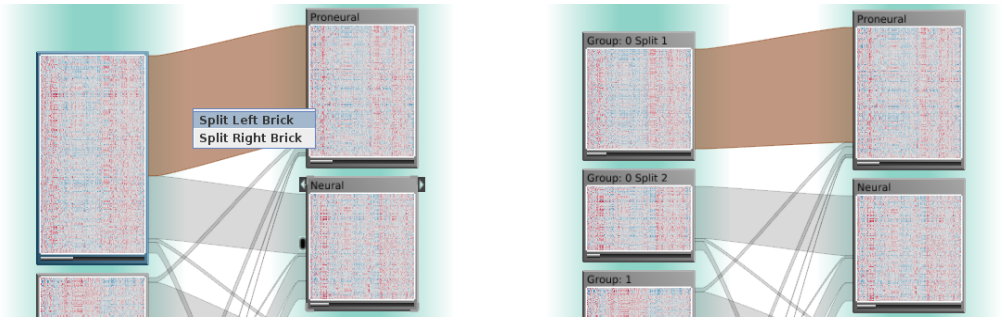
their source column, but show the data of the dependent dataset. As a consequence, the ribbons connecting the source column always connect exactly two bricks. An example for a dependent column is shown on the far right in Figure 7.3. Dependent columns are crucial for two tasks in this application context: to explore survival plots, and to investigate pathways. By using multiple *Kaplan-Meier* curves [146] next to candidate subtypes of mRNA expression, investigators can explore whether the stratification derived from, for instance, clustering of gene expression data, has effects on the clinical status of patients. The small multiples of the Kaplan-Meier curves could, for example, show that the disease-free survival in one subtype is significantly lower than that of another. Figure 7.4 (a) shows a case where patients with a normal copy number status of *EGFR* have a better chance of living longer than those where *EGFR* is amplified. Dependent pathway columns can be used to judge whether there might be different behavior between subtypes in the biological processes that the pathways represent. By placing multiple small thumbnails of pathways, one for each subset, next to an mRNA expression dataset, and overlaying the average expression of the group onto the gene nodes of the pathways, investigators can easily compare the effects of the subtype on the pathway. To visually amplify and make them stand out even in the thumbnail-sized small multiples, we enlarge the expression overlays. An example of pathway small multiples is shown on the right of Figure 7.8. Pathways themselves are graphs, so they do not fall into the cross-referenced data category, but rather into the general heterogeneous category. In this context, however, pathways are considered a special layout for the mapped gene-expression data, which is of the cross-referenced type. They therefore integrate nicely into StratomeX.

7.3.2 Visual Encoding Details

Beyond the high-level visual encoding strategy described above, StratomeX contains a series of important additional encodings that support the analysis tasks. Similar to VisBricks, StratomeX is designed to follow the visual information seeking mantra – "overview first, zoom and filter, then details on demand" [168]. In the following we discuss the features added to enable multi-dataset analysis.

Overview – To facilitate the association between the columns in StratomeX and the dataset nodes in the Data-View Integrator, we use a combination of color coding and labels. The columns have a halo in a color that corresponds to the color of the dataset node in the Data-View Integrator. Every column has a header brick labeled with the name also used in the Data-View Integrator, as shown at (b) in Figure 7.4. The header brick shows a small summary view representing the whole dataset. The type of view shown depends on the dataset and user preference. For tabular and categorical data the header brick shows a histogram of the dataset by default. Pathway columns show the pathway with the average expression encoding of the whole dataset overlaid. Clinical survival data uses a summary Kaplan-Meier plot that overlays the survival curves of each subtype.

Zoom and Filter, Interaction – As subtypes are rarely based on only one factor (and therefore one data type), it is crucial to be able to refine candidate subtypes by splitting and merging bricks. StratomeX supports interactive splitting of bricks based on the ribbons connecting them to other columns, which is illustrated in Figure 7.5, as well as merging of multiple bricks of the same column. The user can add labels for candidate



(a) (b)

Figure 7.5: Split operation based on the ribbons between three bricks. (a) The split operation is triggered using the context menu of a ribbon connecting two bricks. (b) After the split operation the chosen brick was split into two bricks: The first contains all records the two bricks that were connected by the chosen ribbon share, while the second contains all others.

subtypes, which are then shown at the top of the subtype brick. Notice that, in contrast to VisBricks, the GUI elements of a brick are not considered part of the relative height of a brick. Only the portion, where the actual view is shown is also connected with ribbons. This makes it possible to handle bricks with very few or no records. StratomeX initially uses a sorting strategy based on average values, but allows users to arbitrarily arrange bricks within the columns, which can be used to minimize crossings of the ribbons.

Details on Demand – While the process of characterizing subtypes is mainly conducted by investigating global trends in the overview, it is often also necessary to explore some part of the data in detail. If, for example, the small multiples of the pathways show differences in the mapping of the genes between the subsets, a detail-on-demand strategy will be necessary to identify the genes. StratomeX facilitates this by enabling investigators to create focus-duplicates of arbitrary bricks, as illustrated in Figure 7.8.

7.4 Scalability

As StratomeX is an evolution of VisBricks, the scalability discussed for VisBricks in Section 6.3 is largely applicable. However, in contrast to VisBricks, StratomeX uses more space-efficient bricks, hiding menus when they are not required. Also, the performance of views has been improved: The embedded views will switch automatically from texture-based, static views to fully interactive visualizations, if enough space is available. For the case studies described below, seven datasets were loaded. With the exception of the pathways, each contained between 300 and 550 samples, with 1500 genes each for the expression datasets, and between 5000 and 6000 each for copy-number and mutation status data. This makes up a total of roughly six million data points. In an analysis scenario with five to seven columns, roughly one million data points are rendered simultaneously, making it a very effective visualization tool for the visual analysis of large-scale data.

7.5 Case Studies

To evaluate our approach, we asked our collaborators from the *Broad Institute of MIT and Harvard* to use StratomeX to explore data from one of the TCGA cancer types that they are currently analyzing. We prepared datasets for *Glioblastoma Multiforme* (GBM) and *Breast Invasive Carcinoma* (BRCA) based on the output of *Firehose*, the TCGA analysis pipeline and added additional, "external" stratifications provided by our collaborators. Here we only report case studies from the GBM dataset, as none of the findings for the BRCA dataset have been published by the TCGA consortium so far. The following observations and findings were made during the evaluation sessions with our collaborators.

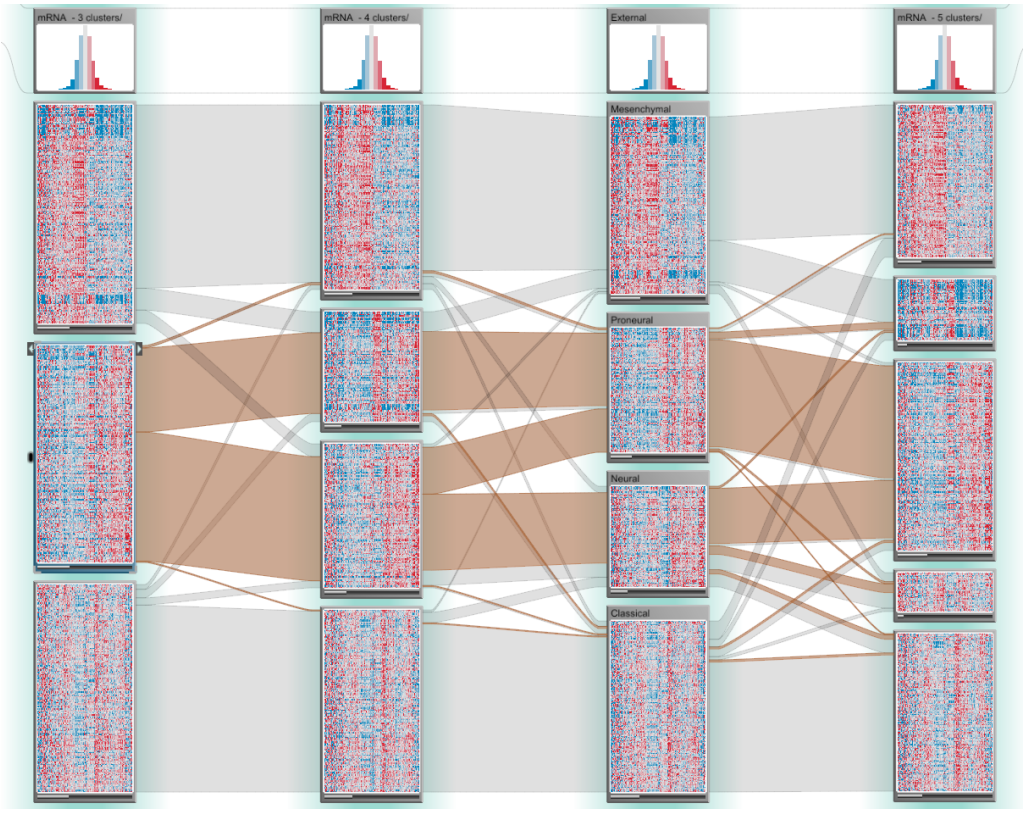


Figure 7.6: Clustering comparisons. Columns 1, 2, and 4 show clusterings from the analysis pipeline with three, four, and five clusters respectively. Column 3 shows a stratification of the patients based on subtypes identified by Verhaak et al. (from top: *mesenchymal, proneural, neural, classical*). The clustering result of the first column was reported to be the best result by the analysis pipeline, however, we know from the literature, that there are four subtypes of glioblastoma, which are shown in column 3. It is also known that the proneural and neural subtypes are hard to distinguish based on their gene expression pattern, which explains the observed problems of appropriately separating them in columns 1, 2 and 4.

7.5.1 Comparing Clusterings

Even though the TCGA analysis pipeline reports a single "best" clustering for each mRNA, microRNA, and DNA methylation data matrix, clusterings with different numbers of clusters are available as well. Since Verhaak et al. [192] identified four mRNA gene expression subtypes, but the analysis pipeline reported three clusters as the best result for mRNA expression data based on one of the implemented clustering algorithms, we were interested in how the clustering for three, four and five clusters compared to the corresponding classification by Verhaak et al. When Verhaak et al. performed their analyses, data from only slightly more than 200 GBM patients was available, but one of our collaborators had access to a more recent classification that assigned the current population of around 530 GBM patients to the Verhaak et al. subtypes, which we used in this and all other case studies described here. The stratifications based on the clustering and on Verhaak et al.'s classification are shown in Figure 7.6. The first observation that we made based on the salient ribbon patterns was that one of the subsets from the three-cluster solution was split into two clusters in the four-cluster solution, but that almost all patients from these two clusters make up a single cluster in the five-cluster solution. This is possibly an artifact of the clustering algorithm, but our collaborator confirmed that this might also be biologically meaningful because of a second observation that we made: said two clusters in the four-cluster solution are a mix of the *neural* and *proneural* subtypes identified by Verhaak et al., whereas the other two clusters almost exactly correspond to the *classical* and *mesenchymal* subtypes. This indicates that the clustering computed by the analysis pipeline is a reasonable and meaningful solution. Verhaak et al. also reported that the tumors in the neural and proneural subtypes exhibit similar gene expression patterns, which are not found in the other two subtypes. This is one possible explanation for why neural and proneural subtypes are harder to separate by clustering than the other types.

7.5.2 Combining Gene Mutation Status and Methylation Data

Noushmehr et al. [138] used clustering of DNA methylation profiles to identify three GBM subtypes, one of which is based on hypermethylation of certain regions of the genome, implicating that gene expression in those regions is repressed. They also found that this subtype is associated with mutations of the gene *IDH1* and mostly falls within the proneural subtype. When in one of our evaluation sessions our collaborator was interested in studying this methylation subtype, he quickly realized that it had not been detected in the clustering from the analysis pipeline that created three clusters. None of the clusters was strongly associated with either *IDH1* mutations or the proneural subtype. Using the Data-View Integrator, we easily added the other clusterings to the view. Our collaborator pointed out that one of the clusters from the clustering with eight clusters had a distinct methylation pattern and only contained patients with an *IDH1* mutation, as is evident when looking at the mutated category at the bottom right of Figure 7.7. We then used this cluster to split the methylation stratification with originally two clusters into three candidate subtypes. We hypothesized that the newly created patient subset contained many patients with the Noushmehr et al. subtype, both due to its strong association with the *IDH1* mutation and the large overlap with the proneural subtype. Our collaborator suggested to confirm this using the survival data, as Noushmehr et al. reported better

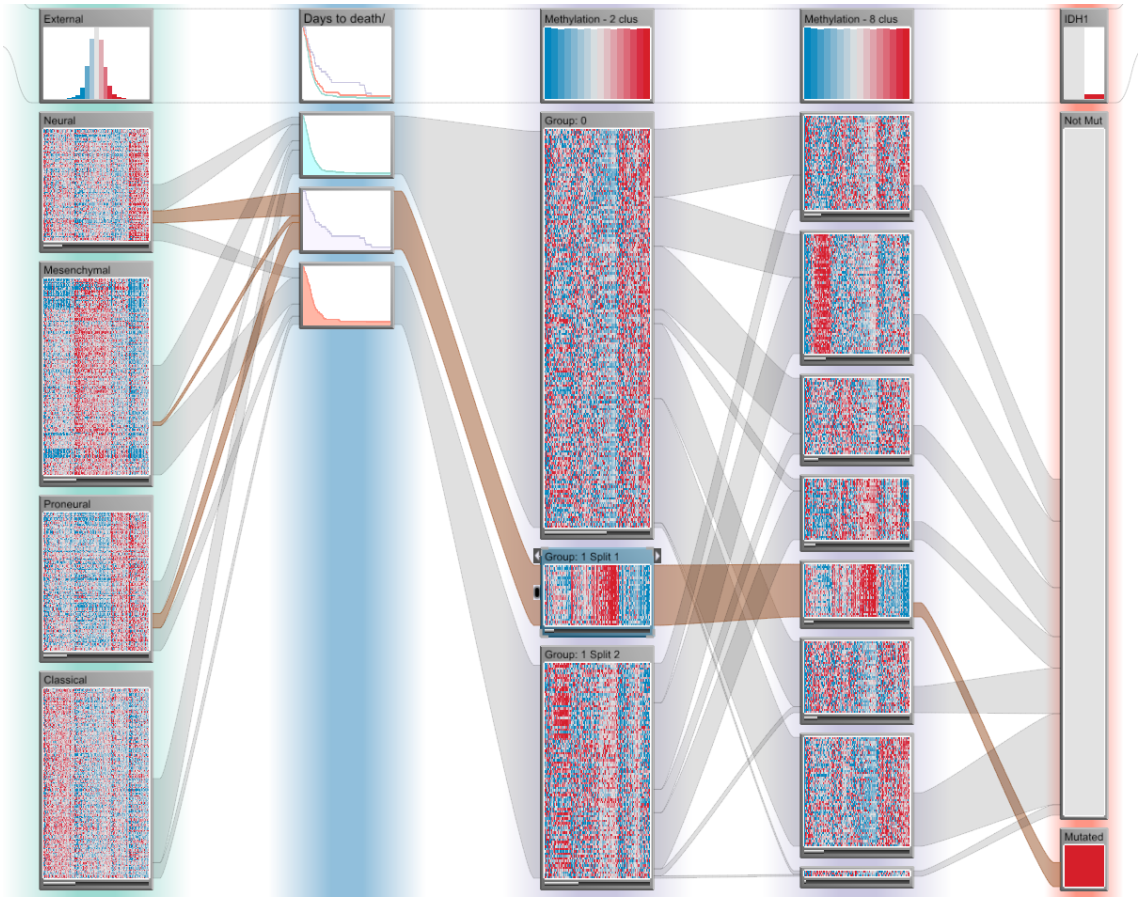


Figure 7.7: Subtypes based on methylation data. Column 3 shows a manually refined stratification of methylation data, which was created by splitting off a part of the original clusters based on the mutation status of *IDH1* shown in Column 5. The created brick reveals a characteristic expression pattern overlooked by the algorithm. Only in the eight-cluster case shown in Column 4 was the clustering algorithm able to detect this pattern, which is surprising, as the pattern is highly salient when visualized. Column 1 shows mRNA gene expression subtypes identified by Verhaak et al. From the ribbons it is clear that the subtype based on the methylation pattern and the *IDH1* mutation has a significant overlap with the proneural and neural subtype. Column 2 shows patient survival outcomes (*days to death*) and was created as a dependent column of Column 3. The better survival outcomes reported by Noushmer et al. are evident in the associated Kaplan-Meier plot. Large parts of the bricks showing mRNA expression and copy-number data are not connected because no methylation data is available for about half of the samples.

survival outcomes of patients with this subtype. Indeed, the newly created patient subset seemed to have better survival outcomes than patients in the two other subsets, as can bee seen in the Kaplan-Meier plot in the second brick from the bottom in the second column in Figure 7.7. This example emphasizes the importance of interactive refinements of stratifications that is supported by StratomeX.

7.5.3 Evaluating the Functional Impact of Subtypes

In one of our evaluation sessions we looked into the effect of the Verhaak et al. gene expression subtypes on molecular processes that are known to play a role in gliomas, which is the family of brain cancers that GBM is part of. We opened the "glioma" pathway from *KEGG (Kyoto Encyclopedia of Genes and Genomes)* [91] as a dependent column to see whether there are any differences in the expression levels of these pathways when stratified according to the subtypes. The small multiples very clearly showed that the glioma pathway indeed has different activation patterns across the four subtypes, as can be seen at the locations indicated by the arrows in Figure 7.8. In particular, we noted that there was a striking difference between the proneural and the classical subtype in the left part of the pathway. With the help of a focus duplicate of the pathway, we were able to

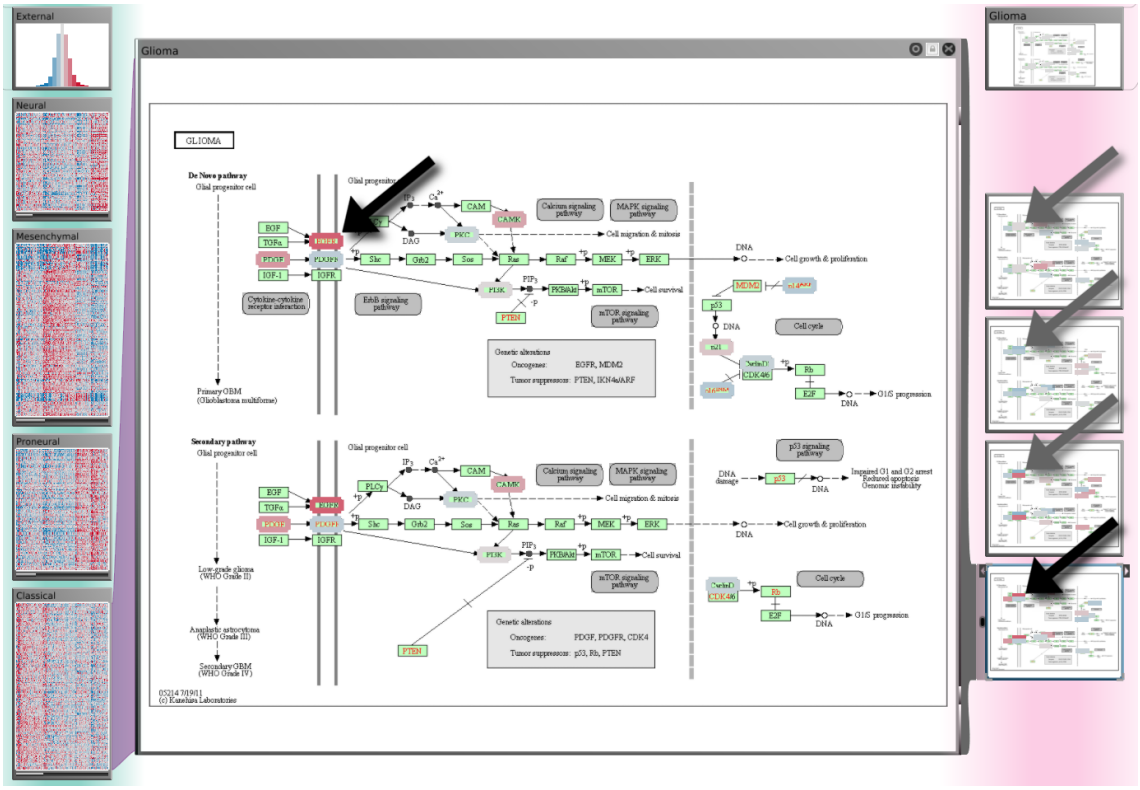


Figure 7.8: Subtypes in the context of pathways. Column 1 shows mRNA gene expression subtypes identified by Verhaak et al. (from top: neural, mesenchymal, proneural, classical). The dependent Column 2 shows small multiples of the *Glioma* pathway from *KEGG* overlaid with the average gene expression levels for each subtype. The detail view in the center shows the same pathway enlarged with the gene expression levels for the classical subtype. The arrows indicate a part of the pathway where we observed notable differences in gene expression levels between the subtypes. Note that not all genes in the pathway have been mapped since the gene expression data matrix only contains a subset of the most variable 1500 genes in the dataset.

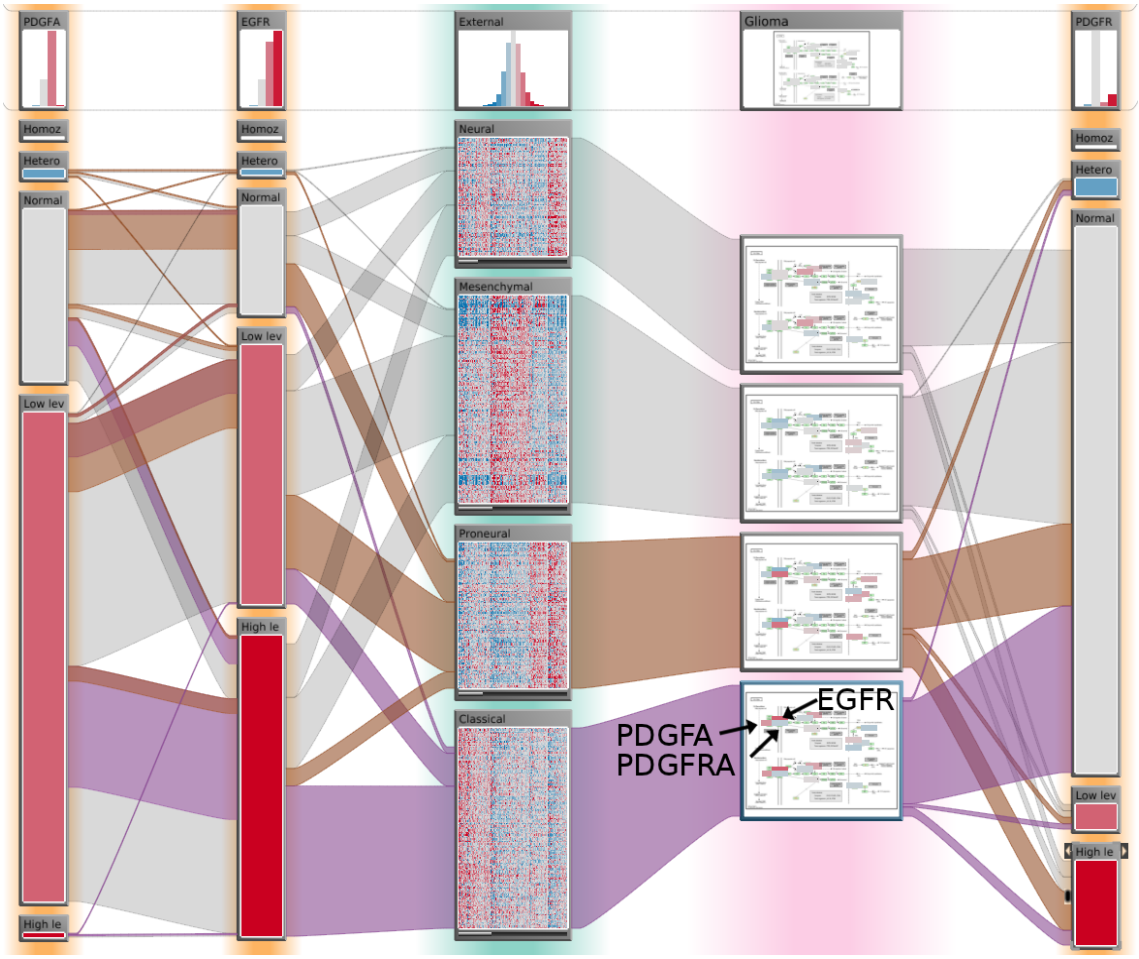


Figure 7.9: Copy-number status of the genes identified to be involved in the *Glioma* pathway. The copy number status of *PDGFA* (first column), *EGFR* (second column) and *PDGFRA* (fifth column), their locations within the small multiples, and their relationships to the proneural and the classical subtypes. The proneural subtype is brushed in brown, the classical in magenta. A strong correlation of the proneural subtype to copy-number amplifications of *PDGFRA* is evident, even though the majority of samples have no altered copy-number status of *PDGFRA* at all. The increased expression of *PDGFRA* is clearly visible in the second pathway from the bottom. The strong relationship between high-level amplifications in *EGFR* and low level amplifications in *PDGFA* to the classical subtype can be observed when looking at the magenta brush. Both, *PDGFA* and *EGFR* have increased expression levels in the pathway at the bottom, where the expression of the classical subtype is mapped.

identify the genes that are showing the most notable differences between the classical and proneural subtypes in the glioma pathway: *EGFR* and *PDGFA* are upregulated whereas *PDGFRA* is downregulated in classical GBM, and vice versa in the proneural subtype. This observation is probably due to a finding that Verhaak et al. reported, namely that increased *EGFR* copy numbers are a hallmark of the classical subtype, whereas copy number

amplifications of *PDGFRA* are a characteristic of the proneural subtype. These increased in copy numbers are likely responsible for the increased gene expression levels that we observed here. The relationships of *PDGFA*, *EGFR*, *PDGFRA* and the classical and proneural subtypes are shown in Figure 7.9. The classical subtype is brushed in magenta, while the proneural subtype is brushed in brown. We can see that high-level amplifications of *PDGFRA* on the far right are clearly correlated with the proneural subtype, as the widest ribbon connects it with the pathway showing the proneural expression levels. In contrast, the classical subtype is strongly associated with high-level amplifications of *EGFR* and low-level amplifications of *PDGFA*, where the proneural type is involved to a much lesser extend.

7.5.4 Discussion

In general, our collaborators noted that the brick and ribbon metaphor to visualize patient subsets and their relationships across different stratifications feels natural and intuitive. They also told us that the combination of small multiples with details on demand is very useful, in particular for the pathway maps. A very positive outcome of the evaluation sessions with our collaborators was that in all cases they asked us to load further data that they wanted to explore with StratomeX. They also made suggestions on how to improve the tool by integrating further analyses, for example to compute statistical significance values for observed differences in patient outcomes.

7.6 Conclusion

In this chapter, we have discussed how the techniques proposed in the previous chapters can be extended to be able to handle multiple, cross-referenced datasets, resulting in the StratomeX technique. We have introduced two new classes of columns, the first for categorical data, the second to visualize data based on stratifications of other columns. The latter allows us to observe the effects of a stratification derived from one dataset or dimension group on meta data. We also described the Data-View Integrator, a visualization technique that makes it possible to conveniently configure which datasets in which configuration should be shown in which view. We thereby address the complexity of setting up visualization scenarios when working with many datasets, configurations (stratifications) and views.

StratomeX and the Data-View Integrator were designed in close collaboration with domain experts and tailored to the task of cancer subtype analysis, which is evident due to the availability of domain specific views, such as pathways and Kaplan-Meier plots. Nevertheless, the underlying concepts are of general validity for the visualization of relationships of cross-referenced datasets. The extensive case studies validate not only the utility of StratomeX, but also, as StratomeX builds on Matchmaker and VisBricks technology, of the techniques introduced in the previous chapters. We can therefore conclude that StratomeX is a tool suitable for the analysis of cross-referenced datasets and that Hypothesis III is fully supported.

Chapter 8

Visualizing Relationships in General Heterogeneous Data

Contents

Visualizing general heterogeneous data is a challenging, but potentially highly rewarding undertaking [98, p. 19], [183, p. 100]. From a visualization research perspective, the conceptual and technical hurdles to provide seamless data visualization across the boundaries of individual datasets are not yet overcome, although they have been discussed for over a decade [190]. We distinguish between two classes of heterogeneous datasets:

1. **Structured heterogeneous data** – Structured heterogeneous data is systematically collected and based on an established data model. Examples are multiple datasets collected and referenced to patients in a medical scenario, such as magnetic resonance imaging data, blood panels, x-ray imaging, etc.; or datasets that are collected in astronomy or physics observations and experiments. What these datasets have in common is that they are recorded or collected with intent, and their relationships to one another is well-defined.

2. **Unstructured heterogeneous data** – Unstructured data is data collected from a variety of sources, where the relationships among datasets, the relevance and the validity of the data are unknown. Typical examples are intelligence analysis scenarios, where vast quantities of documents, videos, images, etc., from multiple sources are available and need to be analyzed.

Cross-referenced data, which was the topic of the previous chapter, is a form of structured heterogeneous data with an underlying data model. The data model is explicitly visualized using the Data-View Integrator, discussed in Section 7.2. While structured heterogeneous data that goes beyond cross-referenced data is not at the core of this thesis,

it is a closely related domain. We describe a model-based approach for analyzing general heterogeneous data sources in the next section. The focus of the model-based approach is to provide a framework for orientation and guidance in a structured, heterogeneous data landscape.

To make sense of unstructured heterogeneous data, data mining and knowledge extraction methods are a necessary precondition for visual analysis. As the focus of this thesis is structured data, we do not discuss unstructured heterogeneous data analysis methods per se. However, we describe visual linking to show relationships among multiple datasets, which do not require a data model and are therefore equally applicable to unstructured heterogeneous data. Visual linking can be used in homogeneous, single-dataset, single-view scenarios as well as in multiple-application, multiple-datasets applications. Still, its benefits are most striking in the latter two cases. In Section 8.2, we describe an approach for visual linking across applications, and a method that optimizes the routing of visual links.

The author of this thesis contributed to all the research presented in this chapter. The main contributions were made by the first authors of the respective papers.

8.1 A Model-Based Approach to Visualizing Structured Heterogeneous Data

Creating data models for multiple datasets is a common method, as already discussed in Chapter 7. Most modeling approaches are limited to relationships among datasets. We believe that modeling dataset relationships is necessary, but not sufficient, and introduce a hierarchical, three-level modeling approach.

The first level, the **setup model**, contains the *datasets*, *operators*, *visual-*, and *analytical interfaces* available in an analysis scenario. Operators describe generic operations that can be executed using either visualization and interaction, or analytical processes. An example for an operator is "create grouping" which could be done either manually in a visualization interface, or automatically, using, for example, a clustering algorithm. The setup model not only lists these parts, but defines relationships between them, containing the information which datasets have a relationship, which analytical or visual interface can be used to analyze a dataset, and which analytical or visual interface can be used to execute an operation.

Based on the setup model, it is possible to provide *orientation* to users. Orientation is important in complex scenarios, since the possible transitions between datasets and the choice which interface to use for a task is not obvious. This level of support is comparable to navigating with a paper-based map, where information on *what is where* is provided, but no instructions on *what to do next* are available.

The setup model can be extended by domain-specific information, yielding a **domain model**. The domain model enriches the setup model by adding tasks from a domain, assigning these tasks to datasets and assigning operators to the tasks. On top of the domain model, the **analysis session model**, describing a concrete workflow composed of the available tasks with a specific analysis goal, can be added. Based on the full three-stage model, *guidance* can be realized, meaning that users can be lead through an analysis

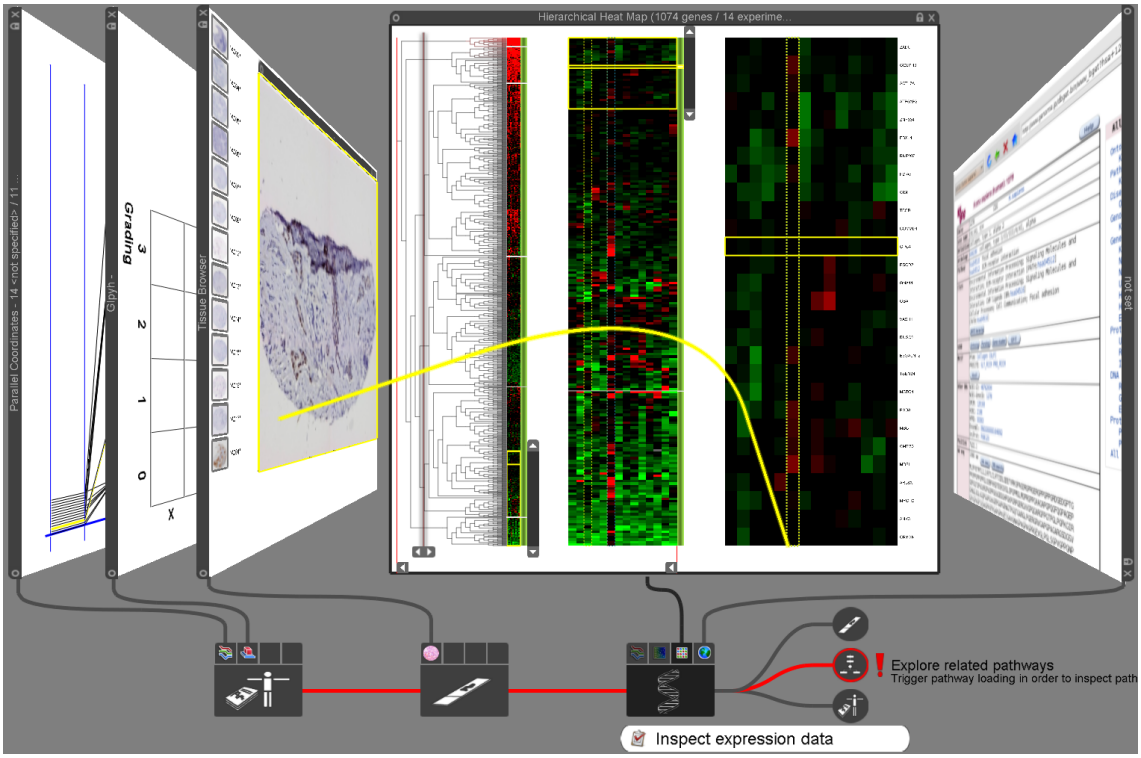


Figure 8.1: Guided analysis in Stack'n'flip. The upper part shows visualization of datasets, where the current focus dataset is at the center, while previous visualizations and candidates for next visualizations are stacked at the sides. The most important contribution of Stack'n'flip is the graph at the bottom. The graph shows the dependencies between the datasets, the associations to the currently open views, the available interfaces, and the recommended paths.

process automatically.

While orientation can be provided in virtually any analysis scenario, model-based guidance is useful for repetitive tasks, as a workflow for the analysis has to be defined, which is impossible for an open, exploratory analysis. Guidance has its place in scenarios such as diagnostics, where a standardized procedure ensures the quality of the process. Of course, using algorithmic methods to create a model for suggestions is also conceivable, which would allow to employ guidance in exploratory analysis scenarios.

We realized the model in **Stack'n'flip**, a prototype for an application in a clinical scenario. Stack'n'flip integrates patient meta-data, histological tissue slices, gene-expression data, data from online databases, and pathways. Figure 8.1 shows Stack'n'flip during an analysis. The screen is divided into two parts: the upper parts shows visualizations of the data, while the lower part shows a graph containing dataset dependencies, dataset-interface assignments and recommended paths. The two parts are connected by visual links, associating the views directly to the datasets. In the bottom right corner, we see the recommended next step highlighted in red. Notice that recommended paths do not need to be followed. A user can decide to branch off into another path, revisit previous steps, and then follow the recommended path at a later time. For details on the modeling

and on Stack'n'flip refer to the original paper [179], or to the thesis by Marc Streit [176].

While Stack'n'flip demonstrates how guidance can be realized based on an authored model, widespread adoption of similar implementations is unlikely, due to the infeasibility of integrating all possible analysis tools for a heterogeneous analysis scenario into a single tool. We believe that the integration of multiple tools in a heterogeneous analysis scenario is more likely to succeed. While a deep integration of arbitrary tools is yet unsolved, we propose a first step towards it in the following section.

8.2 Visual Linking for Heterogeneous Data

When analyzing multiple related, but heterogeneous datasets, it is likely that multiple software tools are involved. As discussed previously, it is not feasible to integrate all possible visualization and analysis tools into a single framework. Nevertheless, integration of these tools is desirable. Examples for important features that should be supported across applications are filtering, i.e., elements removed in one application should also be removed in another, synchronized scrolling, data loading, i.e., it should be possible to load a dataset in one application of a data item selected in another, or across-application highlighting, so that relationships of the datasets are explicit.

While all these features and many others are important, we chose to focus on across-application highlighting, since it is a core topic of visualization, while the others are mainly human-computer interaction and software engineering tasks.

Our approach to across-application highlighting is to employ visual links. We defined visual links as "continuous shapes such as connection lines, curves, or surfaces that connect or surround multiple related pieces of information, thereby augmenting a base representation". In Chapter 3 we have distinguished two classes of base representations: those that are aware of the visual links and adapt to them, and those that do not. The Matchmaker, VisBricks and StratomeX techniques belong to the first class. The techniques presented in the remainder of this chapter are intended for base representation of the second class, which do not adapt to visual links. We will first discuss how visual links can be employed across applications, followed by a discussion of possibilities to improve the visual quality of visual links.

8.2.1 Visual Links Across Applications

Very few visualization tools are able to integrate multiple, simultaneously running applications. Those that do, typically share a common database, through which the communication is handled. A prominent example for such a tool is *Snap-Together* visualization [137]. However, using a shared database requires a shared data model and consequently is not applicable for unstructured heterogeneous data. While a shared data model enables a level of integration not achievable with unstructured data, we will demonstrate that it is not necessary for across-application highlighting. Also, while using color for highlighting is predominant in the related work, it is often not the method of choice, as discussed in Chapter 3. Especially in multiple heterogeneous applications, where different color schemes may already be employed, or where the color coding cannot be changed easily,

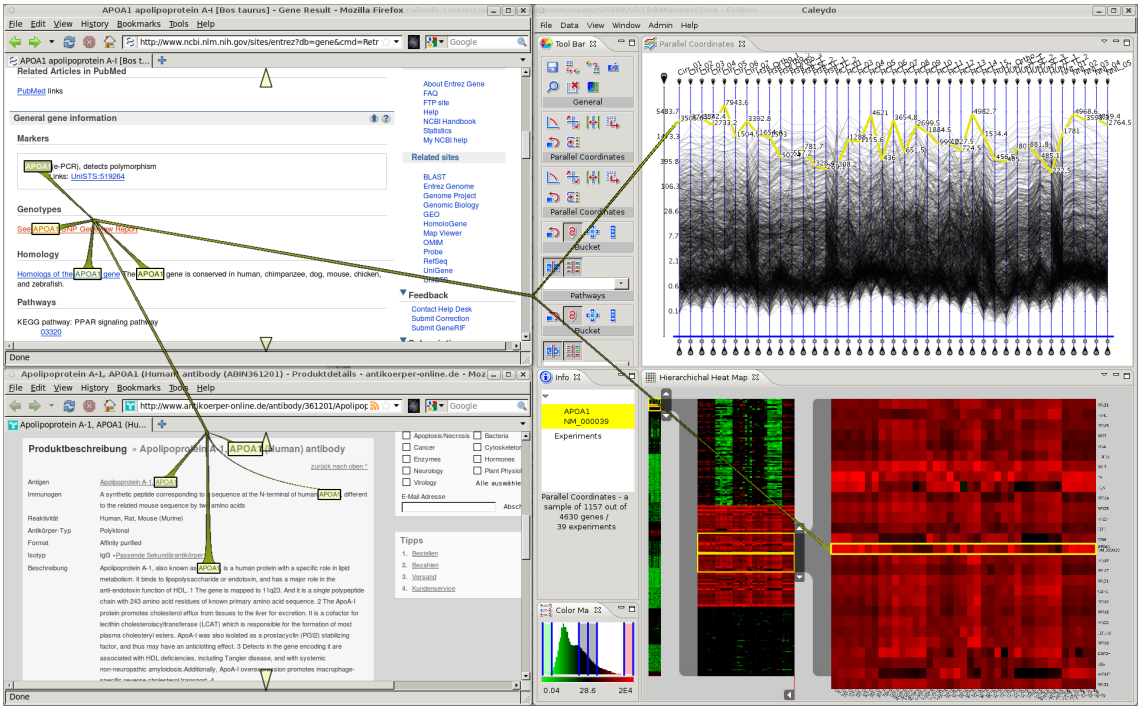


Figure 8.2: Visual links connecting highlighted elements in four independent applications. The links are bundled on a per-window basis, so that edges between views are minimized. Off-screen visualizations (the arrows at the top and bottom of the left windows) indicate hidden matches. This example shows gene expression data in a heatmap, a parallel coordinates view and two browser views containing information on the highlighted genes.

color coding is not the right choice for highlighting. We therefore propose to employ visual linking as an across-application highlighting solution.

Our approach is based on a lightweight background application collecting and distributing selections, a rendering application, which is responsible for drawing the visual links, and interfaces to the integrated applications. Figure 8.2 shows the results: Visual links are rendered on top of four independent applications. The right two windows contain instances of Caleydo, showing separate visualizations for gene expression data, the left two windows show web browsers containing information on the selected entries. To reduce visual clutter, visual links are bundled on a per-window basis. The technique also provides off-screen visualization: Small triangles at the windows' edges hint at information hidden from the current view, as can be seen at the top and bottom edges of the two browser windows in Figure 8.2.

Technically, there are three levels of integration for applications providing and receiving selections for visual links. Applications can provide *direct support* by implementing an interface, can be extended using *plugins*, which is the preferred way to integrate common applications such as browsers or office suites, or, for web-based content, can be based on *mashups*, thereby combining web-applications with the visual links interface. We are currently working on a completely non-invasive alternative using *text recognition*, which

will be able to integrate arbitrary applications.

We evaluated the approach in an informal user study, where users where asked to conduct an information analysis task with data spread across multiple windows. User feedback on the visual links themselves was positive throughout. The method of triggering the visual links and interaction with the employed views were occasionally criticized. For details on the technique and the study refer to the original paper [197] or to the thesis of Manuela Waldner [195]. Waldner et al. have since demonstrated that visual links can also be used in multi-display scenarios for co-located, collaborative visual analytics [198].

While visual links across applications are a very salient highlighting technique, they nevertheless occlude information, especially when many items are connected. How this can be remedied is the topic of the next section.

8.2.2 Context-Preserving Visual Links

As visual links are rendered on top of a base representation, occlusion of the information in the base representation is unavoidable. In this section, we discuss how this occlusion can be minimized so that as much of the important parts of a base representation as possible remain visible. To this end we formulate optimization criteria for visual links:

1. Visual links should have a minimal length.
2. The amount of information occluded by visual links should be minimal.

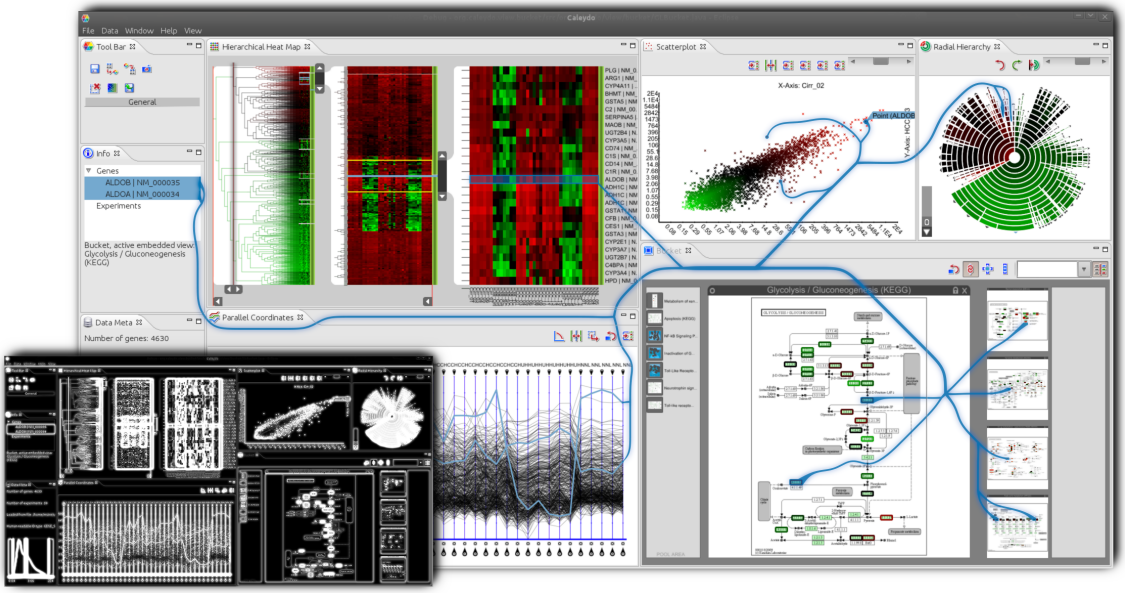


Figure 8.3: Context-preserving visual links. Based on a salience measure, shown in the small black-and-white inset, and other penalties we compute link paths that occlude minimal amounts of salient information, while taking the other optimization criteria into account. This example shows five of Caleydo's visualization techniques, where highlighted elements are connected with context-preserving visual links. Notice how the links choose routes along the view boundaries.

3. Visual links should be easy to distinguish from the base representation, they should stand out.

4. Links in close proximity should be bundled to decrease the overall link length.

Our approach is based on a penalty map, where four factors are considered. The first is an *importance map*, where salient regions of the base representation are given a high penalty. We calculate the saliency of a scene using a model of saliency-based visual attention [84]. A sample result of the saliency calculation for the scene depicted in Figure 8.3 is shown in the small inset in the same figure. An additional penalty is assigned to regions where the *color in the base representation is similar to the chosen link color*, to *regions where highlights are present* and to *other visual links* in case of multiple simultaneous sets of visual links. The routing is defined as an optimization problem, in which the link length is weighted against the accumulated penalties. For details on the process and on the optimization as well as on the necessary discretization refer to the original paper [174]. Figure 8.3 shows the results of the technique applied to a complex visual analysis scenario in Caleydo. It is clearly visible that the links avoid salient regions, for example, by routing around the dense regions in the scatterplots at the top right. When a segment bundles many branches, it is shown thicker. Halos help to distinguish the links from the base representation.

8.2.3 Study on Effectiveness of Visual Links

In addition to the informal study on the utility of interactive visual links across applications, we conducted a formal, quantitative user-study to evaluate the effectiveness of visual links compared to traditional, color-based highlighting. The study included three conditions: **color-based highlighting**, traditional (straight and bundled) **visual links**, as well as **context-preserving visual links**. Examples of each condition are shown in the top row of Figure 8.4. The study was conducted as a within-subjects experiments with 18 participants and 16 repetitions for each condition. The task in each condition was to count the number of highlights in a scene, which varied from five to twelve. We used the same set of base representations for all conditions, but varied the highlighted regions to avoid learning effects. Subjects were timed and accuracy of the results was recorded. Additionally, an SMI* RED 200 stationary eye-tracker was used to record the participants' gaze path. After each condition, participants were asked to answer a set of questions to record subjective assessment. We postulated the following hypothesis:

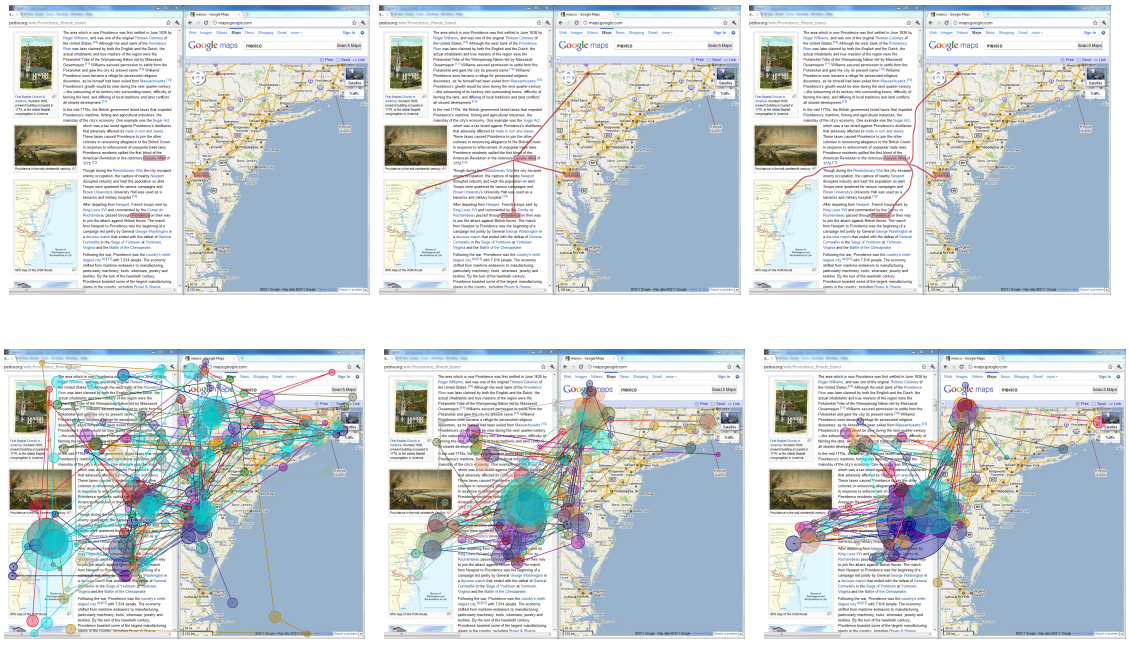
1. Visual links lead to a better performance than conventional highlights.

2. Context-preserving visual links do not have a negative impact on performance.

3. Context-preserving visual links have a positive impact on user satisfaction.

We found a significant main effect for the performance, where post-hoc comparisons revealed that completion time was significantly higher with highlighting ($t_{highlights}$ =

*http://www.smivision.com/



(a) (b) (c)

Figure 8.4: Conditions and eye tracking results of the visual links user study. The top row shows the three different conditions of the study: (a) color-based highlighting, (b) traditional visual links, and (c) context-preserving visual links. The lower row shows the gaze plots for the respective images. While the gazes for the highlighting method are spread over the whole image, those for the linking techniques are much more focused.

$4897ms$) compared to either linking technique ($t_{links} = 4176ms$, $t_{context-preserving} = 4024ms$). Subjective speed and subjective correctness were rated significantly higher for both linking techniques. Actual correctness of the results was high in all conditions. Participants rated context-preserving visual links to be more visually pleasing than the other two techniques, also the usefulness of both liking techniques was rated higher (both differences being statistically significant). Consequently, we can conclude that Hypothesis 1 and 2 were fully supported, and Hypothesis 3 was partially supported. The latter hypothesis – *context-preserving links have a positive impact on user satisfaction* – was supported by the significant effects for attractiveness, but not for the overall preference, or subjective mental demand.

The lower row in Figure 8.4 shows a gaze-plot of eye tracking data for the three conditions. We observe that gaze-paths were more focused in the scenarios employing visual links, while the first plot, for the traditional highlighting, shows a more distributed pattern. We speculate that this is due to a "serial search" approach for the traditional highlights, while visual links make users follow the links with their gaze. For a thorough discussion of the study, including additional results, the employed statistical methodology as well as measures for significances and effect sizes refer to the original paper [174].

8.3 Conclusion

In this chapter, we have discussed two classes of multiple heterogeneous datasets: Structured, heterogeneous datasets, where a data-model for dataset dependencies is available, or unstructured datasets, where no prior knowledge on the relationships of the datasets exists. For the former class, modeling of the data, the views, the analytic interfaces and the tasks can be used in a visual analysis system supporting orientation, or even guidance. Visual linking can be used in both the structured and the unstructured case to show relationships among datasets, applications, and views. Finally, we have shown how visual links can be routed to occlude a minimal amount of information and demonstrated the utility of visual links.

Even with the methods proposed, heterogeneous data analysis is still challenging. The ideas presented here are but first steps of a long path ahead. We believe that convenient analysis of heterogeneous data will not only be achieved by creating new visual encodings. Instead, it might have to be approached from an engineering angle: the tools for individual datasets are here, and they are ready to use, but there is no communication, no connection among them. Visualization research will play its part, for example by providing the glue between those applications through meta-visualization, but software compatibility, connectivity and standards are needed for a successful integrative heterogeneous data analysis.

Chapter 9

Conclusion and Outlook

Visualization of multidimensional data may well be one of the most researched sub-domains of Information Visualization. However, with the widespread adoption of biomolecular data analysis, new challenges of significant practical relevance have arisen: making sense of the vast amounts of data generated can have a profound impact on mankind's understanding of biomolecular processes, often with direct implications for treatment and prognosis of patients. Also, we observe an adoption of biomolecular methods in clinical scenarios. While in the past, due to the high cost of data generation, biomolecular analysis was restricted to research, it is now being used in clinical settings for diagnostic purposes. Furthermore, due to next-generation sequencing, it is now possible to capture data from a wide variety of biomolecular processes and of biomolecular properties. While ten years ago, gene expression data was the only widely available multidimensional data source in molecular biology, it is now complemented with miRNA expression, DNA methylation as well as data on structural variation, such as SNPs, small-scale mutations, and copy number variations, to name just a few. While this makes it possible to analyze relationships and interdependencies and uncover the causes for many biological phenomena, it also poses challenges for the analysis. Hence, analysis tools have to keep up with the following important developments:

1. Analysis tools need to become more user-friendly. Traditional approaches, for example using R-scripts to analyze and plot the data, are not likely to be adopted in clinical settings. They are also a considerable hurdle in the analysis by domain experts without profound knowledge in bioinformatics methods.

2. Visual analysis methods need to scale to vast amounts of data.

3. A wide array of datasets need to be integrated for a comprehensive analysis.

In this thesis, all these points were explored: We proposed interactive visual analysis methods, combined with analytical algorithms to make analysis of complex interdependencies possible, without the help of scripting. We demonstrated how the separation into homogeneous sub-groups can improve scalability in terms of the amount of conditions that can be reasonably analyzed concurrently. We also explained how limited screen-space can be optimally used by employing the divide and conquer approach in combination with the

multiform technique, allowing us to show abstractions for out-of-focus or very homogeneous regions of datasets. At the same time, we demonstrated how interaction combined with the multiform approach can provide advanced focus and context, as well as drill-down techniques. Finally, we established how a variety of multidimensional, cross-referenced datasets can be integrated in a comprehensive analysis scenario.

The main contribution of this thesis is a new visual analysis technique for individual multidimensional and multiple cross-referenced datasets following the paradigm:

Stratify the dataset(s) into homogeneous subsets, show each subset using the best visualization technique available, and visualize the relationships among them.

In line with this main contribution, we discussed a broad variety of considerations, including how to best encode the relationships between these subsets, how to interact with the subsets, how to choose which datasets or subset to show, just to name the most important.

As the discussed techniques cannot be meaningfully evaluated in laboratory settings using random participants and easily measurable tasks, we chose to evaluate our approaches using case studies. Case studies are an established evaluation methodology and suitable "to assess a visualization tool's ability to support visual analysis and reasoning about data"[109]. We conducted our case studies with various experts from different institutions, all of them full-time researchers in molecular biology. Some of them were involved in the user-centered design process, while others were not familiar with the software. The results of the case studies affirm our claim that the proposed methods have a clear benefit in these application scenarios.

We also elaborated on how heterogeneous data can be visualized in an integrative scenario. We distinguished two cases: structured heterogeneous data, where the relationships of the datasets are known beforehand, and unstructured heterogeneous data with no defined relationships. We discussed how, for the former case, modeling of data, views, and analysis interfaces can be used to provide orientation and guidance for the user. While the latter case is not in the scope of this thesis, we nevertheless discussed visual linking as a method usable for general heterogeneous data analysis scenarios, irrespective of their structure. Context-preserving visual links were introduced as a method that can link related items with minimal impact on the legibility of the base representation. We formally evaluated visual links and found them to improve performance in detecting multiple highlights and to be aesthetically pleasing.

9.1 Outlook

In the future, a number of technical improvements for Caleydo are desirable. Examples are a fusion of the features of the hierarchical heat map and the VisBricks technology or out-of-core strategies, to be able to handle datasets of a size beyond what can be loaded into memory.

Conceptually, it would be worth researching, whether and how the restrictions on

the alignment of bricks of the same dimension group, which are currently always stacked on top of each other, can be relieved. If this were possible, it would allow to better analyze relationships across multiple dimension groups. The challenge here is the loss of relationships between dimensions. While in theory it would be possible to use visual links, for example in the form of *Bubble Sets* [31], to encode a brick's membership of a dimension group, it remains to be seen whether this is acceptable for users.

VisBricks and StratomeX currently leave it up to the user to decide, which visualization technique is most suitable for a given brick. Experiments with automatic adaption of views, for example by switching them to compact cluster bricks when not enough space is available, received mixed feedback from users. Nevertheless, we believe that an automatic, task-based adaption of the visualization technique and the level of detail of a brick can improve scalability and increase insight.

A challenge for future research is to increase the number of bricks that can conveniently be displayed in a column. While due to the abstraction capabilities of the multiform technique, a very large number of records can be summarized, the number of bricks is limited. One of the reasons for this is the that the number of ribbons is a quadratic function of the number of bricks, assuming that most bricks are connected. This causes many crossings and clutter. We discussed that only showing selected ribbons is a remedy – at the cost of a worse overview. Another strategy would be to algorithmically minimize the crossings between the bricks. Asides from the clutter caused by many ribbons, the space available for individual bricks is reduced when many bricks are to be shown. This can make culling of some bricks necessary. A possible approach to overcome this would be to introduce a hierarchy of bricks, meaning that on an overview level many bricks could be abstracted into a single brick. Only when drilling down are the other bricks shown.

A tighter integration of statistical analysis, for example to quantify the relationships of two bricks, is desirable. The rationale for this is that while the visualization can help to discover a trend in the data, statistics are nevertheless needed to be able to confirm and report a discovery. Integrating statistical analysis to quantify observed effects reduces the required effort on the user's side and will likely also increase the trust in the visualization system.

Bibliography

[1] J. Abello, F. Ham, and N. Krishnan. ASK-GraphView: a large scale graph visualization system. *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 669–676, 2006. doi:10.1109/TVCG.2006.120. Cited on page 27.

[2] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell*. Garland Science, fifth edn., 2007. ISBN 0815341059. Cited on pages 13 and 14.

[3] M. Ankerst, D. A. Keim, and H. Kriegel. Circle segments: A technique for visually exploring large multidimensional data sets. In *Proceedings of the IEEE Conference on Visualization (Vis '96), Hot Topic Session*. San Francisco, CA, 1996. Cited on page 25.

[4] F. J. Anscombe. Graphs in statistical analysis. *The American Statistician*, vol. 27, no. 1, pp. 17–21, 1973. doi:10.2307/2682899. Cited on page 22.

[5] D. A. Aoyama, J. T. Hsiao, A. F. Cárdenas, and R. K. Pon. TimeLine and visualization of multiple-data sets and the visualization querying challenge. *Journal of Visual Languages & Computing*, vol. 18, no. 1, p. 1–21, 2007. doi:10.1016/j.jvlc.2005.11.002. Cited on page 40.

[6] D. Archambault, T. Munzner, and D. Auber. TopoLayout: multilevel graph layout by topological features. *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 2, pp. 305–317, 2007. doi:10.1109/TVCG.2007.46. Cited on page 28.

[7] D. Asimov. The grand tour: a tool for viewing multidimensional data. *SIAM Journal of Scientific and Statistical Computing*, vol. 6, no. 1, pp. 128–143, 1985. doi:10.1137/0906011. Cited on page 23.

[8] 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. Cited on page 49.

[9] 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. Cited on page 43.

[10] F. Battke, S. Symons, and K. Nieselt. Mayday - integrative analytics for expression data. *BMC Bioinformatics*, vol. 11, no. 1, p. 121, 2010. doi:10.1186/1471-2105-11-121. Cited on page 41.

[11] 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. Cited on page 39.

[12] F. Bendix, R. Kosara, and H. Hauser. Parallel sets: visual analysis of categorical data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '05)*, pp. 133– 140. IEEE, 2005. ISBN 0-7803-9464-X. doi:10.1109/INFVIS.2005.1532139. Cited on page 38.

[13] R. D. Bergeron, W. Cody, W. Hibbard, D. T. Kao, K. D. Miceli, L. A. Treinish, and S. Walther. Database issues for data visualization: Developing a data model. In J. P. Lee and G. G. Grinstein (Editors), *Database Issues for Data Visualization*, vol. 871, pp. 1–15. Springer-Verlag, Berlin/Heidelberg, 1994. ISBN 3-540-58519-2. doi:10.1007/BFb0021141. Cited on page 4.

[14] J. Bertin. *Graphics and graphic information-processing*. de Gruyter, 1981. ISBN 9783110088687. Cited on page 21.

[15] J. Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. ESRI Press, first published in french in 1967 edn., 2010. ISBN 9781589482616. Cited on pages 2, 22, and 63.

[16] C. Beshers and S. Feiner. AutoVisual: rule-based design of interactive multivariate visualizations. *IEEE Computer Graphics and Applications*, vol. 13, no. 4, pp. 41–49, 1993. doi:10.1109/38.219450. Cited on page 26.

[17] A. Beygelzimer, C. Perng, and S. Ma. Fast ordering of large categorical datasets for better visualization. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '01)*, pp. 239–244. ACM, 2001. ISBN 158113391X. doi:10.1145/502512.502545. Cited on page 36.

[18] M. Borkin, K. Gajos, A. Peters, D. Mitsouras, S. Melchionna, F. Rybicki, C. Feldman, and H. Pfister. Evaluation of artery visualizations for heart disease diagnosis. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '11)*, vol. 17, no. 12, pp. 2479–2488, 2011. doi:10.1109/TVCG.2011.192. Cited on page 25.

[19] D. Borland and R. M. Taylor II. Rainbow color map (Still) considered harmful. *IEEE Computer Graphics and Applications*, vol. 27, no. 2, pp. 14–17, 2007. Cited on page 25.

[20] C. A. Brewer. Colorbrewer. http://colorbrewer2.org/, 2009. Last accessed Dez. 09, 2011. Cited on page 71.

[21] Y. Cai, X. Yu, S. Hu, and J. Yu. A brief review on the mechanisms of miRNA regulation. *Genomics, Proteomics & Bioinformatics*, vol. 7, no. 4, pp. 147–154, 2009. doi:10.1016/S1672-0229(08)60044-3. Cited on page 15.

[22] 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. Cited on page 23.

[23] J. M. Chambers. *Graphical methods for data analysis*. Wadsworth International Group, 1983. ISBN 9780534980528. Cited on page 23.

[24] S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. *ACM SIGMOD Record*, vol. 26, no. 1, p. 65–74, 1997. doi:10.1145/248603.248616. Cited on page 27.

[25] L. Chin, J. N. Andersen, and P. A. Futreal. Cancer genomics: from discovery science to personalized medicine. *Nature Medicine*, vol. 17, no. 3, pp. 297–303, 2011. doi:10.1038/nm.2323. Cited on pages 16 and 18.

[26] L. Chin, W. C. Hahn, G. Getz, and M. Meyerson. Making sense of cancer genomic data. *Genes & Development*, vol. 25, no. 6, pp. 534–555, 2011. doi:10.1101/gad.2017311. Cited on page 18.

[27] J. H. Claessen and J. J. van Wijk. Flexible linked axes for multivariate data visualization. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '11)*, vol. 17, no. 12, pp. 2310–2316, 2011. doi:10.1109/TVCG.2011.201. Cited on page 24.

[28] M. Clamp, D. Andrews, D. Barker, P. Bevan, G. Cameron, Y. Chen, L. Clark, T. Cox, J. Cuff, V. Curwen, T. Down, R. Durbin, E. Eyras, J. Gilbert, M. Hammond, T. Hubbard, A. Kasprzyk, D. Keefe, H. Lehvaslaiho, V. Iyer, C. Melsopp, E. Mongin, R. Pettett, S. Potter, A. Rust, E. Schmidt, S. Searle, G. Slater, J. Smith, W. Spooner, A. Stabenau, J. Stalker, E. Stupka, A. Ureta-Vidal, I. Vastrik, and E. Birney. Ensembl 2002: accommodating comparative genomics. *Nucleic Acids Research*, vol. 31, no. 1, pp. 38 –42, 2003. doi:10.1093/nar/gkg083. Cited on page 44.

[29] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Computing Surveys (CSUR)*, vol. 41, no. 1, pp. 1–31, 2008. doi:10.1145/1456650.1456652. Cited on page 29.

[30] C. Collins and S. Carpendale. VisLink: revealing relationships amongst visualizations. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '07)*, vol. 13, no. 6, pp. 1192–1199, 2007. doi:10.1109/TVCG.2007.70521. Cited on pages 34 and 35.

[31] C. Collins, G. Penn, and S. Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and*

Computer Graphics (InfoVis '09), vol. 15, no. 6, pp. 1009–1016, 2009. doi:10.1109/TVCG.2009.122. Cited on pages 34, 35, and 125.

[32] 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. Cited on page 41.

[33] K. Dinkla, M. A. Westenberg, H. M. Timmerman, S. A. van Hijum, and J. J. van Wijk. Comparison of multiple weighted hierarchies: Visual analytics for microbe community profiling. *Computer Graphics Forum (EuroVis '11)*, vol. 30, no. 3, pp. 1141–1150, 2011. doi:10.1111/j.1467-8659.2011.01963.x. Cited on page 76.

[34] A. Dix and G. Ellis. By chance: enhancing interaction with large data sets through statistical sampling. In *Proceedings of the ACM Conference on Advanced Visual Interfaces (AVI '02)*, pp. 167–176. ACM Press, 2002. ISBN 1581135378. Cited on pages 24, 79, and 85.

[35] H. Doleisch. SIMVIS: interactive visual analysis of large and time-dependent 3D simulation data. In *Proceedings of the Conference on Winter Simulation (WSC '07)*, p. 712–720. IEEE Press, Piscataway, NJ, USA, 2007. ISBN 1-4244-1306-0. Cited on pages 32 and 39.

[36] S. dos Santos and K. Brodlie. Gaining understanding of multivariate and multidimensional data through visualization. *Computers & Graphics*, vol. 28, no. 3, pp. 311–325, 2004. doi:10.1016/j.cag.2004.03.013. Cited on page 51.

[37] C. Eaton, C. Plaisant, and T. Drizd. Visualizing missing data: Graph interpretation user study. In *Proceedings of the Conference on Human-Computer Interaction (INTERACT '05)*, vol. 3585 of *Lecture Notes in Computer Science (LNCS)*, pp. 861–872. Springer, 2005. doi:10.1007/11555261_68. Cited on page 6.

[38] 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. Cited on pages 25, 26, 27, 41, 62, 79, 86, and 90.

[39] 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. Cited on page 24.

[40] N. Elmqvist, P. Dragicevic, and J. D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '08)*, vol. 14, no. 6, pp. 1539–1148, 2008. doi:10.1109/TVCG.2008.153. Cited on page 23.

[41] N. Elmqvist and J. Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 3, pp. 439–454, 2010. doi:10.1109/TVCG.2009.84. Cited on page 79.

[42] S. K. Feiner and C. Beshers. Visualizing n-dimensional virtual worlds with n-vision. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '90)*, vol. 24, p. 37–38, 1990. doi:10.1145/91394.91412. Cited on page 26.

[43] J. D. Fekete and C. Plaisant. Interactive information visualization of a million items. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '02)*, pp. 117–124, 2002. doi:10.1109/INFVIS.2002.1173156. Cited on page 22.

[44] M. C. Ferreira de Oliveira and H. Levkowitz. From visual data exploration to visual data mining: A survey. *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 3, pp. 378–394, 2003. doi:http://doi.ieeecomputersociety.org/10.1109/TVCG.2003.1207445. Cited on pages 22, 26, and 79.

[45] J. L. Freeman, G. H. Perry, L. Feuk, R. Redon, S. A. McCarroll, D. M. Altshuler, H. Aburatani, K. W. Jones, C. Tyler-Smith, M. E. Hurles, N. P. Carter, S. W. Scherer, and C. Lee. Copy number variation: New insights in genome diversity. *Genome Research*, vol. 16, no. 8, pp. 949 –961, 2006. doi:10.1101/gr.3677206. Cited on pages 15 and 16.

[46] 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. Cited on pages 27 and 62.

[47] M. Friendly. Extending mosaic displays: Marginal, conditional, and partial views of categorical data. *Journal of Computational and Graphical Statistics*, vol. 8, no. 3, pp. 373–395, 1999. Cited on page 37.

[48] M. Friendly and E. Kwan. Effect ordering for data displays. *Computational Statistics & Data Analysis*, vol. 43, no. 4, pp. 509–539, 2003. doi:10.1016/S0167-9473(02)00290-6. Cited on page 36.

[49] Y. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of the IEEE Conference on Visualization (Vis '99)*, p. 43–50. IEEE Computer Society Press, Los Alamitos, CA, USA, 1999. ISBN 0-7803-5897. doi:10.1109/VISUAL.1999.809866. Cited on page 24.

[50] Y. Fua, M. O. Ward, and E. A. Rundensteiner. Structure-based brushes: a mechanism for navigating hierarchically organized data and information spaces. *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 2, pp. 150–159, 2000. doi:10.1109/2945.856996. Cited on page 24.

[51] G. W. Furnas. Generalized fisheye views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '86)*, pp. 16–23. ACM Press, New York, NY, USA, 1986. doi:10.1145/22339.22342. Cited on page 22.

[52] E. R. Gansner, Y. Hu, S. North, and S. Carlos. Multilevel agglomerative edge bundling for visualizing large graphs. In *Proceedings of the IEEE Symposium on*

Pacific Visualization (PacificVis '11). IEEE Computer Society Press, 2011. doi:10.1109/PACIFICVIS.2011.5742389. Cited on page 34.

[53] K. R. Gegenfurtner and L. T. Sharpe. *Color vision: from genes to perception*. Cambridge University Press, 2001. ISBN 9780521004398. Cited on page 25.

[54] N. Gehlenborg, J. Dietzsch, and K. Nieselt. A framework for visualization of microarray data and integrated meta information. *Information Visualization*, vol. 4, no. 3, pp. 164–175, 2005. doi:10.1057/palgrave.ivs.9500094. Cited on page 41.

[55] 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. Cited on pages 17, 18, 42, and 43.

[56] M. B. Gerstein, C. Bruce, J. S. Rozowsky, D. Zheng, J. Du, J. O. Korbel, O. Emanuelsson, Z. D. Zhang, S. Weissman, and M. Snyder. What is a gene, post-ENCODE? history and updated definition. *Genome Research*, vol. 17, no. 6, pp. 669 –681, 2007. doi:10.1101/gr.6339607. Cited on page 14.

[57] T. Geymayer, A. Lex, M. Streit, and D. Schmalstieg. Visualizing the effects of logically combined filters. In *Proceedings of the International Conference on Information Visualisation (IV'11)*, pp. 47–52. IEEE, London, UK, 2011. ISBN 978-1-4577-0868-8. doi:10.1109/IV.2011.52. Cited on pages 9, 11, 52, and 53.

[58] A. D. Goldberg, C. D. Allis, and E. Bernstein. Epigenetics: A landscape takes shape. *Cell*, vol. 128, no. 4, pp. 635 – 638, 2007. doi:10.1016/j.cell.2007.02.006. Cited on page 15.

[59] M. Graham and J. Kennedy. Combining linking & focusing techniques for a multiple hierarchy visualisation. In *Proceedings of the IEEE Symposium on Information Visualisation (InfoVis '01)*, p. 425. IEEE Computer Society, 2001. ISBN 0-7695-1195-3. doi:10.1109/IV.2001.942092. Cited on page 31.

[60] M. Greenacre. *Correspondence Analysis in Practice*. Chapman & Hall/CRC, second edn., 2007. ISBN 1584886161. Cited on page 36.

[61] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009. doi:10.1145/1656274.1656278. Cited on pages 41 and 62.

[62] S. Han, G. W. Humphreys, and L. Chen. Uniform connectedness and classical gestalt principles of perceptual grouping. *Perception & Psychophysics*, vol. 61, no. 4, pp. 661–674, 1999. doi:10.3758/BF03205537. Cited on page 33.

[63] S. Hanada, P. Strnad, E. M. Brunt, and M. B. Omary. The genetic background modulates susceptibility to mouse liver Mallory-Denk body formation and liver injury. *Hepatology (Baltimore, Md.)*, vol. 48, no. 3, pp. 943–952, 2008. doi:10.1002/hep.22436. Cited on page 71.

[64] 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. Cited on pages 24 and 32.

[65] S. L. Havre, A. Shah, C. Posse, and B. Webb-Robertson. Diverse information integration and visualization. In *Proceedings of the SPIE Conference on Visualization and Data Analysis (VDA '06)*, vol. 6060, pp. 60600M–60600M–11. SPIE, 2006. ISBN 0819461008. doi:10.1117/12.643492. Cited on page 36.

[66] 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. Cited on pages 31 and 82.

[67] 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. Cited on page 54.

[68] 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. Cited on pages 39 and 40.

[69] J. Heer and G. G. Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1240–1247, 2007. doi:10.1109/TVCG.2007.70539. Cited on page 49.

[70] M. J. Heller. DNA microarray technology: Devices, systems, and applications. *Annual Review of Biomedical Engineering*, vol. 4, no. 1, pp. 129–153, 2002. doi:10.1146/annurev.bioeng.4.020702.153438. Cited on page 16.

[71] N. Henry, J. D. Fekete, and M. J. McGuffin. NodeTrix: a hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '07)*, vol. 13, no. 6, pp. 1302–1309, 2007. doi:10.1109/TVCG.2007.70582. Cited on pages 28 and 29.

[72] 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. Cited on page 32.

[73] H. Hofmann. Exploring categorical data: interactive mosaic plots. *Metrika*, vol. 51, no. 1, pp. 11–26, 2000. doi:10.1007/s001840000041. Cited on page 36.

[74] H. Hofmann. Mosaic plots and their variants. In C.-h. Chen, W. Haerdle, and A. Unwin (Editors), *Handbook of Data Visualization*, pp. 617–642. Springer, 2008. ISBN 978-3-540-33036-3, 978-3-540-33037-0. doi:10.1007/978-3-540-33037-0_24. Cited on page 37.

[75] 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. Cited on page 34.

[76] D. Holten and J. van Wijk. Force-Directed edge bundling for graph visualization. *Computer Graphics Forum (EuroVis '09)*, vol. 28, no. 3, pp. 983–990, 2009. doi:10.1111/j.1467-8659.2009.01450.x. Cited on page 34.

[77] D. Holten and J. J. v. Wijk. Visual comparison of hierarchically organized data. *Computer Graphics Forum (EuroVis '08)*, vol. 27, pp. 759–766, 2008. doi:doi:10.1111/j.1467-8659.2008.01205.x. Cited on pages 34 and 65.

[78] C. Holzhüter, A. Lex, D. Schmalstieg, H. Schulz, H. Schumann, and M. Streit. Visualizing uncertainty in biological expression data. In *Proceedings of the SPIE Conference on Visualization and Data Analysis (VDA '12)*, vol. 8294, p. 82940O. IS&T/SPIE, 2012. doi:doi:10.1117/12.908516. Cited on pages 9, 12, 50, and 53.

[79] Z. Hu, J. Hung, Y. Wang, Y. Chang, C. Huang, M. Huyck, and C. DeLisi. VisANT 3.5: multi-scale network visualization, analysis and inference based on the gene ontology. *Nucleic Acids Research*, vol. 37, no. Web Server, pp. W115–W121, 2009. doi:10.1093/nar/gkp406. Cited on page 44.

[80] L. Hunter. *The processes of life: an introduction to molecular biology*. MIT Press, 2009. ISBN 9780262013055. Cited on page 14.

[81] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, vol. 1, no. 4, pp. 69–91, 1985. doi:10.1109/TVCG.2005.2. Cited on pages 23 and 63.

[82] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the IEEE Conference on Visualization (Vis '90)*, pp. 361–378. San Francisco, CA, USA, 1990. doi:10.1109/VISUAL.1990.146402. Cited on page 23.

[83] International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, vol. 409, no. 6822, pp. 860–921, 2001. doi:10.1038/35057062. Cited on page 17.

[84] L. Itti, C. Koch, and E. Niebur. A model of Saliency-Based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1254–1259, 1998. doi:10.1109/34.730558. Cited on page 119.

[85] S. Johansson and J. Johansson. Interactive dimensionality reduction through user-defined combinations of quality metrics. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '09)*, vol. 15, no. 6, pp. 993–1000, 2009. doi:10.1109/TVCG.2009.153. Cited on page 24.

[86] S. Johansson Fernstad and J. Johansson. A task based performance evaluation of visualization approaches for categorical data analysis. In *Proceedings of the Conference on Information Visualisation (IV '11)*, pp. 80–89. IEEE, 2011. ISBN 978-1-4577-0868-8. doi:10.1109/IV.2011.92. Cited on page 35.

[87] 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. Cited on page 51.

[88] I. Jolliffe. *Principal Component Analysis*. Springer, second edn., 2002. ISBN 0387954422. Cited on page 24.

[89] B. H. Junker, C. Klukas, and F. Schreiber. VANTED: a system for advanced data analysis and visualization in the context of biological networks. *BMC Bioinformatics*, vol. 7, no. 1, p. 109, 2006. doi:10.1186/1471-2105-7-109. Cited on page 44.

[90] A. D. Kalvin, B. E. Rogowitz, A. Pelah, and A. Cohen. Building perceptual color maps for visualizing interval data. *Proceedings of SPIE*, vol. 3959, no. 1, pp. 323–335, 2000. doi:doi:10.1117/12.387169. Cited on page 25.

[91] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu, and Y. Yamanishi. KEGG for linking genomes to life and the environment. *Nucleic Acids Research*, vol. 36, no. Database-Issue, pp. 480–484, 2008. Cited on pages 44 and 109.

[92] H. Kang, L. Getoor, B. Shneiderman, M. Bilgic, and L. Licamele. Interactive entity resolution in relational data: A visual analytic tool and its evaluation. *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 5, pp. 999–1014, 2008. doi:10.1109/TVCG.2008.55. Cited on page 38.

[93] P. D. Karp, S. Paley, and P. Romero. The pathway tools software. *Bioinformatics*, vol. 18, no. Suppl 1, pp. S225–S232, 2002. doi:10.1093/bioinformatics/18.suppl_1.S225. Cited on page 44.

[94] 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. Cited on page 71.

[95] D. A. Keim. Designing pixel-oriented visualization techniques: theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 1, pp. 59–78, 2000. doi:10.1109/2945.841121. Cited on pages 22, 24, and 25.

[96] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 1–8, 2002. doi:10.1109/2945.981847. Cited on page 22.

[97] D. A. Keim, M. Ankerst, and H. Kriegel. Recursive pattern: A technique for visualizing very large amounts of data. In *Proceedings of the IEEE Conference on Visualization (Vis '95)*, VIS '95, p. 279–. IEEE Computer Society, Washington, DC, USA, 1995. ISBN 0-8186-7187-4. Cited on page 25.

[98] D. A. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann (Editors). *Mastering The Information Age - Solving Problems with Visual Analytics*. Eurographics, 2010. Cited on pages 2, 6, and 113.

[99] 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. Cited on pages 22 and 78.

[100] A. Khan, J. Matejka, G. Fitzmaurice, and G. Kurtenbach. Spotlight: directing users' attention on large displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*, p. 791–798. ACM Press, 2005. ISBN 1581139985. doi:10.1145/1054972.1055082. Cited on page 32.

[101] B. Kim, B. Lee, S. Knoblach, E. Hoffman, and J. Seo. GeneShelf: a web-based visual interface for large gene expression Time-Series data repositories. *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 905–912, 2009. doi:10.1109/TVCG.2009.146. Cited on pages 24 and 41.

[102] 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. Cited on page 39.

[103] R. Kosara. Turning a table into a tree: Growing parallel sets into a purposeful project. In J. Steele and N. Iliinsky (Editors), *Beautiful Visualization: Looking at Data through the Eyes of Experts*, pp. 193–204. O'Reilly, 2010. ISBN 1449379869. Cited on page 38.

[104] 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. Cited on pages 29, 30, 38, 81, 93, and 103.

[105] R. Kosara, S. Miksch, and H. Hauser. Semantic depth of field. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '01)*, pp. 97–104. IEEE Computer Society Press, 2001. ISBN 1522-404X. doi:10.1109/INFVIS.2001.963286. Cited on page 32.

[106] R. Kosara, S. Miksch, and H. Hauser. Focus+Context taken literally. *IEEE Computer Graphics and Applications*, vol. 22, no. 1, pp. 22–29, 2002. Doi:10.1109/38.974515. Cited on pages 32 and 33.

[107] 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. Cited on pages 34 and 44.

[108] C. Lackner, M. Gogg-Kamerer, K. Zatloukal, C. Stumptner, E. M. Brunt, and H. Denk. Ballooned hepatocytes in steatohepatitis: the value of keratin immunohistochemistry for diagnosis. *Journal of Hepatology*, vol. 48, no. 5, pp. 821–828, 2008. doi:10.1016/j.jhep.2008.01.026. Cited on page 71.

[109] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Empirical studies in information visualization: Seven scenarios. *IEEE Transactions on Visualization and Computer Graphics*, , no. 99, p. 1–1, 2011. Cited on page 124.

[110] J. LeBlanc, M. O. Ward, and N. Wittels. Exploring n-dimensional databases. In *Proceedings of the IEEE Conference on Visualization (Vis '90)*, pp. 230–237. IEEE, 1990. ISBN 0-8186-2083-8. doi:10.1109/VISUAL.1990.146386. Cited on page 26.

[111] A. Lex, P. J. Park, and N. Gehlenborg. Supporting subtype characterization through integrative visualization of cancer genomics data sets. In *Proceedings of The Cancer Genome Atlas' 1st Annual Scientific Symposium: Enabling Cancer Research Through TCGA*. Washington, D.C., USA, 2011. Cited on page 10.

[112] A. Lex, H. Schulz, M. Streit, C. Partl, and D. Schmalstieg. VisBricks: multiform visualization of large, inhomogeneous data. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '11)*, vol. 17, no. 12, pp. 2291–2300, 2011. doi:10.1109/TVCG.2011.250. Cited on pages 9 and 10.

[113] 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, 2010. ISBN 424466856. doi:10.1109/PACIFICVIS.2010.5429609. Cited on pages 9, 11, and 50.

[114] 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, 2010. doi:10.1109/TVCG.2010.138. Cited on pages 9, 10, and 76.

[115] A. Lex, M. Streit, H. Schulz, C. Partl, D. Schmalstieg, P. J. Park, and N. Gehlenborg. StratomeX: visual analysis of Large-Scale heterogeneous genomics data for cancer subtype characterization. *Conditionally accepted for: Computer Graphics Forum (EuroVis '12)*, vol. 31, no. 3, pp. fff–lll, 2012. Cited on pages 9 and 10.

[116] 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. Cited on page 40.

[117] 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. Cited on page 44.

[118] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129– 137, 1982. doi:10.1109/TIT.1982.1056489. Cited on page 27.

[119] T. V. Long and L. Linsen. MultiClusterTree: interactive visual exploration of hierarchical clusters in multidimensional multivariate data. *Computer Graphics Forum (EuroVis '09)*, vol. 28, no. 3, pp. 823–830, 2009. doi:10.1111/j.1467-8659.2009.01468.x. Cited on page 31.

[120] S. Ma and J. Hellerstein. Ordering categorical data to improve visualization. In *Proceedings of the IEEE Information Visualization Symposium (InfoVis '99) Late Breaking Hot Topics*. 15-18, 1999. Cited on page 36.

[121] A. MacEachren, D. Xiping, F. Hardisty, D. Guo, and G. Lengerich. Exploring high-D spaces with multiform matrices and small multiples. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '03)*, pp. 31–38. IEEE Computer Society Press, 2003. doi:10.1109/INFVIS.2003.1249006. Cited on pages 23, 28, and 30.

[122] S. Mansmann and M. H. Scholl. Exploring OLAP aggregates with hierarchical visualization techniques. In *Proceedings of the ACM Symposium on Applied Computing (SAC '07)*, pp. 1067–1073. ACM Press, 2007. doi:10.1145/1244002.1244235. Cited on pages 27, 28, and 30.

[123] 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. Cited on pages 29 and 51.

[124] M. I. McCarthy, G. R. Abecasis, L. R. Cardon, D. B. Goldstein, J. Little, J. P. A. Ioannidis, and J. N. Hirschhorn. Genome-wide association studies for complex traits: consensus, uncertainty and challenges. *Nature Reviews Genetics*, vol. 9, no. 5, pp. 356–369, 2008. doi:10.1038/nrg2344. Cited on page 18.

[125] M. Meyer, T. Munzner, A. DePace, and H. Pfister. MulteeSum: a tool for comparative spatial and temporal gene expression data. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '10)*, vol. 16, no. 6, pp. 908–917, 2010. doi:http://doi.ieeecomputersociety.org/10.1109/TVCG.2010.137. Cited on page 42.

[126] 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. Cited on pages 34 and 44.

[127] 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. Cited on pages 42, 43, 44, and 45.

[128] Y. Miki, J. Swensen, D. Shattuck-Eidens, P. Futreal, K. Harshman, S. Tavtigian, Q. Liu, C. Cochran, L. Bennett, W. Ding, and a. et. A strong candidate for the breast and ovarian cancer susceptibility gene BRCA1. *Science*, vol. 266, no. 5182, pp. 66 –71, 1994. doi:10.1126/science.7545954. Cited on page 18.

[129] K. Misue and T. Yuki. A visual analysis tool that smoothly switches between tabular forms and parallel coordinates. In *Proceedings of the International Symposium on Visual Information Communication*, VINCI '11, p. 3:1–3:7. ACM, Hong Kong, China, 2011. ISBN 978-1-4503-0786-4. doi:10.1145/2016656.2016659. Cited on page 76.

[130] 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. Cited on page 44.

[131] K. Moreland. Diverging color maps for scientific visualization. In *Advances in Visual Computing*, vol. 5876, pp. 92–103. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-10519-7, 978-3-642-10520-3. doi:10.1007/978-3-642-10520-3_9. Cited on page 25.

[132] 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. Cited on page 70.

[133] C. B. Nielsen, M. Cantor, I. Dubchak, D. Gordon, and T. Wang. Visualizing genomes: techniques and challenges. *Nature Methods*, vol. 7, no. 3s, pp. S5–S15, 2010. doi:10.1038/nmeth.1422. Cited on page 44.

[134] T. Nocke and H. Schumann. Meta data for visual data mining. In *Proceedings of the Conference on Computer Graphics and Imaging (CGIM '02)*, 2002. Cited on page 5.

[135] C. North, N. Conklin, K. Indukuri, V. Saini, and Q. Yu. Fusion: interactive coordination of diverse data, visualizations, and mining algorithms. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03) - extended abstracts*, pp. 626–627. ACM Press, 2003. ISBN 1581136374. doi:10.1145/765891.765897. Cited on page 40.

[136] C. North, N. Conklin, and V. Saini. Visualization schemas for flexible information visualization. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '02)*, pp. 15–22. IEEE, 2002. ISBN 1522-404X. doi:10.1109/INFVIS.2002.1173142. Cited on pages 39 and 99.

[137] 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, 2000. ISBN 1581132522. doi:10.1145/345513.345282. Cited on pages 40 and 116.

[138] H. Noushmehr, D. J. Weisenberger, K. Diefes, H. S. Phillips, K. Pujara, and et al. Identification of a CpG island methylator phenotype that defines a distinct subgroup of glioma. *Cancer Cell*, vol. 17, no. 5, pp. 510–522, 2010. doi:10.1016/j.ccr.2010.03.017. Cited on pages 97, 98, and 107.

[139] 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. Cited on pages 33 and 34.

[140] K. Pearson. Contributions to the mathematical theory of evolution. II. Skew variation in homogeneous material. *Philosophical Transactions of the Royal Society of London. A*, vol. 186, pp. 343–414, 1895. doi:10.1098/rsta.1895.0010. Cited on page 22.

[141] T. Pham, R. Hess, C. Ju, E. Zhang, and R. Metoyer. Visualization of diversity in large multivariate data sets. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '10)*, vol. 16, no. 6, pp. 1053–1062, 2010. doi:10.1109/TVCG.2010.216. Cited on page 5.

[142] 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. Cited on page 31.

[143] R Development Core Team. *R: A Language and Environment for Statistical Computing*, 2010. ISBN 3900051070. Cited on pages 41, 51, and 62.

[144] E. Ramos and D. Donoho. *The car dataset*, 1983. Available at: http://lib.stat.cmu.edu/datasets/. Cited on page 24.

[145] R. Rao and S. K. Card. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '94)*, p. 318–322. ACM, New York, NY, USA, 1994. ISBN 0-89791-650-6. doi:10.1145/191666.191776. Cited on pages 22 and 23.

[146] J. T. Rich, J. G. Neely, R. C. Paniello, C. C. J. Voelker, B. Nussenbaum, and E. W. Wang. A practical guide to understanding Kaplan-Meier curves. *Otolaryngology–Head and Neck Surgery*, vol. 143, no. 3, pp. 331–336, 2010. doi:10.1016/j.otohns.2010.05.007. Cited on page 104.

[147] P. Riehmann, M. Hanfler, and B. Froehlich. Interactive sankey diagrams. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '05)*, pp.

233–240. IEEE, 2005. ISBN 0-7803-9464-X. doi:10.1109/INFVIS.2005.1532152. Cited on page 38.

[148] J. C. Roberts. Multiple-View and multiform visualization. In *Proceedings of the SPIE Conference on Visualization and Data Analysis (VDA '02)*, vol. 3960, pp. 176–185, 2000. Cited on pages 28 and 77.

[149] 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. Cited on page 23.

[150] H. Rohn, C. Klukas, and F. Schreiber. Creating views on integrated multidomain data. *Bioinformatics*, vol. 27, no. 13, pp. 1839 –1845, 2011. doi:10.1093/bioinformatics/btr282. Cited on pages 40 and 99.

[151] G. E. Rosario, E. A. Rundensteiner, D. C. Brown, M. O. Ward, and S. Huang. Mapping nominal values to numbers for effective visualization. *Information Visualization*, vol. 3, no. 2, pp. 80–95, 2004. doi:10.1057/palgrave.ivs.9500072. Cited on page 36.

[152] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, vol. 20, no. 0, pp. 53–65, 1987. doi:10.1016/0377-0427(87)90125-7. Cited on page 61.

[153] S. Rufiange, M. J. McGuffin, and C. Fuhrman. Visualisation hybride des liens hiérarchiques incorporant des treemaps dans une matrice d'adjacence. In *Proceedings of the Conference on Association Francophone d'Interaction Homme-Machine (IHM '09)*, pp. 51–54, 2009. doi:10.1145/1629826.1629834. Cited on page 28.

[154] O. Rübel, G. H. Weber, M. Huang, E. W. Bethel, M. D. Biggin, C. C. Fowlkes, C. L. Luengo Hendriks, S. V. Keranen, M. B. Eisen, D. W. Knowles, J. Malik, H. Hagen, and B. Hamann. Integrating data clustering and visualization for the analysis of 3D gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 7, no. 1, pp. 64–79, 2010. doi:10.1109/TCBB.2008.49. Cited on page 42.

[155] O. Rübel, G. H. Weber, S. V. E. Keränen, C. C. Fowlkes, C. L. L. Hendriks, L. Simirenko, N. Y. Shah, M. B. Eisen, M. D. Biggin, H. Hagen, D. Sudar, J. Malik, D. W. Knowles, and B. Hamann. PointCloudXplore: visual analysis of 3D gene expression data using physical views and parallel coordinates. In *Proceedings of the Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis '06)*, pp. 203–210, 2006. doi:10.2312/VisSym/EuroVis06/203-210. Cited on pages 41, 42, and 45.

[156] A. I. Saeed, V. Sharov, J. White, J. Li, W. Liang, N. Bhagabati, J. Braisted, M. Klapa, T. Currier, M. Thiagarajan, A. Sturn, M. Snuffin, A. Rezantsev, D. Popov, A. Ryltsov, E. Kostukovich, I. Borisovsky, Z. Liu, A. Vinsavich, V. Trush, and

J. Quackenbush. TM4: a free, open-source system for microarray data management and analysis. *BioTechniques*, vol. 34, no. 2, pp. 374–378, 2003. Cited on page 41.

[157] 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. Cited on pages 22 and 69.

[158] H. Schulz, M. John, A. Unger, and H. Schumann. Visual analysis of bipartite biological networks. In *Proceedings of the Eurographics Workshop on Visual Computing for Biomedicine (VCBM '08)*, pp. 135–142. Eurographics, 2008. doi:10.2312/VCBM/VCBM08/135-142. Cited on page 81.

[159] 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. Cited on pages 29, 34, 41, 49, and 61.

[160] J. Seo and B. Shneiderman. A Rank-by-Feature framework for unsupervised multidimensional data exploration using low dimensional projections. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '04)*, pp. 65–72. IEEE Computer Society, 2004. ISBN 0-7803-8779-3. doi:10.1109/INFVIS.2004.3. Cited on page 45.

[161] J. Seo and B. Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, vol. 4, no. 2, p. 96–113, 2005. doi:10.1057/palgrave.ivs.9500091. Cited on page 41.

[162] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, vol. 13, no. 11, pp. 2498–2504, 2003. doi:10.1101/gr.1239303. Cited on page 43.

[163] J. Sharko, G. G. Grinstein, K. A. Marx, J. Zhou, C. Cheng, S. Odelberg, and H. Simon. Heat map visualizations allow comparison of multiple clustering results and evaluation of dataset quality: Application to microarray data. In *Proceedings of the International Conference Information Visualization (IV '07)*, pp. 521–526. IEEE Computer Society, 2007. ISBN 0-7695-2900-3. doi:10.1109/IV.2007.61. Cited on page 61.

[164] Z. Shen, J. Sun, Y. Shen, and M. Li. R-Map: mapping categorical data for clustering and visualization based on reference sets. In T. Washio, E. Suzuki, K. M. Ting, and A. Inokuchi (Editors), *Advances in Knowledge Discovery and Data Mining*, vol. 5012, pp. 992–998. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-68124-3, 978-3-540-68125-0. doi:10.1007/978-3-540-68125-0_104. Cited on page 36.

[165] J. Shendure and H. Ji. Next-generation DNA sequencing. *Nature Biotechnology*, vol. 26, no. 10, pp. 1135–1145, 2008. doi:10.1038/nbt1486. Cited on pages 17 and 18.

[166] J. A. Shendure, G. J. Porreca, G. M. Church, A. F. Gardner, C. L. Hendrickson, J. Kieleczawa, and B. E. Slatko. Overview of DNA sequencing strategies. In *Current Protocols in Molecular Biology*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2011. ISBN 0471142727, 9780471142720. doi:10.1002/0471142727.mb0701s96. Cited on page 18.

[167] B. T. Sherman, D. W. Huang, Q. Tan, Y. Guo, S. Bour, D. Liu, R. Stephens, M. W. Baseler, H. C. Lane, and R. A. Lempicki. DAVID knowledgebase: a gene-centered database integrating heterogeneous gene annotation resources to facilitate high-throughput gene functional analysis. *BMC Bioinformatics*, vol. 8, p. 426, 2007. doi:10.1186/1471-2105-8-426. Cited on page 56.

[168] 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. IEEE, 1996. ISBN 081867508X. doi:10.1109/VL.1996.545307. Cited on pages 51, 64, and 104.

[169] 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. Cited on pages 34 and 40.

[170] Y. B. Shrinivasan and J. J. v. 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. Cited on page 39.

[171] H. Siirtola, T. Laivo, T. Heimonen, and K. Räihä. Visual perception of parallel coordinate visualizations. In *Proceedings of the International Conference Information Visualization (IV '09)*, pp. 3–9. IEEE, 2009. ISBN 978-0-7695-3733-7. doi:10.1109/IV.2009.25. Cited on page 24.

[172] H. Siirtola and K. Räihä. Interacting with parallel coordinates. *Interacting with Computers*, vol. 18, no. 6, pp. 1278–1309, 2006. doi:10.1016/j.intcom.2006.03.006. Cited on page 24.

[173] 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. Cited on page 51.

[174] M. Steinberger, M. Waldner, M. Streit, A. Lex, and D. Schmalstieg. Context-Preserving visual links. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '11)*, vol. 17, no. 12, pp. 2249–2258, 2011. doi:10.1109/TVCG.2011.183. Cited on pages 9, 11, 33, 119, and 120.

[175] C. Stolte, D. Tang, and P. Hanrahan. Polaris: a system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transac-*

tions on Visualization and Computer Graphics, vol. 8, no. 1, pp. 52–65, 2002. doi:10.1109/2945.981851. Cited on page 31.

[176] M. Streit. *Guided Visual Analysis of Heterogeneous Data*. Ph.D. thesis, Graz University of Technology, 2011. Cited on pages 39 and 116.

[177] M. Streit, M. Kalkusch, K. Kashofer, and D. Schmalstieg. Navigation and exploration of interconnected pathways. *Computer Graphics Forum (EuroVis '08)*, vol. 27, no. 3, pp. 951–958, 2008. doi:10.1111/j.1467-8659.2008.01229.x. Cited on pages 48 and 50.

[178] 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, 2009. doi:10.1093/bioinformatics/btp432. Cited on page 11.

[179] M. Streit, H. Schulz, A. Lex, D. Schmalstieg, and H. Schumann. Model-Driven design for the visual analysis of heterogeneous data. *IEEE Transactions on Visualization and Computer Graphics*, vol. PP, no. 99, pp. 1–1, 2011. doi:10.1109/TVCG.2011.108. Cited on pages 9, 11, 99, and 116.

[180] A. Telea and D. Auber. Code flows: Visualizing structural evolution of source code. *Computer Graphics Forum (EuroVis '08)*, vol. 27, no. 3, pp. 831–838, 2008. doi:10.1111/j.1467-8659.2008.01214.x. Cited on page 30.

[181] The Cancer Genome Atlas Research Network. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, vol. 455, no. 7216, pp. 1061–1068, 2008. doi:10.1038/nature07385. Cited on page 96.

[182] The International SNP Map Working Group. A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature*, vol. 409, no. 6822, pp. 928–933, 2001. doi:10.1038/35057149. Cited on page 15.

[183] J. J. Thomas and K. A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Center, 2005. ISBN 0769523234. Cited on pages 6, 39, and 113.

[184] C. Tominski, J. Abello, and H. Schumann. Axes-based visualizations with radial layouts. In *Proceedings of the ACM Symposium on Applied Computing (SAC '04)*, SAC '04, p. 1242–1247. ACM, New York, NY, USA, 2004. ISBN 1-58113-812-1. doi:10.1145/967900.968153. Cited on page 23.

[185] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, vol. 12, no. 1, pp. 97–136, 1980. Cited on page 31.

[186] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, Connecticut, second edn., 1983. Cited on pages 23 and 52.

[187] C. Turkay, J. Parulek, N. Reuter, and H. Hauser. Integrating cluster formation and cluster evaluation in interactive visual analysis. In *Proceedings of the Spring Conference on Computer Graphics (SCCG '11)*, 2011. Cited on page 76.

[188] C. Turkay, J. Parulek, N. Reuter, and H. Hauser. Interactive visual analysis of temporal cluster structures. *Computer Graphics Forum (EuroVis '11)*, vol. 30, no. 3, pp. 711–720, 2011. doi:10.1111/j.1467-8659.2011.01920.x. Cited on page 76.

[189] G. J. G. Upton. Cobweb diagrams for multiway contingency tables. *Journal of the Royal Statistical Society. Series D (The Statistician)*, vol. 49, no. 1, pp. 79–85, 2000. Cited on pages 37 and 38.

[190] S. Uselton, J. Ahrens, W. Bethel, L. Treinish, and A. State. Multi-Source data analysis challenges. In *Proceedings of the IEEE Conference on Visualization (Vis '98)*, pp. 501–504. IEEE, 1998. ISBN 0-8186-9176-X. doi:10.1109/VISUAL.1998.745353. Cited on page 113.

[191] E. E. Veas, E. Mendez, S. K. Feiner, and D. Schmalstieg. Directing attention and influencing memory with visual saliency modulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*, CHI '11, p. 1471–1480. ACM, 2011. ISBN 978-1-4503-0228-9. doi:10.1145/1978942.1979158. Cited on page 33.

[192] R. G. Verhaak, K. A. Hoadley, E. Purdom, V. Wang, Y. Qi, M. D. Wilkerson, C. R. Miller, L. Ding, T. Golub, J. P. Mesirov, G. Alexe, M. Lawrence, M. O'Kelly, P. Tamayo, B. A. Weir, S. Gabriel, W. Winckler, S. Gupta, L. Jakkula, H. S. Feiler, J. G. Hodgson, C. D. James, J. N. Sarkaria, C. Brennan, A. Kahn, P. T. Spellman, R. K. Wilson, T. P. Speed, J. W. Gray, M. Meyerson, G. Getz, C. M. Perou, and D. N. Hayes. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer Cell*, vol. 17, no. 1, pp. 98–110, 2010. doi:10.1016/j.ccr.2009.12.020. Cited on pages 18, 37, 97, 98, and 107.

[193] 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. Cited on pages 23 and 24.

[194] M. Viceconti, G. Clapworthy, D. Testi, F. Taddei, and N. McFarlane. Multimodal fusion of biomedical data at different temporal and dimensional scales. *Computer Methods and Programs in Biomedicine*, vol. 102, no. 3, pp. 227 – 237, 2011. doi:10.1016/j.cmpb.2010.04.017. Cited on page 40.

[195] M. Waldner. *WIMP Interfaces for Emerging Display Environments*. PhD thesis, Graz University of Technology, Graz, 2011. Cited on page 118.

[196] M. Waldner, A. Lex, M. Streit, and D. Schmalstieg. Design considerations for collaborative information workspaces in Multi-Display environments. In *Proceedings of the Workshop on Collaborative Visualization on Interactive Surfaces (VisWeek '09) - Technical Report LMU-MI-2010-2, Ludwig Maximilias University Munich:*, pp. 36–39, 2009. ISSN 1862-5207. Cited on page 12.

[197] 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. Cited on pages 9, 11, and 118.

[198] 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. Cited on page 118.

[199] G. H. Weber, O. Rubel, M. Huang, A. H. DePace, C. C. Fowlkes, S. V. Keranen, C. L. Luengo Hendriks, H. Hagen, D. W. Knowles, J. Malik, M. D. Biggin, and B. Hamann. Visual exploration of Three-Dimensional gene expression using physical views and linked abstract views. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, no. 2, pp. 296–309, 2009. doi:10.1109/TCBB.2007.70249. Cited on page 42.

[200] J. N. Weinstein. A postgenomic visual icon. *Science*, vol. 319, no. 5871, pp. 1772–1773, 2008. doi:10.1126/science.1151888. Cited on page 41.

[201] M. Wertheimer. Untersuchungen zur Lehre von der Gestalt. II. *Psychologische Forschung*, vol. 4, no. 1, pp. 301–350, 1923. doi:10.1007/BF00410640. Cited on pages 33 and 66.

[202] M. A. Westenberg, S. A. F. T. Van Hijum, O. P. Kuipers, and J. B. T. M. Roerdink. Visualizing genome expression and regulatory network dynamics in genomic and metabolic context. *Computer Graphics Forum (EuroVis '08)*, vol. 27, no. 3, pp. 887–894, 2008. doi:10.1111/j.1467-8659.2008.01221.x. Cited on page 44.

[203] K. Wetterstrand. DNA sequencing costs: Data from the NHGRI Large-Scale genome sequencing program. www.genome.gov/sequencingcosts, 2011. Last accesed Feb. 12 2012. Cited on page 18.

[204] 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. Cited on pages 41 and 86.

[205] J. M. Wolfe. What can million trials tell us about visual search? *Psychological Science*, vol. 9, no. 1, pp. 33–39, 1998. doi:10.1046/j.1365-2583.1998.00072.x. Cited on page 31.

[206] P. C. Wong and R. D. Bergeron. 30 years of multidimensional multivariate visualization. In *Scientific Visualization, Overviews, Methodologies, and Techniques*, pp. 3–33. IEEE Computer Society,, 1997. ISBN 0818677775. Cited on page 4.

[207] A. Woodruff, C. Olston, A. Aiken, M. Chu, V. Ercegovac, M. Lin, M. Spalding, and M. Stonebraker. DataSplash: a direct manipulation environment for programming semantic zoom visualizations of tabular data. *Journal of Visual Languages &*

Computing, vol. 12, no. 5, pp. 551–571, 2001. doi:10.1006/jvlc.2001.0219. Cited on pages 28, 29, and 39.

[208] R. Xi, T. Kim, and P. J. Park. Detecting structural variations in the human genome using next generation sequencing. *Briefings in Functional Genomics*, vol. 9, no. 5-6, pp. 405 –415, 2010. doi:10.1093/bfgp/elq025. Cited on page 16.

[209] J. Xu, K. Kochanek, S. Murphy, and B. Tejada-Vera. National vital statistics reports. deaths: Final data for 2007. *Center for Disease Control and Prevention Division of Vital Statistics*, vol. 58, 2010. Cited on page 18.

[210] J. Yang and S. Barlowe. A dimension management framework for high dimensional visualization. In *Advances in Information and Intelligent Systems*, no. 251 in Studies in Computational Intelligence, pp. 267–288. Springer, 2009. ISBN 9783642041419. doi:10.1007/978-3-642-04141-9_13. Cited on page 28.

[211] X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu. Scattering points in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics (Infovis '09)*, vol. 15, no. 6, pp. 1001–1008, 2009. doi:10.1109/TVCG.2009.179. Cited on page 24.

[212] S. Zhai, J. Wright, T. Selker, and S. Kelin. Graphical means of directing user's attention in the visual interface. In *Proceedings of the Conference on Human-Computer Interaction (INTERACT '97)*, pp. 59–66. Chapman & Hall, 1997. ISBN 0412809508. Cited on pages 32 and 33.

[213] S. Zhao, M. McGuffin, and M. Chignell. Elastic hierarchies: combining treemaps and node-link diagrams. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '05)*, pp. 57–64. IEEE Computer Society Press, 2005. doi:10.1109/INFVIS.2005.1532129. Cited on page 28.

[214] J. Zhou, S. Konecni, and G. Grinstein. Visually comparing multiple partitions of data with applications to clustering. In *Proceedings of the SPIE Conference on Visualization and Data Analysis (VDA '09)*, pp. 72430J–72430J–12. SPIE, 2009. doi:10.1117/12.810093. Cited on page 38.

[215] C. Ziemkiewicz and R. Kosara. Laws of attraction: From perceptual forces to conceptual similarity. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '10)*, vol. 16, no. 6, pp. 1009–1016, 2010. doi:10.1109/TVCG.2010.174. Cited on page 33.

Curriculum Vitae

Personal Data

Name	Dipl.-Ing. **Alexander Lex**, Bakk.rer.soc.oec.
Contact	Wartingergasse 46a/3, A-8010 Graz +43 699 13 25 25 25 contact@alexander-lex.at www.alexander-lex.at
Born	March 23, 1981 in Graz, Austria
Nationality	Austrian

Research and Work Experience

Aug.-Sept. 2011	**Center for Biomedical Informatics** at **Harvard Medical School** Visiting Researcher
Since Aug. 2010	**Graz University of Technology** Faculty Member and Lecturer (Universitätsassistent)
Since Aug. 2008	**Graz University of Technology** Research Assistant
2005	**Boom Software AG** (Leibnitz, Austria) Summer Intern; Software Development
2002	**Magna Steyr Fahrzeugtechnik AG & Co KG** (Graz, Austria) Summer Intern; IT Consulting, CAD Design
Since 2002	**Lex GesmbH** (Gratkorn, Austria) Part Time; IT Administration
2000	Military Service
1999	**Raiffeisen Zentralbank** (New York, NY, USA) Summer Intern; Office Functions
1998	**Hotel Espace Cite** (Carcassonne, France) Summer Intern; Reception
1996-2002	**Lex GesmbH** (Gratkorn, Austria) Part Time; Purchasing, Bidding, Metalworking

Education

Aug.-Sept. 2011	Visiting Researcher Harvard Medical School Supervision: Prof. Peter Park, Dr. Nils Gehlenborg
Since July 2008	Doctoral Program in Computer Science Graz University of Technology Supervision: Univ. Prof. DI Dr. Dieter Schmalstieg

2006–2008	MS ("Dipl.-Ing."), Graz University of Technology *Software Development and Business Management* graduated with highest distinction
2006–2007	Visiting Graduate Student McMaster University, Hamilton, On, Canada
2002–2006	Bachelor's Degree ("Bakk. rer. soc. oec"), Graz University of Technology *Software Engineering and Knowledge Management*
2000–2002	Engineering school HTBLuVA Graz Gösting (BULME) graduated with highest distinction
1991–1999	High school and secondary school Bundesrealgymnasium Kepler, Graz graduated with distinction

Project Grants, Honors and Awards

2011	Co-recipient of best paper award, IEEE InfoVis, 2011.
2011	Scholarship for short time academic research and expert courses abroad (KU-WI). Granted by the Graz University of Technology.
2011	**Project Grant**: "Caleydoplex - Information Exploration in Teams". Funded by the Austrian Science Fund (FWF), Grant no. P22902.
2010	Co-recipient of best student paper award, ACM Graphics Interface, 2010.
2009	**Project Grant**: "InGeneious - Visualization of biomolecular and clinical data". Funded by the Austrian Research Promotion Agency (FFG), BRIDGE program, Grant no. 385567.
2007	Award for excellent performance as a student ("Leistungsstipendium") granted by the Faculty of Computer Science, Graz University of Technology.
2007	Research grant for students ("Förderstipendium") awarded by the Faculty of Computer Science, Graz University of Technology.
2006	Joint Study scholarship for student exchange with McMaster University, Hamilton, On, Canada.

Teaching Experience: Courses at Graz University of Technology

Since 2010	**Selected Topics Computer Grapics** Lectures on Perceptional Issues, Color, Information Visualization, Visual Analytics, Flow Visualization.
Since 2009	**Distributed Systems** Development and supervision of lab assignments.
Since 2010	**Introduction to Scientific Work** Supervision of focus groups.
Since 2011	**Computer Graphics 1** Development and supervision of lab assignments.
Since 2011	**Computer Graphics 2** Development and supervision of lab assignments.

Since 2008 **Student Projects Supervision**
Supervised 14 student projects (bachelor's and master's projects).

Conference Functions

- Website and publicity co-chair for BioVis 2012
- Program committee member of IV 2010 and 2011.

Reviewing for

- IEEE Information Visualization (InfoVis) 2010, 2011
- IEEE Visual Analytics (VAST) 2010, 2011
- EuroVis 2010, 2011, 2012
- IEEE PacificVis 2010, 2011
- IV 2009, 2010, 2011

Graz, March 1, 2011