

Submitted by
DI Samuel Gratzl, Bsc.

Submitted at
**Institute of Computer
Graphics**

Supervisor and
First Examiner
**Assoz.Univ.-Prof.
Dipl.-Ing. Dr.techn.
Marc Streit**

Second Examiner
**Prof. Dr. Tobias
Schreck**

March 2017

Visually Guiding Users in Selection, Exploration, and Presentation Tasks



Doctoral Thesis
to obtain the academic degree of
Doktor der technischen Wissenschaften
in the Doctoral Program
Technische Wissenschaften

Abstract

Making scientific discoveries based on large and heterogeneous datasets is challenging. The continuous improvement of data acquisition technologies makes it possible to collect more and more data. However, not only the amount of data is growing at a fast pace, but also its complexity. Visually analyzing such large, interconnected data collections requires a user to perform a combination of selection, exploration, and presentation tasks. In each of these tasks a user needs guidance in terms of (1) **what** data subsets are to be investigated from the data collection, (2) **how** to effectively and efficiently explore selected data subsets, and (3) **how** to effectively reproduce findings and tell the story of their discovery. On the basis of a unified model called the *SPARE model*, this thesis makes contributions to all three guidance tasks a user encounters during a visual analysis session: The *LineUp* multi-attribute ranking technique was developed to order and prioritize item collections. It is an essential building block of the proposed guidance process that has the goal of better supporting users in data selection tasks by scoring and ranking data subsets based on user-defined queries. *Domino* is a generic visualization technique for relating and exploring data subsets, supporting users in the exploration of interconnected data collections. *Phoeva* is a novel open-source visual analytics platform designed to speed up the creation of domain-specific exploration tools. The final building block of this thesis is *CLUE*, a universally applicable framework for capturing, labeling, understanding, and explaining visually driven exploration. Based on provenance data captured during the exploration process, users can author “Vistories”, visual stories based on the history of the exploration. The practical applicability of the guidance model and visualization techniques developed is demonstrated by means of usage scenarios and use cases based on real-world data from the biomedical domain.

Zusammenfassung

Wissenschaftliche Entdeckungen anhand großer heterogener Datensätzen zu machen ist eine Herausforderung. Die kontinuierlichen technologischen Verbesserungen ermöglichen es, immer schneller große Datenbestände zu erfassen und zu speichern. Jedoch steigt nicht nur die Datengröße sehr schnell an, sondern auch deren Komplexität. Die visuelle Analyse solch großer verknüpfter Datenmengen erfordert, dass der Benutzer eine Kombination aus Selektions-, Explorations-, und Präsentationsaufgaben durchführt. Für jede dieser Aufgaben benötigen BenutzerInnen Unterstützung (Guidance) bezüglich: (1) **welche** Datenuntermenge aus einer Datensammlung näher untersucht werden soll; (2) **wie** diese Datenuntermenge effektiv und effizient exploriert werden kann; (3) **wie** Entdeckungen während der Analyse gefunden wurden und wie diese auch effektiv reproduziert und präsentiert werden können. Basierend auf einem vereinheitlichten Modell — dem *SPARE Modell* — beschreibt diese Dissertation Lösungen zu allen bisher genannten Guidance Aufgaben, denen BenutzerInnen während einer Analyse begegnen. Die *LineUp* Visualisierungstechnik wurde für die Reihung und Priorisierung von beliebigen Sammlungen basierend auf einer Kombination von Datenattributen entwickelt. Sie ist ein essentieller Bestandteil des vorgeschlagenen Guidance Prozesses, welcher das Ziel hat, BenutzerInnen bei der Datenselektion zu unterstützen, indem Daten basierend auf Benutzereingaben gereiht werden. *Domino* ist eine generische Visualisierungstechnik für die visuelle Exploration von Datenuntermengen und deren Relationen. *Phovea* ist eine neuartige Open-Source Plattform für die visuelle Analyse, welche die Erstellung von domainenspezifischen Explorationsanwendungen beschleunigt. Den Abschluss dieser Dissertation bildet *CLUE*, ein universell einsetzbares Konzept zur Sammlung (Capture), Annotation (Labelling), Verständigung (Understanding) und Erklärung (Explaining) von visuell geleiteten Analysen. Basierend auf der aufgezeichneten Analyse können BenutzerInnen “Vistories” erstellen, um Entdeckungen zu präsentieren und zu teilen. Die praktische Anwendbarkeit der entwickelten Modelle und Visualisierungstechniken wird mithilfe von Anwendungsbeispielen aus dem biomedizinischen Bereich gezeigt.

“I want to remember that I am not my title, my position, my possessions, my appearance, my neighborhood, my age, my body size, my race or my income level. I am a soul.”

— Unknown

Acknowledgements

I would like to thank my supervisor Marc Streit. Since day one he fully integrated me in every aspect, showed me everything and taught me a lot. Thank you for that. Moreover, Alexander Lex and Nils Gehlenborg. Together with Marc we wrote all my papers. Thank you for the fruitful discussions, the help, the brainstormings, and the last minute improvements before the deadline. To appreciate your contributions and efforts I write this thesis using the “we” form instead of “I”. These are our papers not mine alone.

My thank also goes to the remaining members of the Caleydo team, especially Hendrik Strobelt, Holger Stitz, and Christian Partl. I always had the feeling that we are not a bunch of individuals but a team in which everyone contributes what he or she can. I also want to thank Hanspeter Pfister for giving me the opportunity to stay a couple of months at his lab at Harvard University. This was a memorable time from a personal, teaching, and work point of view.

My family and friends, my anchors in this sometimes crazy and stressful world. Thanks mom and dad for supporting me over the years. My siblings and friends that stand by my side. I don’t know what future will bring, but I know we will be there for each other.

Finally, I would also like to thank all the one that challenged me over the years. Those who pushed me to go beyond my limits in various ways. All of the mentioned great people and the one I forgot unluckily helped to be exactly at this point, in which you my dear reader can read the result of a four-year long journey of my life.

This thesis and individual papers were supported in part by the Austrian Research Promotion Agency (FFG) (840232), the Austrian Science Fund (P27975-NBL, P 22902, and J 3437-N15), the State of Upper Austria (FFG 851460 and Dissertation scholarship), the Austrian Marshallplan Foundation, the European Union (324554), the Air Force Research Laboratory and DARPA grant FA8750-12-C-0300, the United States National Cancer Institute (U24 CA143867), the United States NIH/National Human Genome Research Institute (U01 CA198935, U24 CA144025, U24 CA143845, K99 HG00758, U01 CA1989353) and Boehringer Ingelheim Regional Center Vienna.

Contents

1. Introduction	1
1.1. Problem Statement and Goals	1
1.2. Contributions	2
1.3. Guidance Classification	9
1.4. Structure	12
2. Biomedical Background	14
2.1. Problem Domain and Data	14
2.2. Cancer Subtype Characterization	15
2.3. Challenges	16
3. Related Work	18
3.1. Visual Analysis of Heterogeneous Stratifications: StratomeX	19
3.2. Multi-Attribute Ranking	21
3.3. Multi-Dimensional Data Exploration	22
3.4. Presentation and Storytelling	26
4. LineUp	29
4.1. Introduction	30
4.2. Requirement Analysis	31
4.3. Ranking Design Space and Related Work	34
4.4. Multi-Attribute Ranking Visualization Technique	38
4.5. Use Cases	48
4.6. Evaluation	51
4.7. Conclusion	54
5. Guided StratomeX	55
5.1. Introduction	56
5.2. Application Areas	57
5.3. Method	59
5.4. Case Study	64

5.5. Conclusion	66
6. Domino	67
6.1. Introduction	68
6.2. Domino Visualization Technique	69
6.3. Spectrum of Supported Visualizations	78
6.4. Interface and Interaction Design	78
6.5. Use Cases	83
6.6. Discussion	87
6.7. Conclusion	89
7. CLUE	90
7.1. Introduction	91
7.2. CLUE Model	93
7.3. Realizing the CLUE Model	95
7.4. Usage Scenarios	100
7.5. Discussion	103
8. Phovea	106
8.1. Key Aspects	107
8.2. Related Platforms	108
8.3. Ecosystem	109
8.4. Additional Libraries	112
8.5. Discussion	113
9. Conclusion	114
9.1. Discussion and Future Work	114
9.2. Conclusion	117
Bibliography	118
A. Phovea Applications	131
A.1. Caleydo LineUp and LineUp.js	131
A.2. Caleydo Gapminder with CLUE	132
A.3. StratomeX.js with CLUE	133
A.4. Caleydo Pathfinder	134

List of Figures

1.1. The SPARE model	3
1.2. Relationships between my published contributions and the SPARE model .	4
1.3. Five aspects of guidance	10
1.4. Structure of this thesis with respect to the SPARE model	12
3.1. StratomeX at a glance	19
3.2. StratomeX user interface	20
3.3. Block visualizations in StratomeX	21
4.1. Illustration of different ranking visualization techniques.	34
4.2. A simple example demonstrating the basic design of the LineUp technique	39
4.3. Strategies for aligning serial combinations of attribute scores.	42
4.4. Parallel combination of three attributes	42
4.5. Comparison between rankings	44
4.6. Visual mapping editor for mapping attribute values to normalized scores .	47
4.7. Example of a customized food nutrition ranking	49
4.8. LineUp University usage scenario	50
5.1. Seamless integration of visual and computational components	57
5.2. Proposed guidance process	59
5.3. Guided StratomeX user interface	61
5.4. Flow chart of query options and the query wizard menu items	62
5.5. LineUp view with multi-attribute ranking and filtering	64
5.6. Illustration of Guided StratomeX, confirming a hypotheses from a study .	65
5.7. Screenshots of Guided StratomeX, illustrating finding	66
6.1. Domino showing relationships between subsets of a music charts dataset . .	68
6.2. The three block types in Domino	70
6.3. Block visualization techniques categorized by block type	72
6.4. Example sequence demonstrating the creation of a combined block	73
6.5. Overview of possible relationship representations characterized	76

6.6.	Examples of supported visualization techniques	79
6.7.	Interface of the Domino prototype	80
6.8.	Usage of placeholders and live previews to add new blocks to Domino . . .	81
6.9.	Relationship interaction mode	82
6.10.	Key findings on subtypes in glioblastoma multiforme	85
7.1.	Traditional workflow flow for visual data exploration and presentation . . .	91
7.2.	Information flow and stage transitions using the CLUE model	92
7.3.	Three examples of transitions within the CLUE model	94
7.4.	Provenance graph data model	96
7.5.	Close-ups of the provenance and story view	97
7.6.	Screenshot of a Gapminder-inspired application	101
7.7.	Screenshot of CLUE applied to the StratomeX technique	102
8.1.	Ecosystem of Phovea	109
A.1.	Screenshot of Caleydo LineUp	131
A.2.	Screenshot of Caleydo Gapminder	132
A.3.	Screenshot of Caleydo StratomeX	133
A.4.	Screenshot of Caleydo Pathfinder	134

List of Tables

1.1. Guidance classification of the primary contributions	10
6.1. Properties of the possible block relationship degrees	74

1 | Introduction

Contents

1.1. Problem Statement and Goals	1
1.2. Contributions	2
1.3. Guidance Classification	9
1.4. Structure	12

1.1. Problem Statement and Goals

Making scientific discoveries based on large and heterogeneous datasets is challenging. The continuous improvement of data acquisition technologies makes it possible to collect more and more data. However, not only the amount of data is growing at a fast pace, but also its complexity. Visually analyzing such large, interconnected data collections requires a user to perform a combination of selection, exploration, and presentation tasks. In each of these tasks a user needs **guidance** in terms of (1) **what** data subsets are to be investigated from the data collection, (2) **how** to effectively and efficiently explore selected data subsets, and (3) **how** to effectively reproduce findings and tell the story of their discovery.

In this thesis a data subset is defined as a homogeneous mathematical subset of a dataset. In the case of a heterogeneous multi-dimensional dataset consisting of multiple dimensions D of set I , a subset is $S \subseteq I_D$. For example, in a patient dataset, a data subset can be a single dimension, such as the patient's gender, or a group of patients for this dimension. In the case of two-dimensional homogeneous datasets $R \times C$ consisting of sets of rows R and columns C , a data subset S is defined as $S = (R_S \subseteq R) \times (C_S \subseteq C)$. This includes special cases in which only one column or row is included.

Ceneda et al. [CGM⁺17] have recently characterized guidance in the context of Visual Analytics using the following definition:

Guidance is a computer-assisted process that aims to actively resolve a knowledge gap encountered by users during an interactive visual analytics session.

The authors further clarified that “according to this definition, guidance is a dynamic process that aims to support users in a particular task” [CGM⁺17]. The goal of this work was to develop models and techniques that guide users in the three tasks of what data subset is to be selected, how to explore selected data subsets, and how to reproduce findings and tell the story of their discovery. Challenges included the heterogeneity of the different guidance tasks and how to combine solutions for these tasks into one coherent guidance system. A common approach is to prioritize the data collection, which results in a ranking. However, when considering rankings that are based not on a single but on a combination of multiple scores, current approaches lack support in effectively communicating what drives the ranking. In addition, most visual exploration tools do not cover the aspects of presentation and reproducibility. Taking a screenshot is a common approach to presenting the findings gained in a visual analysis session. However, it has several drawbacks, for instance, that analysts cannot go back to the original exploration, since a screenshot is a static artifact.

1.2. Contributions

As outlined in the previous section, the goal of this work was to propose solutions for visually guiding users in selection, exploration, and presentation tasks. This goal can be divided into five interleaving stages that form the **SPARE** model (**S**coreing, **P**resentation, **A**uthoring, **R**anking, and **E**xploration), as illustrated in Figure 1.1. The transitions between the stages are split into user stage transitions on the one hand and information flow on the other hand. While information flow is unidirectional, users can transition freely between the stages.

Within the *Exploration* stage the analyst explores the dataset using interactive visualizations and defines a query that describes the targets of an automated search for particular patterns in the data collection and needs support in the selection task. The *Scoring* stage computes for each possible subset in the dataset collection a score that indicates how well the subset matches the query. The scoring algorithm itself is domain- and query-specific. Further, the interpretation of the score depends on the algorithm used; for example, in the case of a score that computes p-values, the smaller the value the better, while for a similar-

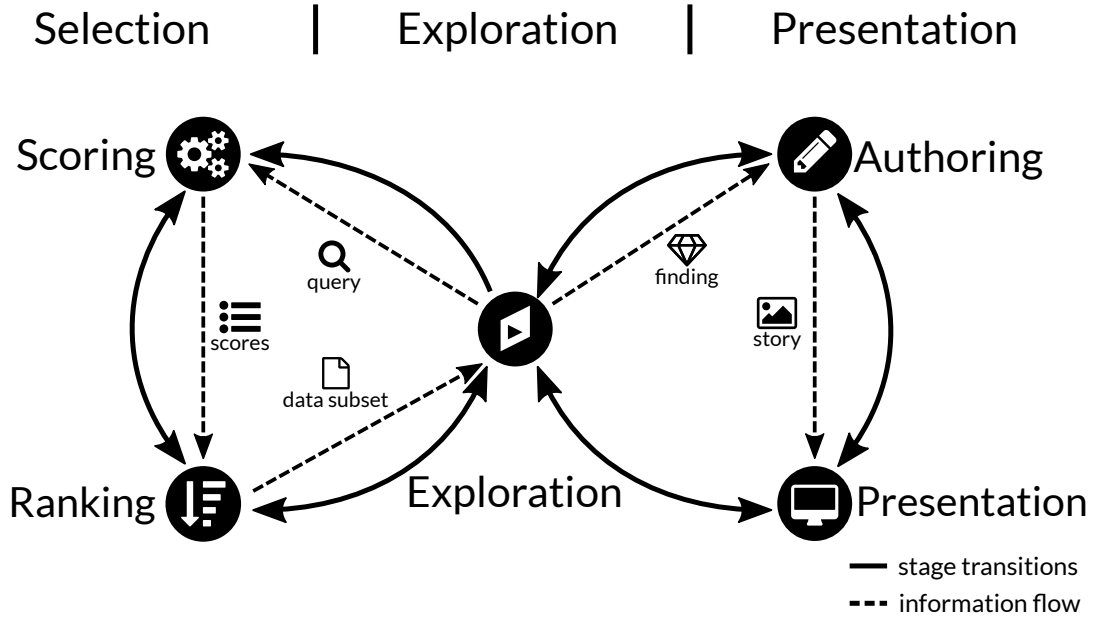


Figure 1.1.: The SPARE (Scoring, Presentation, Authoring, Ranking, and Exploration) model that constitutes the conceptional basis for guiding the user in selection, exploration, and presentation tasks.

ity score the higher the better. To further analyze the resulting scores, the user switches to the *Ranking* stage in order to prioritize and rank the computed scores. The ranking itself is influenced by multiple attributes, including the computed score, meta-data, and other user-defined parameters, such as filters. Based on the resulting ranking, the analyst selects one or multiple results that match the query and continues the exploration in the *Exploration* stage, in which guidance mechanisms support the analyst in relating the currently added data subset to existing ones. This iterative process of querying, scoring, ranking, and exploring data subsets follows the visual analytics process by Keim et al. [KKEM10].

After discovering a finding in the *Exploration* stage, the analyst can switch to the *Authoring* stage in order to prepare a shareable, understandable, and reproducible story describing how the finding was discovered. The resulting story will finally be presented in the *Presentation* stage, in which consumers of the story can go back to the *Exploration* stage in order to continue the exploration.

The realization of the SPARE model was described in multiple publications that are part of this thesis. Figure 1.2 illustrates the relationship between the SPARE model and the publications. Publications indicated in bold are primary papers on which this thesis builds. The author of this thesis is also the first author of these paper. An exception is the

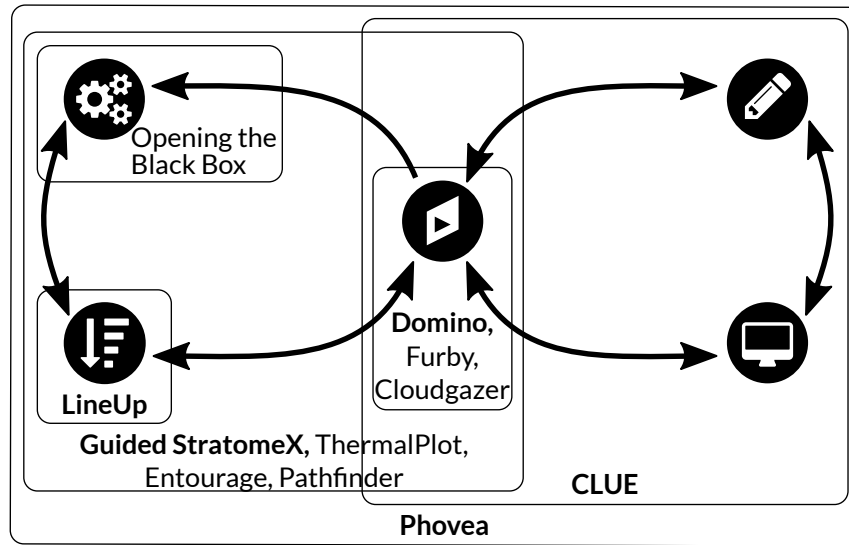


Figure 1.2.: Relationships between individual published contributions and the SPARE model. Publications indicated in bold form the basis of this thesis, while the rest are secondary publications to which the author of this thesis contributed.

publication on *Guided Visual Exploration of Genomic Stratifications in Cancer*, work on which began before the author of this thesis started his PhD studies. However, he later joined the team and made significant contributions.

1.2.1. Primary Publications

The following list of peer-reviewed publications forms the core of this thesis. Individual chapters are based on these publications.

LineUp: Visual Analysis of Multi-Attribute Rankings [GLG⁺13]: Samuel Gratzl, Alexander Lex (then Harvard University, now University of Utah), Nils Gehlenborg, Hanspeter Pfister (both Harvard University), and Marc Streit (JKU Linz). *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 19(12):2277-2286. Acceptance Rate: 25.1%

LineUp is a visualization technique for interactively creating and working with multi-attribute rankings. It is therefore assigned to the Ranking stage of the SPARE model, which is concerned with guiding users in the selection tasks by prioritizing data collections. Using intuitive and easy-to-use encodings and interactions, analysts are able to better un-

derstand, analyze, and manipulate complex rankings such as university rankings. The initial concept already existed when the author started his PhD studies, since it was originally designed merely as a support view for the publication on *Guided StratomeX* [SLG⁺14]. However, we soon found out that rankings are a common problem in various domains and thus we extracted and generalized our idea.

Together with the co-authors, the author of this thesis refined the initial concept, generalized it, and wrote the manuscript. In addition, he implemented the improved version of the original prototype and conducted the accompanying user study.

LineUp has been adopted both by the scientific community and by companies. For example, Kitware Inc. integrated LineUp in their *Candela*¹ suite, and Microsoft provides LineUp as an extension² to their *Power BI* visual analysis product. The publication received the Best Paper Award (152 submissions) at the IEEE VIS conference in 2013. Tamara Munzner also used LineUp as an example in her text book “Visualization Analysis and Design” [Mun14], which is used as primary literature in numerous visualization courses.

Guided StratomeX: Guided Visual Exploration of Genomic Stratifications in Cancer [SLG⁺14]: Marc Streit (JKU Linz) & Alexander Lex (then Harvard University, now University of Utah), **Samuel Gratzl**, Christian Partl, Dieter Schmalstieg (both Graz University of Technology), and Hanspeter Pfister, Peter J. Park & Nils Gehlenborg (all Harvard University). *Nature Methods*, 11(9):884-885. Impact Factor: 32.07

In this paper we introduced a guidance process for selection tasks in visual exploration applications, that corresponds to the left part of the SPARE model. We focused on supporting users in cancer subtype characterization by applying our guidance process to an extended version of the previously published comparative set visualization technique *StratomeX* [LSS⁺12] combined with our ranking technique LineUp. For an introduction to the domain problem of cancer subtype characterization refer to Section 2.2.

The initial concept and project was conceived by Marc Streit, Alexander Lex, and Nils Gehlenborg. Building on the *StratomeX* visualization technique [LSS⁺12], the author of this thesis contributed to the guidance process definition and supplementary material. In addition, he extended, integrated, and implemented the guidance process in the Caleydo framework. The extended version is hereafter referred to as “Guided StratomeX”.

¹<https://candela.readthedocs.io/en/latest/components/lineup.html>

²<https://app.powerbi.com/visuals/show/TableSorter1450434005853>

Domino: Extracting, Comparing, and Manipulating Subsets across Multiple Tabular Datasets [GGL⁺14]: Samuel Gratzl, Alexander Lex (then Harvard University, now University of Utah), Nils Gehlenborg, Hanspeter Pfister (Harvard University), and Marc Streit (JKU Linz). *IEEE Transactions on Visualization and Computer Graphics (InfoVis '14)*, 20(12):2023-2032. Acceptance Rate: 23.0%

Domino is a general visualization technique for relating attributes between different data subsets on various levels of granularity. By reducing the basic elements to blocks and relationships, users can recreate a variety of existing visualization techniques. *Domino* is assigned to the Exploration stage of the SPARE model as it supports users in relating data subsets by using placeholders and previews.

The author designed the technique in collaboration with the other co-authors of the paper. Together with Marc Streit, he wrote major parts of the paper. In addition, he implemented the prototype. This paper won a Best Paper Honorable mention Award at IEEE VIS 2014 (196 submissions).

Phovea/Caleydo Web: An Integrated Visual Analysis Platform for Biomedical Data [GGL⁺15]: Samuel Gratzl, Nils Gehlenborg (Harvard University), Alexander Lex (University of Utah), Hendrik Strobelt (Harvard University), Christian Partl (Graz University of Technology), and Marc Streit (JKU Linz). *Poster Compendium of the IEEE Conference on Information Visualization (InfoVis '15)*.

This poster introduced the *Caleydo Web* platform, which has recently been renamed to *Phovea*. *Phovea* is an open-source visual analysis platform designed for the challenges encountered in the biomedical domain. Unlike its predecessor, the Java-based *Caleydo* framework [LSKS10], *Phovea* is based on a web-client-server architecture. In addition to the platform and a corresponding client and server framework, *Phovea* provides development tools that support visualization researchers in efficiently creating new *Phovea* applications. The source code is available at <http://github.com/phovea>. A collection of applications written using *Phovea* is hosted at <http://caleydoapp.org>. Further information can be found at <http://phovea.caleydo.org>.

The author designed the architecture and implemented most of the platform. In addition, he wrote major parts of the manuscript. This poster presented a preview of the platform, since it is still under active development at the time this thesis is written.

CLUE: From Visual Exploration to Storytelling and Back Again [GLG⁺16]: Samuel Gratzl, Alexander Lex (University of Utah), Nils Gehlenborg (Harvard University), Nicola Cosgrove, and Marc Streit (both JKU Linz). *Computer Graphics Forum (EuroVis '16)*, 35(3):491–500. Acceptance Rate: 27.3%

This paper introduced a generic model called *CLUE* for capturing, labeling, understanding, and explaining visualization-driven exploration. The CLUE model corresponds to the right side of the SPARE model. In this work, we proposed tracking all actions performed during the visual analysis in a provenance graph. On the basis of the provenance graph, analysts can create “Vistories”—visual stories based on the exploration—in order to present findings. Vistories can be shared with and viewed by other analysts. Moreover, analysts can go back to the exploration and continue the analysis at any point of the Vistory. In addition to the model, a prototype library based on *Phovea* was implemented that allows visual exploration applications to be extended with CLUE capabilities.

In collaboration with the other co-authors of the paper, the author of this thesis designed the CLUE model. In addition, he wrote major parts of the manuscript and implemented the prototype.

1.2.2. Secondary Publications

In addition to the primary publications that form the core of this thesis, the author contributed to several related peer-reviewed papers. The following list contains an overview including a short description of each paper, the author’s contributions and how each paper relates to this thesis.

Furby: Fuzzy Force-Directed Bicluster Visualization [SGG⁺14]: Marc Streit, Samuel Gratzl, Michael Gillhofer, Andreas Mayr, Andreas Mitterecker, and Sepp Hochreiter (all JKU Linz). *BMC Bioinformatics*, 15(Suppl 6):S4. Impact Factor: 2.789

Furby is a visualization technique for exploring fuzzy bicluster results. *Furby* generalizes the *StratomeX* visualization technique by removing the restriction of relating data subsets in the horizontal direction only. Together with Marc Streit, the author of this thesis designed the concept and wrote large parts of the manuscript. In addition, he extended and improved the prototype implemented as part of Michael Gillhofer’s Bachelor thesis [Gil13].

Opening the Black Box: Strategies for Increased User Involvement in Existing Algorithm Implementations [MPG⁺14]: Thomas Mühlbacher, Harald Piringer (both VRVis Forschungs GmbH), **Samuel Gratzl**, Michael Sedlmair (University of Vienna), and Marc Streit (JKU Linz). *IEEE Transactions on Visualization and Computer Graphics (VAST '14)*, 20(12):1643-1652. Acceptance Rate: 22.6%

This paper characterized types of user involvement in ongoing computation based on a systematic literature review, and presented strategies for achieving user involvement in visual analytics applications. This paper focused on the Scoring stage of the proposed SPARE model and on how users can be involved in various ways in the automatic computation of scores. The author contributed to the overall concept, literature review, and systematic evaluation of existing algorithm packages.

CloudGazer: A Divide-and-Conquer Approach to Monitoring and Optimizing Cloud-Based Networks [SGKS15]: Holger Stitz (JKU Linz), **Samuel Gratzl**, Michael Krieger (RISC Software GmbH), and Marc Streit (JKU Linz). *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis '15)*, pages 175-138. Acceptance Rate: 30.4%

Cloudgazer is a platform and visualization technique for monitoring large network nodes described by time-varying node and relationship attributes. Cloudgazer supports analysts in the exploration of heterogeneous networks by splitting the graph into multiple linked hierarchies. It therefore relates to the Exploration stage of the SPARE model. The author contributed to the initial concept, design and implementation of the platform and the manuscript.

ThermalPlot: Visualizing Multi-Attribute Time-Series Data Using a Thermal Metaphor [SGAS16]: Holger Stitz (JKU Linz), **Samuel Gratzl**, Wolfgang Aigner (St. Poelten University of Applied Sciences), and Marc Streit (JKU-Linz). *IEEE Transactions on Visualization and Computer Graphics*, 2016. Impact Factor: 1.4

ThermalPlot is a visualization technique designed to provide an overview of large time varying item collections such as stock-market data. The ThermalPlot technique uses LineUp to filter and preselect interesting items in the collection, implementing the guidance process. The author contributed to the initial design, the implementation, and the manuscript.

Entourage: Visualizing Relationships between Biological Pathways using Contextual Subsets [LPK⁺13]: Alexander Lex (then Harvard University, now University of Utah), Christian Partl, Denis Kalkofen (both Graz University of Technology), Marc Streit

(JKU Linz), Anne Mai Wassermann (then Novartis Institute for BioMedical Research, now Pfizer), **Samuel Gratzl**, Dieter Schmalstieg (Graz University of Technology), and Hanspeter Pfister (Harvard University). *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 19(12):2536-2545. Acceptance Rate: 25.1%

Entourage is a visual exploration technique for exploring experimental data in interconnected but logically separated biological pathways. Analysts can prioritize pathways based on the similarity to a selected pathway using LineUp. The author contributed to the implementation of the prototype by integrating the guidance process.

Pathfinder: Visual Analysis of Paths in Graphs [PGS⁺16]: Christian Partl (Graz University of Technology), **Samuel Gratzl**, Marc Streit (JKU Linz), Anne Mai Wassermann (Pfizer), Hanspeter Pfister (Harvard University), Dieter Schmalstieg (Graz University of Technology), and Alexander Lex (University of Utah). *Computer Graphics Forum (EuroVis '16)*, 2016, 35(3):71-80. Acceptance Rate: 25.1%

Pathfinder is a technique for querying, ranking, and exploring paths in large graphs. Pathfinder uses an adapted version of the guidance process in which the Ranking and Exploration stages are merged by focusing the visual analysis on relations of the prioritized paths. The author contributed to the implementation by extending the underlying Phovea platform and by implementing the server components of the technique for incremental fetching of query results from the underlying graph database. This paper won a Best Paper Honorable Mention Award (183 paper submissions) at EuroVIS 2016.

1.3. Guidance Classification

This section describes the individual primary publications and classifies them according to the guidance characterization published by Ceneda et al. [CGM⁺17]. An exception is the work on the *Phovea* visual analysis platform, since it is a technical framework that is not directly exposed to users. Ceneda et al. characterize guidance according to five aspects, as shown in Figure 1.3.

The *Knowledge Gap Type* identifies whether (a) the target the user wants to achieve, (b) the path via which a certain result can be achieved, or (c) both are unknown. For example, the analyst knows that she wants to cluster the data subset (= target known) but does not know how to trigger it (= path unknown). The *Knowledge Gap Domain* defines the domain in which the user needs guidance—most importantly data, but also tasks or infrastructure. *Input* refers to the information that is used as a basis for the guidance, ranging from data,

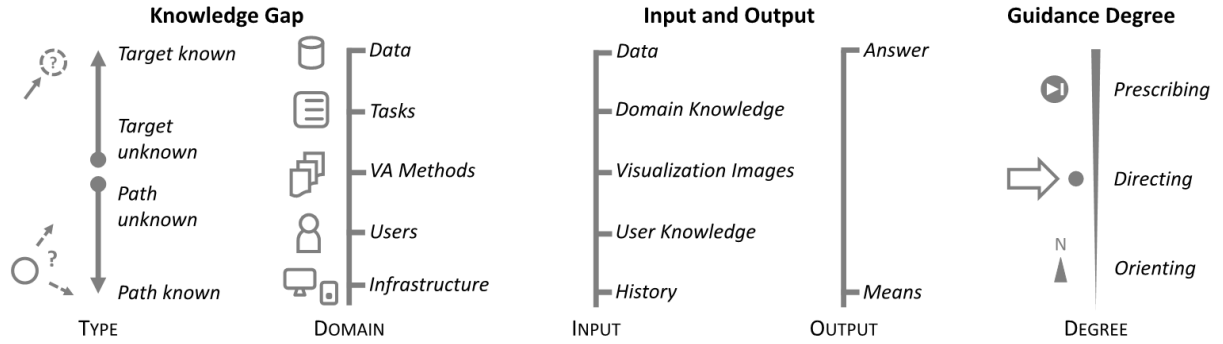


Figure 1.3.: Five aspects of guidance, according to Ceneda [CGM⁺17]

to visualization images and the history of the analysis. *Output* characterizes what kind of answers the user receives (direct vs. indirect) and by which means they are presented, most importantly visually. Finally, the *Guidance Degree* specifies to which degree users are guided, ranging from *Orienting* (no active guidance but provides overview), to *Directing* (points to the solution) and *Prescribing* (activity leads to solution). For a more detailed description of the aspects of guidance refer to the original paper [CGM⁺17].

Table 1.1 shows the classification of the primary publications.

Paper	Knowledge Gap		Input and Output		Guidance Degree
	Type	Domain	Input	Output	
<i>LineUp</i>	<ul style="list-style-type: none"> • target • path 	<ul style="list-style-type: none"> • data 	<ul style="list-style-type: none"> • data 	<ul style="list-style-type: none"> • answer indirect by visual means 	<ul style="list-style-type: none"> • orienting • directing
<i>Guided StratomeX</i>	<ul style="list-style-type: none"> • target • path 	<ul style="list-style-type: none"> • data 	<ul style="list-style-type: none"> • data • domain models 	<ul style="list-style-type: none"> • answer direct • answer indirect by visual means 	<ul style="list-style-type: none"> • orienting • directing • prescribing
<i>Domino</i>	<ul style="list-style-type: none"> • target 	<ul style="list-style-type: none"> • data • VA methods 	<ul style="list-style-type: none"> • data 	<ul style="list-style-type: none"> • answer direct by visual means 	<ul style="list-style-type: none"> • orienting
<i>CLUE</i>	<ul style="list-style-type: none"> • target • path 	<ul style="list-style-type: none"> • tasks 	<ul style="list-style-type: none"> • history 	<ul style="list-style-type: none"> • answer direct by visual means 	<ul style="list-style-type: none"> • orienting • directing • prescribing • fully automated

Table 1.1.: Guidance classification of the primary contributions according to Ceneda et al. [CGM⁺17].

1.3.1. LineUp

LineUp supports users in filling the knowledge gap of finding an unknown target, i.e., the best possible data subsets from a data collection, by ranking them. In addition, as part of our future work, we propose to help users in finding a particular desired ranking by optimizing how attributes can be combined, thus addressing the “path unknown” knowledge gap. *LineUp* operates in the data domain and uses the data as input for its guidance. However, the answers are just indirect through the ranking, since the actions based on the top-ranked items are up to the analyst. In summary, *LineUp* realizes two guidance degrees: (1) *Orientation* by providing histograms showing distributions of individual attributes and (2) *Directing* through the ranking itself, which is the basis for the decision process.

1.3.2. Guided StratomeX

The guidance process we proposed as part of *Guided StratomeX* supports users in filling the knowledge gap of deciding which data subset to select by providing a “wizard-style” interface that lets the user choose between predefined query options. In addition to the dataset collection, domain models are the input for the guidance. Domain models in this context are specialized scoring algorithms chosen for a particular domain problem. This includes, for example, a log-rank test in the biomedical domain for quantifying the significance of differences in the survival of patients. Answers are both direct, since the scores provide the analyst with specific results based on the query formulated and indirect as in *LineUp*. With our guidance process we support multiple guidance degrees: *Orientation* through the overview provided by *LineUp*; *Directing* by the ranking that can be created via *LineUp*; and *Prescribing* through the wizard interface that supports users in formulating queries.

1.3.3. Domino

This technique supports users in exploring relationships of data subsets. Not only are users supported in the data domain, visualization techniques (VA methods) are also suggested through placeholders and previews. However, since users are not actively steered, *Domino* provides guidance only to the degree of *Orientation*.

1.3.4. CLUE

The *CLUE* model takes the history of the exploration as input to provide guidance for creating Vistories. The analyst who creates the Vistory guides the consumer of the Vistory through an analysis. In this sense, CLUE covers the full spectrum of guidance degrees: *Orientation* in the task domain is provided by showing the provenance graph during the exploration itself and through indicating the current step in the Vistory. Providing *directions* will be part of future work that seeks to analyze recorded provenance graphs to guide users to similar states in the exploration. *Prescribing* and *fully automated guidance* are realized through Vistories, in which the consumer can either use a stepper interface to watch a Vistory or play the Vistory as an animation.

1.4. Structure

This thesis is structured as follows. Figure 1.4 illustrates the relation to the SPARE model.

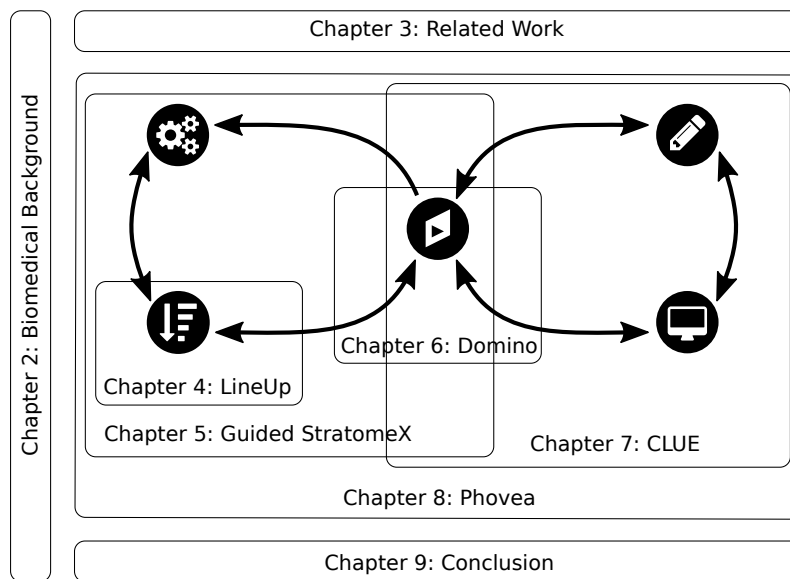


Figure 1.4.: Structure of this thesis with respect to the SPARE model

Chapter 2 introduces the characteristics and semantics of typical dataset types in the biomedical domain. In addition, the specific challenges of this domain are highlighted based on the problem of cancer subtype characterization. This chapter is partly based on the section “Biological Background and Data” in [LSS⁺12].

Chapter 3 discusses related work based on the presented model. Further, it explains the preexisting visualization technique *StratomeX* [LSS⁺12], which is an example exploration technique used throughout this thesis.

Chapter 4 introduces the *LineUp* visualization technique for analyzing multi-attribute rankings. LineUp is a general technique for prioritizing item collections for decision-making. It is later used as an essential part of the model presented in the following chapter. This chapter is based on [GLG⁺13].

Chapter 5 describes a guidance process for selection-based guidance tasks, reflecting the left side of the SPARE model. The presented model is described using the biomedical use case of cancer subtype characterization as an example. The StratomeX visualization technique was adapted and extended by the proposed guidance process and the previously presented LineUp technique for visualizing rankings. This chapter is based on [SLG⁺14].

Chapter 6 introduces the *Domino* visualization technique for extracting, comparing, and manipulating subsets across multiple tabular datasets. Domino is a general exploration technique that allows many existing visualization techniques to be replicated, such as StratomeX, Scatterplots, and Parallel Coordinates. This chapter is based on [GGL⁺14].

Chapter 7 presents the *CLUE* model for capturing, labeling, understanding, and explaining visualization-driven explorations. This model reflects the right side of the SPARE model. In order to reproduce discoveries and visual analysis sessions, provenance data is collected. This chapter is based on [GLG⁺16].

Chapter 8 focuses on *Phovea*, an integrated visual analysis platform for biomedical data. Motivated by a collection of key aspects this chapter describes the ecosystem and its architecture. This chapter is based on [GGL⁺15].

Chapter 9 concludes this thesis with a discussion of the presented contributions along with possible future work.

2 | Biomedical Background

Contents

2.1. Problem Domain and Data	14
2.2. Cancer Subtype Characterization	15
2.3. Challenges	16

Biomedicine is a prime example of a field in which Visual Analytics is applied. In this chapter, the characteristics of this domain, a subset of data that is relevant to this thesis, and challenges are introduced. In addition, the specific domain problem of characterizing cancer subtypes is discussed in detail. The following chapters describe use cases and usage scenarios from this domain in order to show the practical applicability of the proposed techniques and models.

2.1. Problem Domain and Data

The cell is a fascinating building block of life. Complex processes, so-called *metabolic pathways*, within a cell create and define what we are. The blueprints for all processes are stored in the DNA—a sequence of base nucleotides. We differentiate between four different base types: cytosine (C), guanine (G), adenine (A), and thymine (T). DNA is packed into multiple chromosomes, and within a chromosome it is logically grouped into genes. Around 20,000 genes exist in the human genome. The DNA acts as a blueprint for building proteins. An introduction to molecular biology can be found in the book “The Processes of Life” by Laurence Hunter [Hun09], which explains molecular biology to computer scientists. The process of converting the blueprint stored in the DNA into a protein consists of multiple steps, including transcribing the DNA into mRNA and the synthesis of polypeptides and thus proteins. While each cell of an organism contains the same DNA, different cell types have different purposes and functions. The reason is that

genes are differently expressed in different cell types, which means that cell types differ in the set of genes that are expressed and also in the expression levels, i.e. the amounts of gene product (mostly proteins) produced.

One of the drivers of evolution are alterations in the process of creating proteins. These alterations can occur at different levels and granularities. A *Single Nucleotide Polymorphism (SNP)* is the smallest change, indicating that one nucleotide (CGAT) base pair at a particular position in the genomic DNA is altered. A *Mutation* indicates that one or more base pairs are inserted or deleted. *Copy Number Variations (CNVs)* can occur during cell division and result in particular certain regions of the DNA and whole genes being copied incorrectly to the replicated DNA - the variations range from DNA not being copied at all (deletion) to multiple copies being produced (amplification). Finally, *chromosomal alterations* are special a kind of CNV in which whole chromosomes are copied incorrectly. All of the aforementioned alterations may influence the function of a cell. For example, the *APOE* SNP was found to play a role in Alzheimer's disease. Mutations in the genes *BRCA1* and *BRCA2* are related to breast cancer. CNVs can lead to an increased/decreased or even loss of function of a gene and have often an effect on the gene expression level of the altered gene. For example, an increased gene expression level can be caused by too many copies of the corresponding gene.

While this covers only a small portion of how a cell works, it shows the type of data that is usually captured in this context. Commonly acquired data on a gene includes: its expression level (numerical value indicating amplification level), whether it is mutated (yes/no), and what kind of CNV it has (amplified/normal/deleted). These values are acquired for each of the approximately 20,000 genes for each tissue sample or even for each cell within a sample. If the sample is taken from a patient, additional meta-data is collected, such as age, race, and gender of the patient. In the case of multiple patients or samples, the resulting dataset can be represented as matrices containing for each gene and sample the expression level, mutation, or CNV data.

2.2. Cancer Subtype Characterization

Cancer is a class of complex diseases caused by a series of alterations in the DNA and related elements in the cell. These alterations can trigger abnormal cell growth rates which results in a tumor. The *Cancer Genome Atlas (TCGA)* project ¹ is one of the most comprehensive sources for cancer genomics data sets. The goal of this project was to collect for each of the twenty most common cancer types 500 patient samples and to acquire for each

¹<http://cancergenome.nih.gov>

sample high-quality genomic data including gene expression levels, mutation frequencies, and CNVs. In addition, clinical parameters for each patient were collected, such as gender, race, and how many days the patient survived (days to death). Due to the incremental collection and processing of tumor samples, the datasets are changing frequently. The Broad Institute of MIT and Harvard University developed an automated analysis pipeline called Firehose² to preprocess and perform comprehensive automated analyses on each tumor cohort. The results are publicly available.

One of the goals of the TCGA project was to identify and characterize cancer subtypes. Traditionally, cancer types are classified according to the tissue or cell type they occur in, but it was found that this classification is too coarse-grained. For example, different breast cancers can be distinguished based on different genomic alterations. These cancer subtypes are not fully identified, but play an important role in prognosis and treatment of patients.

In addition to the raw-data processing performed within the Firehose pipeline, unsupervised clustering methods are applied to the gene expression level matrices, to automatically identify cancer subtypes by grouping patients. However, a variety of algorithms with different parameter settings are applied as part of the pipeline. Each clustering result is called a *stratification*, in which each cluster constitutes a cancer subtype candidate. In order to identify which candidates are actual cancer subtypes, biologists need to find supporting evidence. For example, a statistically significant difference in survival rates between two candidates supports a separation into two groups. Survival rates correspond to the collected days-to-death attribute and show how soon patients with a specific type of cancer die. Other types of supporting evidence are mutations and CNV. For example, scientists found that mutations in the genes *BRCA1* and *BRCA2* are related to certain breast cancer subtypes. However, cancer is a multifactorial disease (i.e. the result of multiple alterations), which complicates the identification of supporting evidence.

2.3. Challenges

The previous section introduced the problem of cancer subtype characterization, which involves several challenges that need to be addressed:

Heterogeneity plays a major role in this context. Multidimensional datasets range from homogeneous gene expression level matrices, to binary mutation and ordinal CNV matri-

²<http://gdac.broadinstitute.org>

ces, and heterogeneous clinical variable tables. Furthermore, graphs representing metabolic pathways of a cell (with node and link attributes) pose challenges in exploring of these datasets.

Data Size: The TCGA projects aimed to collect data for over 10,000 patients and to acquire data for each of the approximately 20,000 genes. Although, in contrast to big data produced in other fields, this data size is manageable compared to big data produced in other fields, automated data mining methods have limited applicability. Within the TCGA Firehose pipeline, for example, several different clustering algorithms with different parameter settings are executed to identify cancer subtypes. A problem arises in identifying which resulting stratification can be confirmed by supporting evidence. Visual Analytics is used for these purpose. While automatic methods can support the user in this task by preselecting potential candidates, users with their domain knowledge and ability to detect hidden patterns are essential in this process. Having the user in the loop, however, requires interactive visualizations of the underlying data, for which the data size becomes an issue.

Combinatorial Explosion is another problem when trying to identify cancer subtypes based on supporting evidence. In theory, each mutation, CNV, or categorized meta-data of the 20,000 genes could characterize a cancer subtype. In practice, combinations of genes often have effects on each other, which leads to an extensive number of possibilities to test. Visualizations can help to identify these patterns when analysts look at the right combination of data subsets. However, choosing which data subsets to visualize in what order is challenging. Guidance systems supporting users in this task can help to prioritize potential interesting data subsets.

Missing Values and Uncertainty both play a role in the biomedical field. Clinical variables can be missing or contain invalid values, for instance, a negative days-to-death value. Both low signal-to-noise ratio and measurement errors during acquisition of the genomic data increase uncertainty. In addition, methods that process the raw measurements are evolving, which needs to be considered in the analysis. Clusters identified in one analysis run may be invalid in another, due to changes in the processing engine.

Domain-specific knowledge is a major challenge in this area. Since not all biological processes are fully understood, specialized knowledge in various domains (e.g, genetics, biochemistry, immunology) is required to correctly interpret the data that has to be encoded in the visualization. Thus, domain experts must always be in the loop when specialized visualizations are created for them.

3 | Related Work

Contents

3.1. Visual Analysis of Heterogeneous Stratifications: StratomeX	19
3.2. Multi-Attribute Ranking	21
3.3. Multi-Dimensional Data Exploration	22
3.4. Presentation and Storytelling	26

This chapter starts with introducing the *StratomeX* visualization technique for analyzing heterogeneous stratifications. *StratomeX* is the starting point for this thesis. It is used in an extended version as an essential building block of the tool described in Chapter 5 for which the work has started before the author of this thesis started his PhD studies. The related work of this thesis is grouped by different aspects of the proposed SPARE model introduced in Section 1.2. It starts with a detailed discussion about multi-attribute rankings and how they can be visualized. It follows with a discussion of existing work related to the proposed multi-dimensional visual exploration technique *Domino*, representing the center stage of the SPARE model. Related work how to present a visual analysis using storytelling techniques concludes this chapter. Note that the missing Scoring stage of the SPARE model is not discussed, since the available and useful scoring algorithms heavily depend on the problem domain. A set of scoring algorithms for the biological domain will be presented as part of Chapter 5.

3.1. Visual Analysis of Heterogeneous Stratifications: StratomeX

StratomeX is a visualization technique first published by Lex et al. [LSS⁺12] in 2012. It extends the parallel set visualization technique by showing the stratification together with the data that leads to the stratification. This allows the analyst to better compare sets and highlights the causes that lead to a stratification. Cancer subtype characterization as described in Section 2.2 is a prime example in which *StratomeX* can be used.

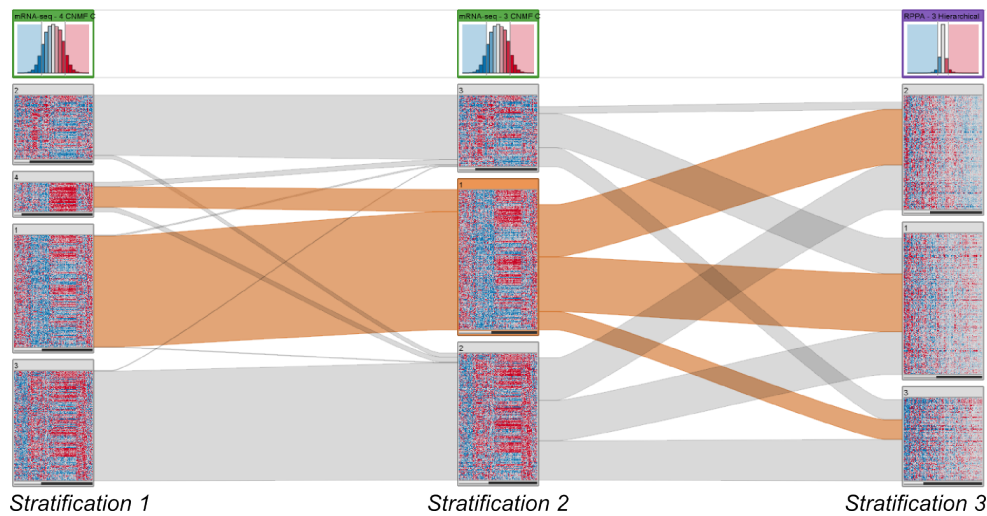


Figure 3.1.: The width of the bands denotes the overlap between subsets of adjacent stratifications. While there is a strong correlation between stratifications 1 and 2, stratifications 2 and 3 are much more dissimilar. The subsets in stratifications 1 and 2 are almost identical, except for two subsets (clusters 1 and 4) in stratification 1 that are merged into a single larger one (cluster 1) in stratification 2. In contrast, the bands between stratification 2 and 3 fan out almost equally, which is an indicator for weak correlation.

Figure 3.1 illustrates the basic principle at a glance. Data subsets are represented as vertical blocks, connected by bands. Each column consists of a summary block at the top and individual blocks for each cluster that form the stratification. The height of the blocks is scaled to be proportional to the number of items they contain, if such scaling can be applied to the corresponding visualization technique. A detailed description of the items is given in Figure 3.2, focusing on a biomedical use case in which patient stratifications are explored.

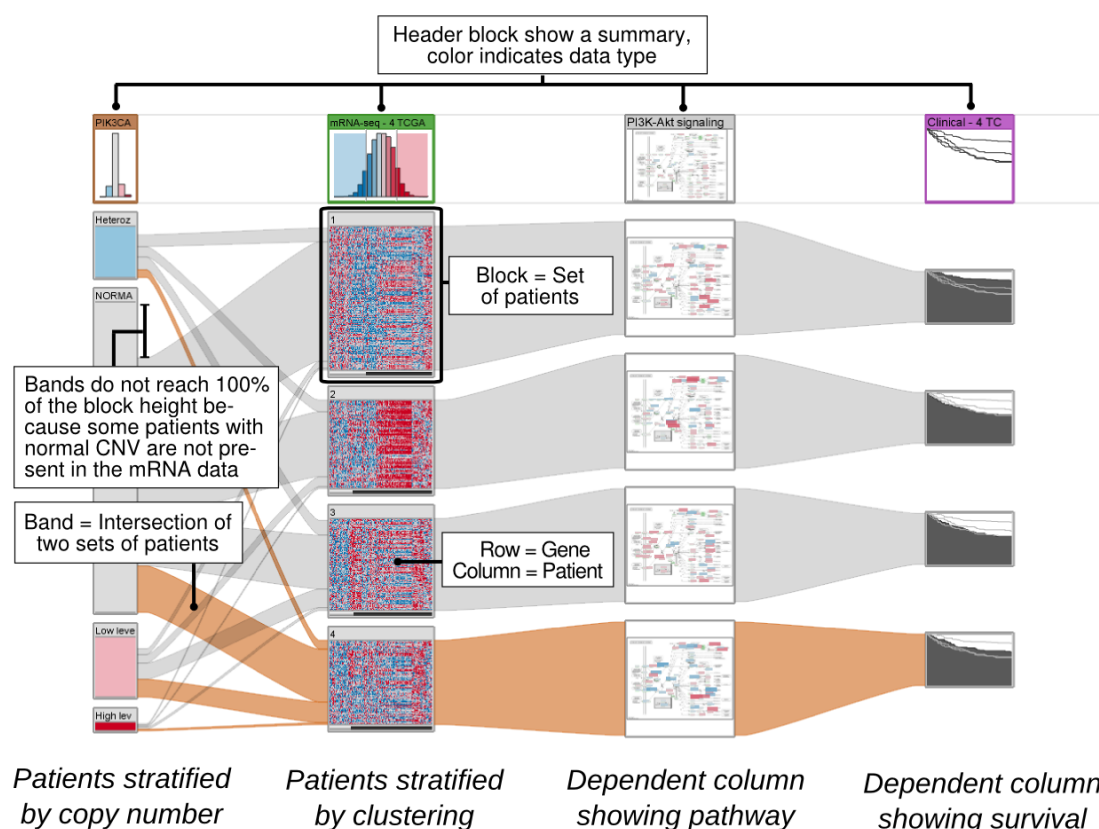


Figure 3.2.: StratomeX user interface. Stratifications are represented as columns, showing additional data per block. Bands indicate set relationships between two neighboring columns.

StratomeX allows analysts to flexible switch between different visualization techniques for each block (like a group of patients), depending on the data type, different visualizations are used to present the data associated with a block (Figure 3.3). While the header block at the top summarizes the data of all patients in a given column using a histogram, the visualization in each block below only represents the data of the patients from the corresponding patient subset. The height of the blocks is scaled to be proportional to the number of patients they contain, if such scaling can be applied to the corresponding visualization technique. Column 3 and 4 are *dependent columns*, meaning that they use the same stratification as the column that they depend on. In this case, the third and fourth column use the stratification of the second column, but apply the stratification to a different dataset. In the third column, the average mRNA expression of all four groups from the second column is color-coded onto the KEGG PI3K-Akt signaling pathway (hsa04151). The fourth column shows survival data using the same stratification as the second column.

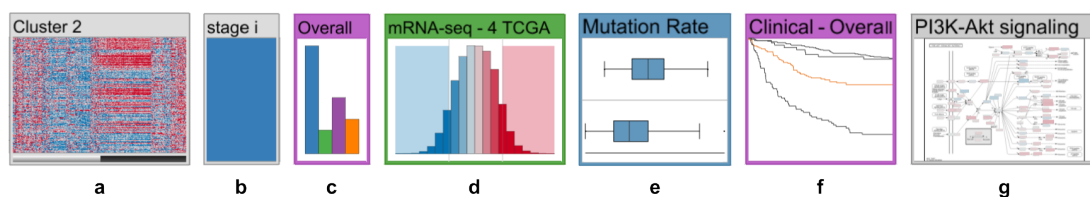


Figure 3.3.: Various visualization techniques are provided for visualizing the data associated with a block: (a) heatmaps for tabular data, (b) uniformly colored blocks for categorical data, (c) histograms for tabular data, (d) histograms for categorical data, (e) box plots for numerical data, (f) Kaplan-Meier plots, and (g) pathways with average values mapped onto the nodes. The height of the visualizations in (a) and (b) is scaled proportionally to the number of patients they represent, while the height of the other visualizations is constant

In summary, StratomeX is a flexible and powerful visual exploration technique for exploring complex set relationships along with data. Through the intuitive relation of band and block widths to the number of their contained items, analysts can explore and relate complex stratifications effectively.

3.2. Multi-Attribute Ranking

The purpose of the Ranking stage is to order, prioritize, and rank data subset collections. A wide variety of visualization techniques have been developed for, or have been applied to show, ranked data. The simplest form is a spreadsheet, in which the item is shown along with the value. A detailed discussion of the design of spreadsheets was published by Few [Few12]. Established general purpose tools such as *Microsoft Excel* are feature-rich and well known by many users. An early implementation of using the visual attribute length to encode the score of an item is the *table lens* technique [RC94], which embeds bars within a spreadsheet. Besides being one of the first focus+context techniques, it allows users to set multiple focus areas and also supports categorical values. As in a spreadsheet, users can sort the table items according to an arbitrary column. The rank-by-feature approach by Seo and Shneiderman [SS05] uses ordered bars to present a ranked list together with a score. The ranking suggests potentially interesting features between two variables. The scores are calculated according to criteria such as correlation and uniformity. In addition, the authors propose the rank-by-feature prism, a heat map that shows the feature scores between multiple variables. However, as the scores are based only on a single attribute, the rank-by-feature system does not address multi-attribute rankings. The

RankExplorer system by Shi et al. [SCL⁺12] uses stacked graphs [BW08] with augmented color bars and glyphs to compare rankings over time. While the system effectively let user compare rankings, it can only incorporate the information about the cause of the rank based on multiple attributes by showing details on demand in a coordinated view fashion. Sawant and Healey [SH08] visualize multi-dimensional query results in a space-filling spiral. Items from the query result are ordered by a single attribute and placed on a spiral. A glyph representation is used for encoding the attributes of the items. By using animation, the visualization can morph between different query results, highlighting the similarities and differences. Here, the ranking is based only on a single attribute. Recent work by Behrisch et al. [BDS⁺13] addresses the comparison of multiple rankings using a small-multiple approach in combination with a radial node-link representation; however, it is not designed to encode the cause of the rankings. The work by Kidwell et al. [KLC08] focuses on the comparison of a large set of incomplete and partial rankings, for example, user-created movie rankings. The similarity of rankings is calculated and visualized using multi-dimensional scaling and heat maps. While their approach gives a good overview of similarities between a large number of rankings with many items, an in-depth comparison of rankings is not possible. As trees can be ranked, tree comparison techniques, such as *Tree Juxtaposer* [MGT⁺03], can also be used to compare rankings. However, encoding multiple attributes using trees is problematic.

3.3. Multi-Dimensional Data Exploration

The Exploration stage in the center of the SPARE model represents explorations tools and techniques that support analysts in relating data subsets with each other. As defined in Section 1.1, a subset is a mathematical subset of the dimensions and rows in a heterogeneous dataset. A subset of dimensions in a heterogeneous dataset is also known as a subspace of the dimensions. Chapter 6 introduces the *Domino* visualization technique as one possible technique for this purpose. Domino is designed for multiple heterogeneous, high-dimensional datasets and relationships between individual dimensions of them. These relationships can also be interpreted as set relationships. Therefore, the relevant body of related work comprises visualization techniques for both sets and high-dimensional heterogeneous data. As Domino is a meta-visualization technique that enables users to create new visualizations which are interlinked, the body of related work includes multiple coordinated view systems in general, and integrated views systems that explicitly represent relationships between linked visualizations in particular.

Set and Subset Relationships Visualizing sets and their relationships is a problem frequently encountered in many domains. The most widely used set visualization techniques are Venn and Euler diagrams, which, however, do not scale beyond very small numbers of sets. Consequently, the visualization of relationships between sets has been, and continues to be, an active area of research. In a recent survey paper [AMA⁺14], Alsallakh et al. reviewed and classified existing set visualization techniques based on three classes of tasks: element-related, set-related, and attribute-related tasks. Most techniques that address attribute-related tasks can handle one or a few attributes. Domino, however, is designed to deal with relationships between tabular datasets in which each of potentially thousands of dimensions could be seen as a separate attribute.

A technique that effectively compares groups of sets (categories) with each other is parallel sets [KBH06]. Parallel sets arranges the (non-overlapping) sets of a group in a column and compares them to adjacent groups of sets using bands. The width of a band encodes the elements shared between two sets. In previous work, we introduced techniques that use this parallel sets metaphor for comparing relationships of partitioned datasets and embed the tabular data from which the partitions were derived, thus visualizing both, set relationships and attributes efficiently. A common example is clustering performed on a multi-dimensional dataset (e.g., gene expression data) where items (*genes*) are partitioned according to similarities in the data. Analysts are usually not only interested in the partitioning of the items, but also in the actual data associated with the items. The *Matchmaker* technique [LSP⁺10] supports this task by juxtaposing multiple partitioned tabular datasets, represented as heatmaps, and connecting the groups with bands and the items with individual splines. This column-based approach enables analysts to compare different partitions of the same dataset (e.g., the results of different clustering algorithms) or to explore relationships among multiple partitioned datasets containing the same items (e.g., multiple clustered expression datasets from different groups of patients containing the same genes). *VisBricks* [LSS⁺11] generalized this approach by allowing users to switch between various visualization techniques independently for each subset – a concept known as multiform visualization [LSS⁺11, Rob07]. Analysts can, for instance, choose parallel coordinates as the representation for individual groups to perform filtering tasks and a heatmap to see the overall pattern. *StratomeX* [LSS⁺12] as described before introduced subset visualization for multiple heterogeneous datasets.

All three techniques—Matchmaker, VisBricks, and StratomeX—show the relationships between multiple partitioned sets together with the multi-dimensional datasets on which the partitions are defined. However, due to the column-based arrangement of the tabular datasets, the analyst is limited to relationships of one type. For instance, if two clustered gene expression tables are positioned side by side (*genes* x *patients*), the analyst must choose to visualize either gene or patient relationships, depending on the arrangement of

the datasets. The free arrangement of subsets in Domino removes this restriction.

Another limitation of these techniques is that they require partitioned (categorical) datasets as columns, while Domino allows analysts to mix partitioned and numerical data. While our previous work focused on relationships between the partitioned sets, drawn as bands connecting the columns, Domino is able to show relationships between datasets at different levels of granularity: at the level of items, groups of items, or whole datasets. Finally, our earlier techniques do not allow users to extract and manipulate subsets, which is a central aspect of Domino.

In recent work we described *Furby* [SGG⁺14], a method for visualizing biclustering results in a force-directed layout. Bands between the clusters represent their overlaps in both rows and columns. However, similar to the approaches discussed above, Furby only encodes the overlap between whole biclusters and does not enable users to see individual relationships between shared items (rows/columns). Furthermore, Furby focuses on the overlap between matrix subsets, preventing users from including partitioned and numerical subsets in the sense-making process. Furby also does not address the extraction and manipulation of subsets.

Multiple Coordinated and Integrated Views An alternative to the multiform approaches described above are standard multiple coordinated views (MCV) [Rob07], where each subset is visualized as a separate view. Using linking & brushing, users can then explore the relationships between the subsets. In this sense, *Juxtaposition* [GAW⁺11] is used for comparing multiple subsets and their relationships. MCV is also applicable to scenarios that include multiple datasets connected by different item types. However, as MCV systems keep relationships between linked views [JE12] only implicitly, the approach has a major limitation: it requires users to actively select or filter items in order to see relationships between the subsets. Further, as the selected items from one view are simultaneously highlighted in all other views, users can generally only see that the item is part of the selection, but cannot determine which item in one view corresponds to which item in the other views if multiple items are selected.

Instead of implicitly linking items in a MCV setup, explicit links can be drawn to connect items across the views—an approach Javed and Elmqvist call integrated views [JE12] and a variant of *Explicit encoding* as defined by Gleicher et al. [GAW⁺11]. Examples of integrated views are semantic substrates [SA06], the VisLink method [CC07], or context-preserving visual links [SWL⁺11]. Note that the multiform approaches described earlier in this section can also be seen as integrated views. While explicit links remove the need for interaction to see relationships, it comes at the cost of added visual clutter. In Domino we also make use of explicitly connected multiform visualizations; however, we reduce

visual clutter by representing the relationships at various levels of granularity. In addition to representing relationships between individual items as single lines, Domino aggregates lines from the same group of a partition to bands.

Subspace Comparison Detecting patterns in high-dimensional datasets is possible on multiple levels. Domino focus on patterns between individual dimensions by visualizing their relationships in various ways. Another approach is exploring combinations of dimensions, i.e., subspaces and comparing them with each other. Hund et al. [HBS⁺16] propose a system for exploring subspaces of patient groups. Anand et al. [AWD12] use scores based on random projections of dimensions to guide users to interesting data subspaces. In Domino a subspace can be created by combining multiple dimensions into a combined block. Therefore, subspaces can be compared by relating patterns contained in combined blocks with each other.

Meta-Visualization Techniques The approaches most closely related to Domino are *Flexible Linked Axes (FLINA)* [CvW11] by Claessen and van Wijk and *ConnectedCharts* [VM12] by Viau and McGuffin. Domino is akin to FLINA and ConnectedCharts, as it is also a meta-visualization for creating advanced visualization setups. However, due to its holistic conceptual approach that integrates one-dimensional numerical, categorical, and tabular data by connecting them at multiple levels of granularity, Domino can be applied to a broader spectrum of tasks.

FLINA [CvW11] is a meta-visualization approach for creating axis-based visualization techniques, whose axis relationships can be described using the ARGOI formalism. Although it is a general and powerful concept, FLINA is restricted to showing item relationships between one-dimensional numerical data represented as 1D scatterplots (axes). ConnectedCharts [VM12] are related to FLINA in the sense that they also explicitly draw item-based relationships between visualizations. However, ConnectedCharts and Domino conceptually go beyond FLINA, as an axis-based representation is only one of many possible visualization techniques for displaying a subset. Another important difference from FLINA is that ConnectedCharts and Domino include one-dimensional and tabular datasets. While FLINA and ConnectedCharts are both limited to item-based relationships between the visualizations, Domino visually links them also on coarser abstractions, making it possible to show relationships between single items, between groups of categorical data, or between whole datasets. Another difference between the Domino technique from both FLINA and ConnectedCharts is its ability to let users not only explore relationships between subsets, but also refine existing and extract new subsets, which provides a great deal of flexibility in exploratory analysis scenarios.

3.4. Presentation and Storytelling

Chapter 7 introduces the CLUE model, reassembling the right side of the SPARE model. CLUE closes the gap between exploration and presentation using provenance data. Since our model is independent of the exploratory visualization techniques employed, we limit our discussion of related work for the CLUE model to the capturing and use of provenance data, presentation and communication in visualization, and visual storytelling techniques.

3.4.1. Provenance

In the context of our work, provenance of the state of a visual exploration process refers to all information and actions that lead to it. The provenance of the visual analysis as a whole is comprised of the provenance of all states that were visited during the exploration.

Ragan et al. [RESC16] have recently characterized provenance in visualization and data analysis according to the *type* and *purpose* of the collected provenance information. The type of provenance information includes *data* (i.e., the history of changes to the data), *visualization* (i.e., the history of changes to views), *interaction* (i.e., the interaction history), *insight* (i.e., the history of outcomes and findings), and *rationale*, the history of reasoning. Our prototype implementation currently captures all of this information. Data and interaction provenance are captured automatically, while insight and rationale are captured through the externalization of a thought process by the user. CLUE enables this, for instance via annotations and bookmarks.

Provenance information is used for several purposes [RESC16]: for *recalling* the current state during the analysis, for *recovering actions* (undo and redo), for *replicating* (reproducing) steps of the analysis, for *collaborative communication* (i.e., for sharing the analysis process with others), for *presenting* the analysis, and for *meta-analysis* (i.e., for reviewing the analysis process).

VisTrails [BCS⁺05], for example, collects and visualizes provenance data for computational and visualization workflows for large-scale scientific data analysis. Users of *VisTrails* interactively model a workflow that produces, for instance, a visualization; the process of creating this workflow is tracked as provenance data. In CLUE the focus is not on modelling a specific artifact with a goal in mind but rather automatically capture the provenance of an interactive visual exploration process that may lead to discoveries that are later being told using integrated storytelling approaches.

The works by Heer et al. [HMSA08] and Kreuseler et al. [KNS04] discuss a concept for

visual histories (i.e., provenance of a visual exploration) including types, operations, management, and aggregation and provide a prototypical implementation. However, in both cases, provenance data is used for exploration only and not to address storytelling aspects. Heer et al. pointed to storytelling based on provenance as future work.

Action recovery (undo, redo) is commonly integrated into software applications. Most tools, however, do not visualize the history of actions and rely on a linear undo-redo path, which makes recovery from analysis branches impossible. Exceptions to this implicit approach are integrated in *Small Multiples*, *Large Singles* [vdEvW13], *Stack’n’Flip* [SSL⁺12], and *GraphTrail* [DHRL⁺12]. In the former two, the history of the current artifact is explicitly shown at the bottom and implicitly through the strict left-to-right workflow. GraphTrail also supports branches in the history: It explicitly visualizes how a plot is derived from previous ones using basic data operations. However, only a fraction of the provenance of the visual exploration is being captured by focusing on the data operations leaving out the parameters of the visualizations, etc.

In regard to provenance, the paper by Shrinivasan and van Wijk [SW08] is most closely related to CLUE. The authors proposed a technique that integrates three views: Data, Navigation, and Knowledge. The data view contains the actual visualization of the data. The navigation view shows the exploration process in a tree (i.e., the provenance). Using the knowledge view, users can capture and relate annotations to document findings, assumptions, and hypotheses, and link them to specific states in their exploration for justification. The knowledge view in combination with the navigation view is then used to communicate findings. In contrast to their work, the CLUE model also covers aspects of storytelling: by enabling authoring, we allow users to produce concise and effective stories based on the original exploration. This linear narrative approach is closely related to the traditional workflow of publishing results with the additional benefit of having a back-link to the real exploration at all time.

3.4.2. Storytelling

Kosara and MacKinlay [KM13] highlighted the importance of visual data stories for visualization research. They defined a story as “an ordered sequence of steps, each of which can contain words, images, visualizations, video, or any combination thereof”. They further stated that “stories can thus serve as part of a finding’s provenance, similar to an event’s narrated history”. Different approaches can be used for telling a story. Stories are mostly told individually and linearly, but management of multiple stories and branching have also been proposed [LHV12]. The degree to which users can influence how the story is being told may vary from automated replay to crafting their own story.

In CLUE, we apply a narrative approach storytelling inspired by the work of Figueiras [Fig14a, Fig14b]. Figueiras discussed how to include narrative storytelling elements such as annotations and temporal structure in existing visualizations to enrich user involvement and understanding through story flow. Similarly, the authors of *VisJockey* [KSJ⁺14] noted that when integrating interactive visualizations into data-driven stories, user guidance how to interpret the visualizations are lacking by default. Therefore, the authors proposed the VisJockey technique that enables readers to easily access the author’s intended view through supplementing the visualization with highlights, annotation and animation. In *Tableau* ¹, storytelling features are integrated using an annotated stepper interface. This enables users to navigate through a series of interactive dashboards. These works have in common that they are dealing with existing visualizations and insights, purely focusing on the authoring and presentation state, yet neglecting the underlying process of how the insights were discovered.

Ellipsis [SH14] is a domain-specific language for creating visual data stories based on existing visualizations. Journalists can combine visualizations, define triggers, states, annotations, and transitions via a programming interface or a visual editor. This allows them to define a wide range of story types, including linear, non-linear, interactive, automatic, and stepped stories. However, *Ellipsis* is only concerned with creating scripted stories, and does not utilize the visual data exploration process that leads to an insight.

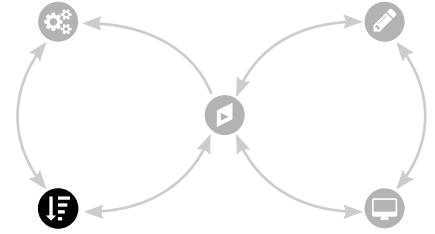
With regards to storytelling, the work most closely related to CLUE is by Wohlfart and Hauser [WH07]. Their technique allows users to record interactions with a volume visualization, modify this recording, and annotate it in order to tell a story. In their work, however, the recording or capturing has to be actively triggered, and only a linear story can be captured. When users press a record button, they typically already know what they want to show. CLUE, in contrast, captures all actions and exploration paths, allowing the user to extract or base their story on the provenance of the analysis.

Storytelling approaches prompting user involvement through play and pause techniques were explored by Pongnumkul et al. [PDL⁺11] to support navigation in step-by-step tutorials. Adobe Labs ² provides a Photoshop plugin to create such tutorials: users can first record actions and then author them.

In summary, existing storytelling tools focus on how to tell a story, but rarely base the story on provenance data. The CLUE model solves this by providing links between points in the story to corresponding states in the exploration. This allows users switch freely and easily between exploration and presentation.

¹<http://www.tableau.com/>

²<http://labs.adobe.com/technologies/tutorialbuilder/>



4 | LineUp

Visual Analysis of Multi-Attribute Rankings

Contents

4.1. Introduction	30
4.2. Requirement Analysis	31
4.3. Ranking Design Space and Related Work	34
4.4. Multi-Attribute Ranking Visualization Technique	38
4.5. Use Cases	48
4.6. Evaluation	51
4.7. Conclusion	54

This chapter describes *LineUp* a visualization technique for multi-attribute rankings. Rankings are a popular and universal approach to structuring otherwise unorganized collections of items by computing a rank for each item based on the value of one or more of its attributes. This allows us, for example, to prioritize data subsets based on a computed score as presented in the SPARE model in Section 1.2. While the visualization of a ranking itself is straightforward, its interpretation is not, because the rank of an item represents only a summary of a potentially complicated relationship between its attributes and those of the other items. Driven by a requirement analysis the visualization technique itself is introduced with all its visual encodings. The practical applicability is shown using two general usage scenarios. In addition, we conducted a qualitative user study testing the visual encodings that will be discussed in detail. More information about LineUp can be found at <http://lineup.caleydo.org> including links to a JavaScript library ¹ and public instance ².

¹<http://github.com/Caleydo/lineupjs>

²<http://lineup.caleydoapp.org>

4.1. Introduction

We encounter ranked lists on a regular basis in our daily lives. From the “top at the box office” list for movies to “New York Times Bestsellers”, ranked lists are omnipresent in the media. Rankings have the important function of helping us to navigate content and provide guidance as to what is considered “good”, “popular”, “high quality”, and so on. They fulfill the need to filter content to obtain a set that is likely to be interesting but still manageable.

Some rankings are completely subjective, such as personal lists of favorite books, while others are based on objective measurements. Rankings can be based either on a single attribute, such as the number of copies sold to rank books for a bestseller list, or on multiple attributes, such as price, miles-per-gallon, and power to determine a ranking of affordable, energy-efficient cars. Multi-attribute rankings are ubiquitous and diverse. Popular examples include university rankings, rankings of food products by their nutrient content, rankings of computer hardware, and most livable city rankings.

When rankings are based on a single attribute or are completely subjective, their display is trivial and does not require elaborate visualization techniques. If a ranking, however, is based on multiple attributes, how these attributes contribute to the rank and how changes in one or more attributes influence the ranking is not straightforward to understand. In order to interpret, modify, and compare such rankings, we need advanced visual tools.

When interpreting a ranking, we might want to know why an item has a lower or a higher rank than others. For the aforementioned university rankings, for example, it might be interesting to analyze why a particular university is ranked lower than its immediate competitors. It could either be that the university scores lower across all attributes or that a single shortcoming causes the lower rank.

Another crucial aspect in multi-attribute rankings is how to make completely different types of attributes comparable to produce a combined ranking. This requires mapping and normalizing heterogeneous attributes and then assigning weights to compute a combined score. A student trying to decide which schools to apply to might wish to customize the weights of public university rankings, for example, to put more emphasis on the quality of education and student/faculty ratio than on research output. Similarly, a scientist ranking genes by mutation frequency might want to try to use a logarithmic instead of a linear function to map an attribute.

Another important issue is the comparison of multiple rankings of the same items. Several publications, for example, release annual university rankings, often with significantly

different results. A prospective student might want to compare them to see if certain universities receive high marks across all rankings or if their ranks change considerably. Also, university officials might want to explore how the rank of their own university has changed over time.

Finally, if we can influence the attributes of one or more items in the ranking, we might want to explore the effect of changes in attribute values. For example, a university might want to find out whether it should reduce the student/faculty ratio by 3% or increase its research output by 5% in order to fare better in future rankings. If costs and benefits can be associated with these changes, such explorations can have an immediate impact on strategic planning.

Interactive visualization is ideally suited to tailoring multi-attribute rankings to the needs of individuals facing the aforementioned challenges. However, current approaches are largely static or limited, as discussed in our review of related work. In this chapter we describe a new technique that addresses the limitations of existing methods and is motivated by **a comprehensive analysis of requirements of multi-attribute rankings considering various domains**. Based on this analysis, we describe, **the design and implementation of LineUp, a visual analysis technique for creating, refining, and exploring rankings based on complex combinations of attributes**. We demonstrate the application of LineUp in two use cases in which we explore and analyze university rankings and nutrition data.

We evaluate LineUp in a qualitative study that demonstrates the utility of our approach. The evaluation shows that users are able to solve complex ranking tasks in a short period of time.

4.2. Requirement Analysis

We identified the following requirements based on research on the types and applications of ranked lists, as well as interviews and feedback from domain experts in molecular biology, our initial target group for the application. We soon found, however, that our approach is much more generalizable, and thus included a wider set of considerations beyond expert use in a scientific domain. We also followed several iterations of the *nested model for visualization design and validation* [Mun09] and looped through the four nested layers to refine our requirements (i.e., *domain problem characterization*). We concluded our iterations with the following set of requirements:

R 1: Encode rank Users of the visualization should be able to quickly grasp the ranks of the individual items. Tied ranks should be supported.

R II: Encode cause of rank In order to understand how the ranks are determined, users must be able to evaluate the overall item scores from which the ranking is derived and how they relate to each other. In many cases, scores are not uniformly distributed in the ranked list. For example, the top five items might have a similar score, while the gap to the sixth item could be much bigger. Depending on the application, the first five items might thus be much more relevant than the sixth. To achieve this, users should see the distribution of overall item scores and their relative difference between items and also be able to retrieve exact numeric values of the scores. If item scores are based on combinations of multiple attribute scores (see R III), the contribution of individual attributes to the overall item score should also be shown.

R III: Support multiple attributes To support rankings based on multiple attributes, users must be able to combine multiple attributes to produce a single all-encompassing ranking. It is also important that these combinations are salient. To make multiple attributes comparable, they must be normalized, as described in R V. In the simplest case, users want to combine numerical attributes by summing their scores. This combined sum of individual attribute scores then determines the ranking. However, more complex combinations, including weights for individual attributes and logical combinations of attributes, are helpful for advanced tasks (see Section 4.4.2).

R IV: Support filtering Users might want to exclude items from a ranking for various reasons. For example, when ranking cars, they might want to exclude those they cannot afford. Hence, users must be able to filter the items to identify a subset that supports their task. Filters must be applicable to numerical attributes as ranges, nominal attributes as subsets, and text attributes as (partial) string matches.

R V: Enable flexible mapping of attribute values to scores Attributes can be of different types (e.g., numerical, ordered categorical), scales (e.g., between 0 and 1 or unbounded) and semantics (e.g., both low and high values can be of interest). The ranking visualization must allow users to flexibly normalize attributes, i.e., map them to normalized scores. For example, when using a normalization to the unit interval $[0, 1]$, 1 is considered “of interest” and 0 “of no interest”. While numerical attributes are often straightforward to normalize through linear scaling, other data types require additional user input or more sophisticated mapping functions.

Numerical attributes can have *scales* with arbitrary bounds, for instance, from 0 to 1 or -1 to 1. They might also have no well-defined bounds at all. However, for a static and known dataset the bounds can be inferred from the data. For instance, while there is no upper limit for the number of citizens in a country, an upper bound can be inferred by using the number of citizens in the largest country as bound. In addition, values in a range can have different meanings. For example, if the attribute

is a p -value expressing statistical significance, it ranges from 0 to 1, where 0 is the “best” and 1 the “worst”. In other cases, such as log-ratios, where attribute values range from -1 to 1 , 0 could be the “worst” and the extrema -1 and $+1$ the “best”.

Additionally, users might be interested in knowing the score of an attribute without the attribute actually influencing the ranking, thereby providing contextual information.

R VI: Adapt scalability to the task While it is feasible to convey large quantities of ranked items using visualization, there is a trade-off between level of detail (LoD) and scalability. Where to make that trade-off depends largely on the given task. In some tasks only the first few items might be relevant, while in others the focus is on the context and position of a specific item. Also, some tasks may be primarily concerned with how multi-attribute scores are composed, while in other tasks individual scores might be irrelevant. A ranking visualization technique must be designed either with a specific task in mind or aim at optimizing the trade-off between LoD and scalability.

R VII: Handle missing values As real-world data is often incomplete, a ranking visualization technique must be able to deal with missing values. Trivial solutions for handling missing values are to omit the items or to assign the lowest normalized score to these attributes. However, “downgrading” or omitting an item because of missing values might not be acceptable for certain tasks. A well-designed visualization technique should include methods to sensibly deal with missing values.

R VIII: Interactive refinement and visual feedback The ranking visualization should enable users to dynamically add and remove attributes, modify attribute combinations, and change the weights and mappings of attributes. To enable users to judge the effect of modifications, it is critical that such changes are immediately reflected in the visualization. However, as these changes can have profound influences on the ranking, it is essential that the visualization helps users to keep track of the changes.

R IX: Rank-driven attribute optimization Optimizing the ranking of an item is an important task. Instead of analyzing how the ranking changes upon modifications of the attribute values or the weights, it should be possible, for example, to optimize the settings (i.e., values and/or weights) to find the best possible ranking of a particular item. Identifying the sensitivity of attributes, i.e., how they are influencing the ranking, for example, for finding the minimum attribute value change needed to gain ranks, is another rank optimization example.

R X: Compare multiple rankings An interactive ranking visualization that fulfills R I - R IX is a powerful tool addressing many different tasks. However, in some situations users are interested in putting multiple rankings into context with each other. An example is the comparison of competing university ranking methodologies. Observing changes over time, for instance, investigating university rankings over the last 10 years, is another example that requires the comparison of multiple ranking results.

4.3. Ranking Design Space and Related Work

Due to the ubiquitous presence of rankings and their broad applicability, a wide variety of visualization techniques have been developed for, or have been applied to show, ranked data. Based on the requirements introduced in the previous section, we discuss the design space of visual encodings suitable for ranking visualization, as outlined in Figure 4.1, and as some specific ranking visualization techniques.

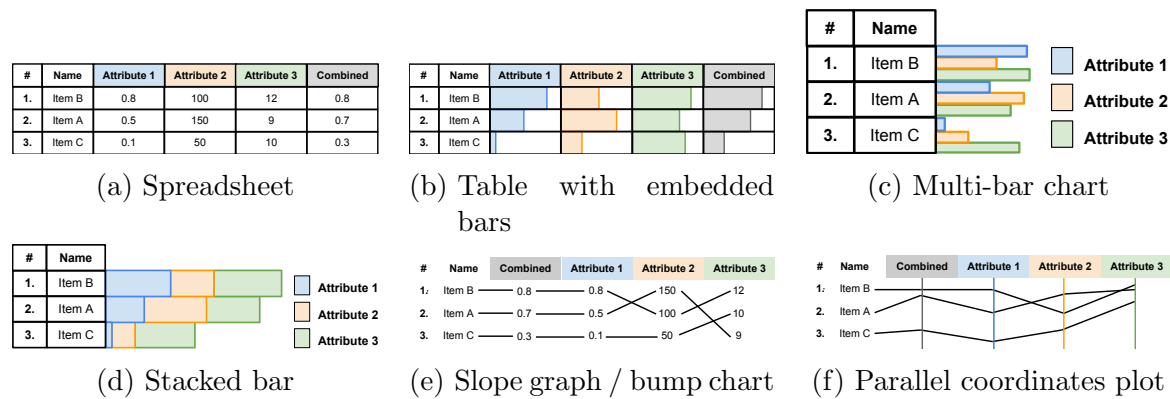


Figure 4.1.: Illustration of different ranking visualization techniques.

4.3.1. Spreadsheets

The most basic way to present a set of ordered items is a ranked list showing the rank together with a label identifying the item. While simple ranked lists allow users to see the rank of the item (R I), they do not convey any information about what led to the rank – which violates R II. It is trivial to extend a ranked list by multiple columns resulting in a table or spreadsheet addressing the multi-attribute requirement (R III), as shown in Figure 4.1a. A detailed discussion of the design of spreadsheets was published by Few [Few12].

Established general purpose tools such as *Microsoft Excel* are feature-rich and well known by many users. These tools provide scripting interfaces that can help to address requirements R I - R VII. While scripting provides a great deal of flexibility, it is typically only mastered by advanced users. The major drawback of spreadsheets, however, is that they lack interactive visualizations of the tabular data (R VIII). Also, spreadsheets typically lack the ability to compare multiple lists intuitively (see requirement R X). A comparison of lists can only be achieved by linking several spreadsheets in a multiple coordinated views fashion [Rob07], which is not supported by most spreadsheet applications. In such a setup, however, answering questions related to the essential requirements to encode the rank (R I) and to encode the cause of the rank (R II) is tedious and time-consuming, especially as the number of ranked items increases.

Reading numerical values in a spreadsheet and comprehending the data, however, is a cognitively demanding, error-prone, and tedious task. It is therefore more effective to communicate trends and relationships by encoding the values in a graphical representation using visual variables such as position, length, and color, etc. We discuss below how visual variables can be used to create visual representations that can cope with ranking data. In line with Ward et al. [WGK10], we divide the related work into techniques that are point-based, region-based, or line-based.

4.3.2. Point-Based Techniques

Using position as a visual variable is considered to be the most effective way of encoding quantitative data [Mac86]. Simple *scatterplots* can be used to compare two rankings (see R X). A scatterplot, however, can focus either only on communicating the rank itself (R I), by mapping the rank to interval scales, or on the cause of the rank (R II), by encoding the attribute value pairs in the position of the dots. While we can overcome the limitation of only comparing two rankings by using a *scatterplot matrix (SPLOM)*, neither scatterplots nor SPLOMs can deal with R III, the multi-attribute requirement, which makes them an inefficient solution for complex rankings.

4.3.3. Region-Based Techniques

According to various established sources [Mac86, CM84, Ber10], length is another very effective visual variable for encoding quantitative data. For representing ranked lists, simple bar charts, for instance, can show the value of multiple items for a single attribute. To make use of the preattentive processing capabilities in interpreting relative changes

in the length of bars (i.e., height), they are usually aligned to a common axis. Aligning the bars also redundantly uses position in addition to length. In cases where ranks are determined based on a single attribute, using bars to encode the attribute values is an effective way to communicate the cause of the rank, satisfying R II.

Bars can be used to encode multiple attributes (R III) in three different ways: by aligning bars for every attribute to a separate baseline, as shown in Figure 4.1b, by showing multiple bars per item (one for each attribute) on the same baseline, see Figure 4.1c, or by using stacked bar charts, see Figure 4.1d.

An early implementation of the first approach is the *table lens* technique [RC94], which embeds bars within a spreadsheet. Besides being one of the first focus+context techniques, it allows users to set multiple focus areas and also supports categorical values. As in a spreadsheet, users can sort the table items according to an arbitrary column. John et al. [JTS08] proposed an extension of the original table lens technique that adds two-tone pseudo coloring as well as hybrid clustering of items to reduce strong oscillation effects in the representation. The major benefit of using bar charts with multiple baselines is that it is easy to compare one attribute across items. In contrast, by switching to a layout that draws multiple bars per item side by side on the same baseline, the comparison across attributes for a single item is better supported, but comparing the bars across items becomes more difficult.

Stacked bar charts are appropriate when the purpose of the visualization is to present the totals (sum) of multiple item attributes, while also providing a rough overview of how the sum is composed [Few12]. Stacked bar charts are usually aligned to the baseline of the summary bar. However, shifting the baseline to any other attribute results in *diverging stacked bar charts*, see Figure 4.3b, which were discussed by Willard Brinton as early as 1939 [Bri39]. Changing the baseline makes it difficult to compare the sum between different stacked bars, but easier to relate the values of the aligned attribute. Diverging stacked bar charts are also known to be well suited to visualizing survey data that is based on rating scales, such as the Likert scale [RH11].

Compared to spreadsheets, bar-based techniques scale much better to a large number of items (R VI). While the minimum height of rows in spreadsheets is practically determined by the smallest font that a user can read, bars can be represented by a single-pixel line. It is even possible to go below the single pixel limit by using overplotting. This aggregation of multiple values to a single pixel introduces visual uncertainty [HLS⁺12]. While details such as outliers will be omitted, major trends will remain visible. Combined with additional measures such as a fish-eye feature, bar-based approaches are an effective technique for dealing with a large number of items.

4.3.4. Line-Based Techniques

Line-based techniques are also a widely used approach to visualizing rankings. In principle, lines can be used to connect the value of items across multiple attributes (R III) or to compare multiple rankings (R X). Although a wide array of line-based techniques exist [WGK10], only a few of them are able to also encode the rank of items (R I).

The first technique relevant in the context of ranking visualization are *slope graphs* [Tuf83, p.156]. According to Tufte, slope graphs allow users to track changes in the order of data items over time. The item values for every time point (attribute) are mapped onto an ordered numerical or interval scale. The scales are shown side by side, in an axis-like style without drawing them explicitly, and identical items are connected across time points. By judging differences in the slope, users are able to identify changes between the time points. Lines with slopes that are different to the others stand out. Note that slope graphs always use the same scale for all attributes shown. This makes it possible not only to interpret slope changes between two different time points, but also to relate changes across multiple or all time points. Although Tufte used the term slope graph only for visualizing time-dependent data, the technique can be applied equally to arbitrary multi-attribute data that uses the same scale.

Slope graphs that map ordered data to a scale using a unique spacing between items are referred to as *bump charts* [Tuf95, p.110]. Bump charts are specialized slope graphs that enable users to compare multiple rankings, fulfilling R X. The example in Figure 4.1e shows a bump chart where each column is sorted individually and the lines connect to the respective rank of each item. *Tableau*³, for example, uses bump charts to trace ranks over time. However, while bump charts show the rank of items (R I), they do not encode the cause of the rank (R II), which means the actual values of the attributes that define the ranking are lost.

A line-based multi-attribute visualization that mixes scales, semantics, and data types across the encoded attributes is a *parallel coordinates plot* [Ins85], as shown in Figure 4.1f. In contrast to a slope graph, a parallel coordinates plot uses the actual attribute values mapped to the axis instead of the rank of the attribute. While this is more general than slope graphs, users lose the ability to relate the slopes across different attributes. A thorough discussion of the differences between the aforementioned line-based techniques was published by Park [Par11a, Par11b].

Fernstad et al. [FSJ13], for instance, propose a parallel coordinates plot that shows one axis for each attribute with an extra axis showing the overall rank. While this addresses

³<http://www.tableau.com>

R I - R III, adding the possibility to compare multiple rankings (R X) is difficult. In theory, we could create parallel coordinates showing multiple axes that map the rank and add further axes to show the attributes that influence each of the rankings. However, in this case it would be difficult to make it clear to the user which rank axis belongs to which set of attribute axes.

4.4. Multi-Attribute Ranking Visualization Technique

LineUp is an interactive technique designed to create, visualize, and explore rankings of items based on a set of heterogeneous attributes. The visualization uses bar charts in various configurations. By default, we use stacked bar charts where the length of each bar represents the overall score of a given item. The vertical position of the bar in the bar chart encodes the rank of the item, with the most highly ranked item at the top. The basic design of the LineUp technique, as shown in Figure 4.2, is introduced in detail in Section 4.4.1. The components of the stacked bars encode the scores of attributes, which can be weighted individually. Combined scores can be created for sets of attributes using two different operations (see Section 4.4.2). Either the sum of the attribute scores is computed and visualized as stacked bars – a *serial combination* – or the maximum of the attribute scores in the set is determined and visualized as bars placed next to each other – a *parallel combination*. Such combinations can be nested arbitrarily.

Furthermore, LineUp can be used to create and compare multiple rankings of the same set of items (see Section 4.4.4). When comparing rankings, the individual rankings are lined up horizontally, and slope graphs are used to connect the items across rankings, as shown in Figure 4.8. The angle of the slope between two rankings represents the difference in ranks between two neighboring rankings.

Formally, rankings in LineUp are defined on a set of items $x_i \in X = \{x_1, \dots, x_m\}$ and a heterogeneous set of attribute vectors $\mathbf{a}_j \in A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ so that each item x_i is assigned a set of attribute values $A_i = \{a_{1i}, \dots, a_{ni}\}$. Since the attributes are heterogeneous, for instance, numerical on different scales or categorical, the user first needs to normalize the attribute vectors \mathbf{a}_j by mapping them onto a common scale. To achieve this, the user provides a mapping function $m_j(\mathbf{a}_j) = \mathbf{a}'_j \in A'$ with $m_j : a_{ji} \rightarrow [m_{j\min}, m_{j\max}]$ for each attribute with $0 \leq m_{j\min} \leq m_{j\max} \leq 1$. The values of $m_{j\min}$ and $m_{j\max}$ can be defined by the user. Throughout this chapter we refer to mapped attribute values a'_{ji} as *attribute scores* and to A' as the *mapped attributes*.

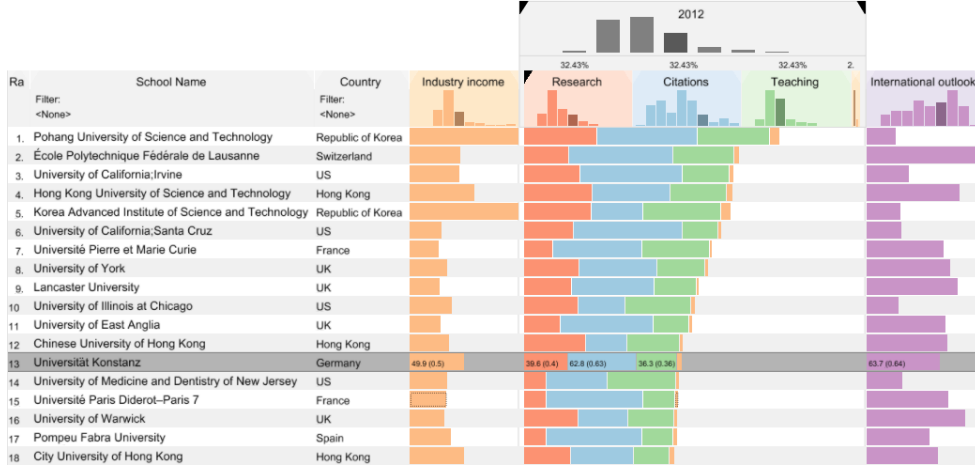


Figure 4.2.: A simple example demonstrating the basic design of the LineUp technique. The screenshot shows the top-ranked universities from the Times Higher Education Top 100 under 50 datasets. The first column shows the ranks of the universities, followed by their names and the categorical attribute *Country*. The list is sorted according to the combined attribute column containing four university performance attributes. Two numerical attribute columns which do not influence the ranking are also shown.

Additionally, the user may specify filters $f_{j_{\min}}$ and $f_{j_{\max}}$ on the original attribute values to remove items with attribute values a_{ji} outside the filter range $[f_{j_{\min}}, f_{j_{\max}}]$ from the ranking. The visualizations and interactions provided for data mapping and filtering are described in detail in Section 4.4.6.

To assign an item score $s_i \in \mathcal{S} \subset \mathbb{R}_0^+$ to each item x_i , the user interactively defines a scoring function s over the mapped attributes in A' through the LineUp user interface. The user selects a list $B = (\mathbf{a}'_q)$ of one or more attributes from A' with $1 \leq q \leq n$, where an attribute may be added more than once by cloning the attribute. The item score $s_B(x_i)$ over a list of mapped attributes B from A' is defined as

$$s_B(x_i) = \sum_{a'_{qi} \in B} w_{a'_q} a'_{qi} \quad | \quad 0 \leq w_{a'_q} \leq 1 \wedge \sum w_{a'_q} = 1,$$

where $w_{a'_q}$ are weights assigned to each instance of a mapped attribute \mathbf{a}' . Since the user may divide B into multiple lists B_l and combine, weight, and nest them arbitrarily, as discussed in Section 4.4.2, the final item score s_i is defined recursively over nested lists of mapped attributes as

$$s_i = s_B(x_i) = \begin{cases} \sum_{a'_{q_i} \in B} w_{a'_q} a'_{q_i} & | 0 \leq w_{a'_q} \leq 1 \wedge \sum w_{a'_q} = 1 \\ \sum_{B_l} w_{B_l} s_{B_l}(x_i) & | 0 \leq w_{B_l} \leq 1 \wedge \sum w_{B_l} = 1 \\ \max_{B_l} s_{B_l}(x_i). \end{cases}$$

The operators \sum and \max represent the sum (*serial combination*) and the maximum (*parallel combination*) of the item scores over a list of attribute scores, respectively. Users can interactively change the weights $w \in \mathbb{R}_0^+$ for each list of attributes by changing the width of the corresponding column header in LineUp.

LineUp determines the rank $r_i \in \mathbb{N}^+$ of an item x_i based on its item score s_i (which is equivalent to a'_{j_i} for cases in which ranks are based on a single attribute \mathbf{a}_j), with $\max(S) = 1$ so that $r_j - r_i = d \in \mathbb{N}^+$ if $s_j < s_i$, and there are exactly $d - 1$ other scores $s_k \in S$ with $s_j < s_k \leq s_i$. Ties are allowed since two or more items may have the same score. To resolve ties, the scoring method described above can be applied recursively to a tied set of items, for instance, using different attributes.

4.4.1. Basic Design and Interaction

LineUp is a multi-column representation where users can arbitrarily combine various types of columns into a single table visualization, as shown in Figure 4.2. The following column types are supported:

- **Rank columns** showing the ranks of items.
- **Textual attribute columns** for labels of items or nominal attributes. Text columns provide contextual information on the basis of which users can also search or filter items.
- **Categorical attribute columns** can be used in a similar fashion as textual attribute columns.
- **Numerical attribute columns** encoding numerical attribute scores as bars. In addition to the name of the attribute, the header can show the distribution of attribute scores on demand. When the user selects a particular item, the corresponding bin in the histogram will be highlighted.
- **Combined attribute columns** representing combinations of sets of numerical attributes. The process and visual encoding of combined columns is explained in Section 4.4.2.

In its simplest form, LineUp presents each attribute as a separate column where numerical columns use bars to represent their values. The ranking can be driven by any column, using sorting based on scores or on lexicographic order. Figure 4.2 also shows a combined attribute column labeled “2012” with four nested attributes. The decreasing lengths of the stacked bar charts indicate that this combined column drives the ranking, which is also shown using black corners at the top of its column header. In this example, the item labeled “Universität Konstanz” is selected. For selected items we show the original and the normalized attribute value (score) inside the bars if sufficient space is available. To simplify tracking of items across columns, rows are given alternating background colors.

4.4.2. Combining Attributes

A fundamental feature of our ranking visualization is the ability to flexibly combine multiple attributes, as described in requirement R III. LineUp supports the combination of attributes either in series or in parallel, as formally introduced earlier in this section. Both types of combinations are created by dragging the header of a (combined) attribute column onto the header of another (combined) column. Removing an attribute from a combined column is possible by using drag-and-drop or by triggering an explode operation that splits the combined column into its individual parts.

Serial Combination

In a serial combination the combined score is computed by a weighted sum of its individual attributes scores. The column header of such a combined column contains the histograms of all individual attributes that contribute to the combined score as well as a histogram showing the distribution of the combination result. While the combined scores are represented as stacked bars, the weight of an attribute is directly reflected by the width of its column header and histogram. Altering weights can be realized either by changing the width of a column using drag-and-drop or by double-clicking on the percentages shown above the histograms to specify the exact distribution of weights. While the former approach is particularly valuable for experimenting with different weights because the ranking will be updated interactively, the latter is useful to reproduce an exactly specified distribution, as demonstrated in the university ranking use case presented in Section 4.5.2.

Stacked bars allow users to perceive both the combined attribute score and the contribution of individual attribute scores to it. However, stacked bars complicate the comparison of individual attribute scores across multiple items, as only the first one is aligned to

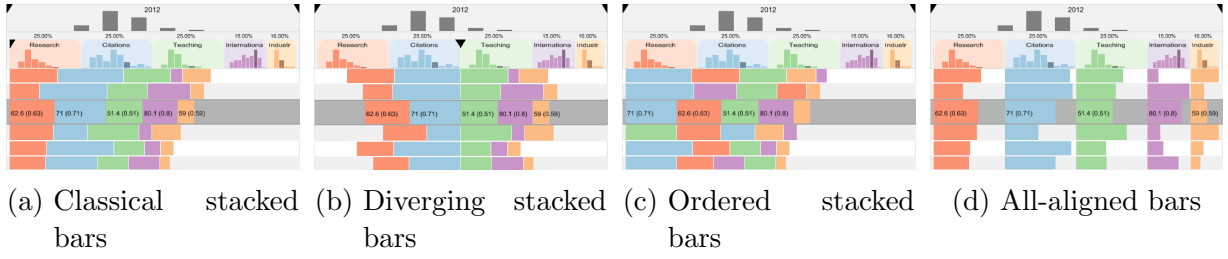


Figure 4.3.: Strategies for aligning serial combinations of attribute scores.

the baseline. Therefore, LineUp realizes four alignment strategies, which are shown in Figure 4.3. Besides classical stacked bars, diverging stacked bars, where the baseline can be set to an arbitrary attribute, are provided. The third strategy sorts the bars of each row in decreasing order, highlighting the attributes that contribute the most to the score of an item. The last strategy aligns every attribute by its own baseline, resulting in a regular table with embedded bars. These strategies can be toggled dynamically and use animated transitions for state changes.

Parallel Combination

In contrast to the serial combination that computes a combined score by adding up multiple weighted attribute values, the parallel combination is defined as the maximum of a set of attribute scores. Due to the limited vertical space, only the attribute with the largest score is shown as the bar for a given item. The attribute scores that do not contribute to the rank of the item are only shown when the item is selected. The corresponding bars are drawn on top of each other above the largest bar, as illustrated in Figure 4.4. In order to avoid small values overlapping bigger ones, the bars are sorted according to length.

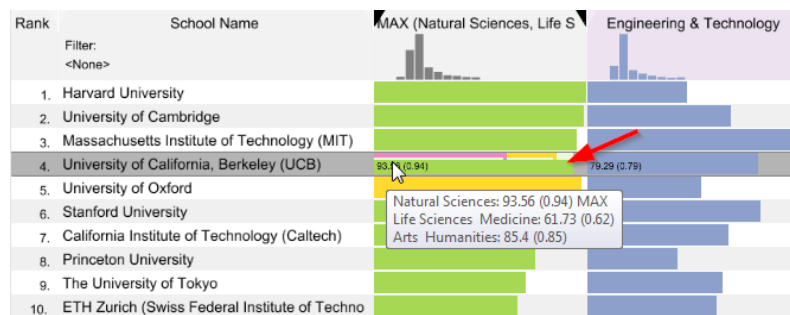


Figure 4.4.: Parallel combination of three attributes. Only the bar for the attribute with the largest score is shown for unselected items.

4.4.3. Rank Change Encoding

One of the major strengths of the proposed approach is that users receive immediate feedback when they interactively change weights, set filters, or create and refine attribute combinations. LineUp supports users in keeping track of rank changes by using a combination of animated transitions [HR07] and color coding. Animated transitions are an effective way to encode a small number of changes; however, when the number of changing items is very large and the trajectories of the animation paths cross each other, it is hard to follow the changes. Therefore, we additionally use color to encode rank changes, as demonstrated in Figure 4.5 (right). Items that move up in the ranking are visualized in green, whereas those that move down are shown in red. The rank change encoding is shown only for a limited time after a change has been triggered and is then faded out. The animation time and color intensity depend on the absolute rank change, which means the more ranks an item moves up or down, the longer and the more intense the highlighting. By providing interactive rank change and refinement capabilities combined with immediate visual feedback, LineUp is able to address requirement R VIII.

4.4.4. Comparison of Rankings

Encoding rank changes within a ranking visually is an essential feature to help users track individual items. However, animated changes and color coding are of limited assistance when analyzing differences between rankings. In order to address this problem, a more persistent visual representation is needed that allows users to evaluate the changes in detail. In fact, we want to enable users to compare different rankings of the same set of items, as formulated in R X. However, comparing rankings is not only important to support users in answering “What if?” questions when underlying attribute values change, but are also highly relevant for analyzing how multiple different attribute combinations or weight configurations compare to each other. An example, presented in the use case of university rankings (see Section 4.5.2), is the comparison of rankings over time.

We realize the comparison of rankings by lining up multiple rankings horizontally next to each other, each having its own item order, and connect identical items across the rankings with lines as in a slope graph. An example with two rankings and one with multiple rankings are given in Figures 4.5 and 4.8 respectively. The slope graph acts as a *rank separator*. This means that every attribute column that is separated by a slope graph has its own ranking. Also, changes in weights, mappings, filters, or attribute combinations only influence the order of items between the rank separators.

4.4.5. Scalability

A powerful ranking visualization needs to scale in the number of attributes and the number of items it can handle effectively, as formulated in the scalability requirement (R VI). Here we discuss our approaches for both.

Many Attributes

In order to handle dozens of attributes, in addition to providing scrollbars, we allow users to reduce the width of attribute columns by collapsing or compressing them on demand. *Collapsing* a column reduces its width to only a few pixels. As bars cannot effectively encode data in such a small space, we switch the visualization from a bar chart to a grayscale heat map representation (darker values indicate higher scores). Examples of collapsed columns are the three rightmost in Figure 4.8.

While collapsing can be applied to any column, *compressing* is applicable only to serial combinations of attribute columns. To save space, users can change the level of detail for the combined column to replace the stacked bar showing all individual scores with a single summary bar. An example of this is shown in the “2011” and “2010” columns in Figure 4.8.

To further increase scalability with respect to the number of attributes, we provide a *memo pad* [SKKS08], as shown at the bottom of Figure 4.8. The memo pad is an additional area for storing and managing attributes that are currently not of immediate interest but might become relevant later in the analysis. It can hold any kind of column that a user removes from the table, including full snapshots. Attributes can be removed completely from the dataset by dragging them to the trash can icon of the memo pad.

We assign colors to each attribute based on a carefully chosen qualitative color scheme and repeat colors when we exceed seven [Hea96]. However, this approach becomes increasingly problematic with a growing number of attributes. One option to address this problem is to use the same color for semantically related attributes, as illustrated in the use case in Section 4.5. For instance, when the goal is to rank food products, we use the same color for all vitamin-related attributes. However, whether this approach is useful depends on the specific scenario and the task. Therefore, the mapping between attributes and colors can be refined by the users.

Many Items

In order to make our technique useful in real-world scenarios, we need to cope with thousands of items. We use two strategies to achieve this: filtering and optimizing the visual representation. While filtering is straightforward, we also allow users to choose between uniform line spacing and a fish-eye selection mode [SSTR93]. Most users will be more familiar with uniform line spacing; which is, however, limited to showing only up to about 100 items at a time on a typical display. The fish-eye, in contrast, scales much better. The disadvantage is that changes in slope for comparison are less reliable due to distortion.

4.4.6. Data Mapping

Data mapping is the process of transforming numerical or ordered categorical attribute values to a normalized range between 0 and 1 (R/V) that can then be used to determine the length of the attribute score bars (1 corresponds to a full bar). By default, LineUp assumes a linear mapping and infers the bounds from the dataset so that no user interaction is required. To create more complex mappings, LineUp provides three approaches: choosing from a set of essential mapping functions (e.g., logarithmic or inversion), a visual data mapping editor that enables users to interactively create mappings, and a scripting interface to create sophisticated mappings. As non-linear mappings, inversions, etc. can have a profound impact on the ranking, we use a hatching pattern for all non-linear bars to communicate this.

In order to let users create mappings with different levels of complexity, the visual data mapping editor provides two options to interactively define them, as illustrated in Figure 4.6. All interactive changes in the mapping functions are immediately reflected in the visualization.

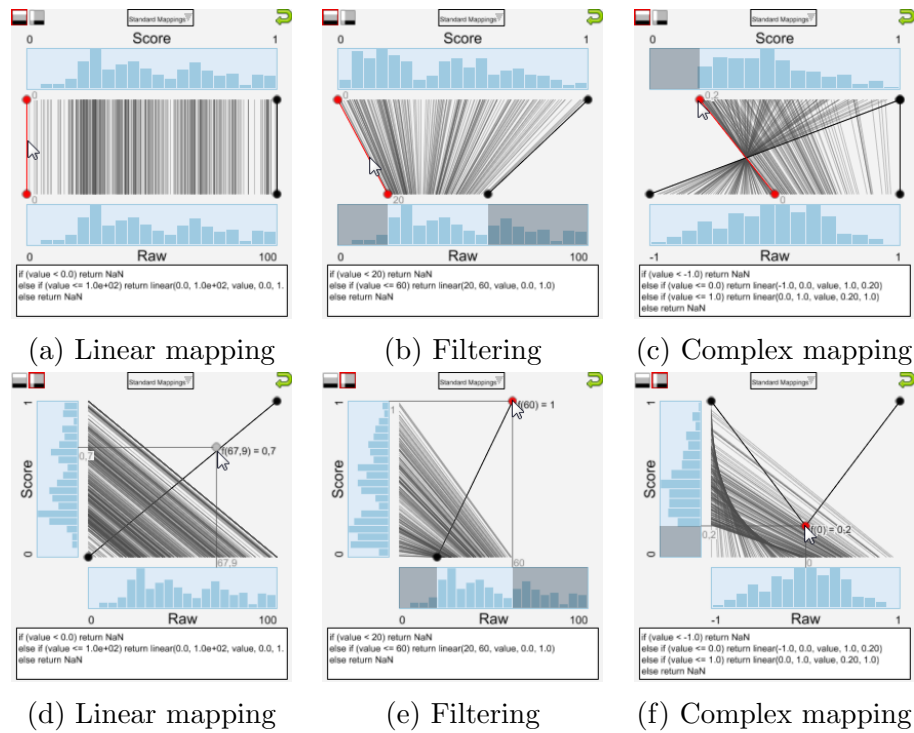


Figure 4.6.: Visual mapping editor for mapping attribute values to normalized scores. The parallel mapping editor used in (a)-(c) shows the distribution of values as well as the normalized scores as histograms on top of each other. Connection lines between the histograms make it easy to interpret the mapping. In (d)-(f) the layout of the mapping editor is changed to an orthogonal arrangement that resembles an actual mapping function. We show three mapping examples defined using the two different mapping editor layouts. (a) and (d) show the default case, where raw values are linearly mapped. In (b) and (e) raw values below 20 and above 60 are filtered. The remaining value range is spread between 0 and 1. In (c) and (f) the mapping is driven by three markers, to produce a mapping that emphasizes high and low values and at the same time filters low scores.

In the *parallel mapping editor* we show the histogram of the normalized attribute values above the original histogram of the raw data. Connection lines between the two histograms, drawn for every item, help the user to assess which attribute value maps to which score in the normalized range. By default, we apply a linear mapping that results in parallel lines between the histograms, as shown in Figure 4.6a. By dragging the minimum or maximum value markers in the histograms, users can filter items above or below a certain threshold,

as shown in Figure 4.6b. To flexibly create arbitrary mappings, mapping markers can be added, moved, and removed. Figure 4.6c shows a mapping scenario where the attribute values range from -1 to 1 but the scores are based on the absolute values. In addition, items with an attribute score of less than 0.2 are filtered.

The *orthogonal mapping editor* is an alternative view that uses a horizontal histogram of raw attribute values and a perpendicular vertical histogram of the normalized scores to visualize the mapping. This layout has the advantage that it can be interpreted just like a regular mapping function. Users can flexibly add, move, and remove support points to define the shape of the mapping function. We use linear interpolation to calculate the mappings between the user-defined support points. Figures 4.6d to 4.6f show the same examples that were used above to illustrate the parallel layout. Users can hover over any part of the mapping function to see the raw and normalized value pairs.

The visual data mapping editor also shows a formal representation of the mapping function that is always in sync with the interactive mapping editor. Clicking this representation opens the *JavaScript function editor* that can be used to define complex mapping functions, such as polynomial and logarithmic functions, which cannot easily be defined in the visual editors.

4.4.7. Missing Values

As real-world datasets are seldom complete, we need to deal with missing attribute values and encode them in the visualization. The only way to obtain a meaningful combined score based on multiple attributes where at least one has a missing value is to infer them. However, there is no general solution for inferring missing values that works for every situation. We currently apply standard methods such as calculating the mean and median; however, the integration of more complex algorithms is conceivable [Sch02]. Besides the computation of missing value replacements, their visualization is crucial. As inference of missing values introduces artificial data, it is important to make the user aware of this fact. In LineUp we encode inferred data with a dashed border inside the bars.

4.5. Use Cases

We demonstrate the technique in two use cases: (1) the nutrition content of food products based on a subset of the *USDA National Nutrient Database* [Nut13] and (2) ranking of universities using data from *Times Higher Education 100 Under 50 University Ranking* [Tim12] and the *QS World University Ranking* [Qua13].

4.5.1. Food Nutrition Data



Figure 4.7.: Example of a customized food nutrition ranking to identify healthier choices of breakfast cereals.

The first use case demonstrates how users can interactively create complex attribute combinations using the LineUp technique. Let us assume that John receives bad news from his doctor during his annual physical exam. In addition to his pre-existing high blood pressure, his cholesterol and blood sugar levels are elevated. The doctor advises John to increase his physical activity and improve his diet. John decides to systematically review his eating and drinking habits and begins with evaluating his usual breakfast. He loads a subset of the comprehensive food nutrition dataset from the *USDA National Nutrient Database* containing 19 nutrition facts (attributes) for each of about 8,200 food products (items) into LineUp. After loading the dataset, every attribute ends up in a separate column. As a first task, he filters the list to only include products in the category *breakfast cereals*. When he looks up his favorite breakfast cereals, he is shocked to find that they contain very high amounts of sugar and saturated fat. In order to find healthier choices, John searches for products that are high in *dietary fiber* and *protein* but low in *sugar*, *saturated fat*, and *sodium*. To rank the products according to these criteria, he creates a new serial attribute combination that assigns all attributes equal weight. In addition, as he is interested only in products that have “ready-to-eat” in their description, he applies

another filter. Since he wants low values of *sugar*, *saturated fat*, and *sodium* to receive a high score while high values should receive a low score, he uses the parallel mapping editor to invert the mapping function for these attributes. After looking at the top 20 items in the ranking, he realizes that none of the products matches his taste. He starts to slowly decrease the weight assigned to the *sugar* attribute, which means he reduces its impact on the overall ranking, and tracks the changes by observing the rank change color encoding and animations. He also uses the fish-eye to handle the large number of products when browsing the list until he finally finds his new breakfast cereal. Figure 4.7 shows the result of his analysis.

4.5.2. University Ranking

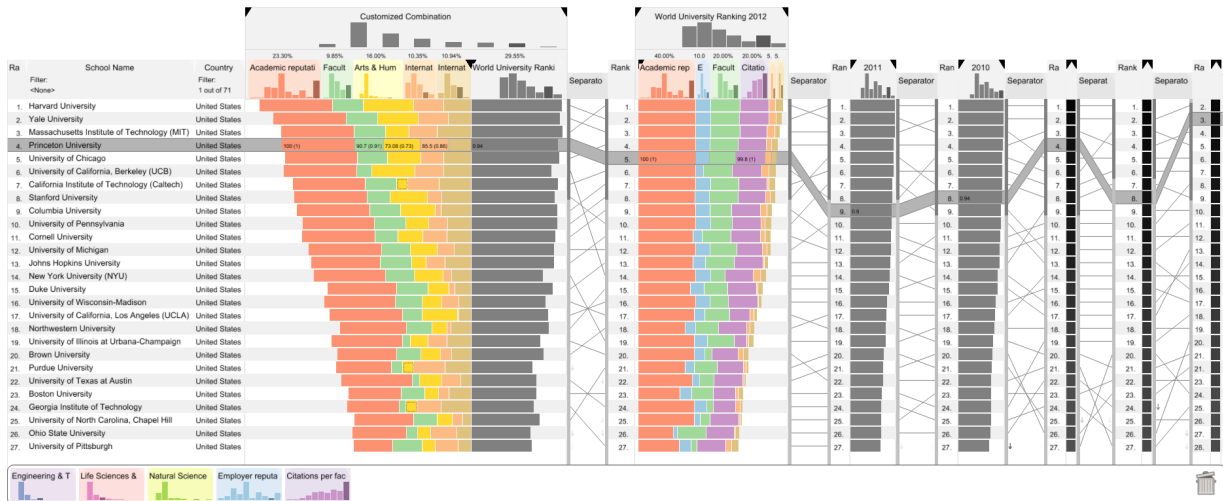


Figure 4.8.: LineUp showing a ranking of the top Universities according to the QS World University Ranking 2012 dataset with custom attributes and weights, compared to the official ranking.

In the second use case we demonstrate how Jane, a prospective undergraduate student, utilizes LineUp to find the best institutions to apply to. As a basis for the selection process, Jane chooses the well established *QS World University Ranking*. She starts by loading the annual rankings published from 2007 through 2012. As Jane does not want to leave the US, she adds a filter to the categorical *country* attribute to remove universities outside the US, and reviews the rankings for US institutions. By looking at the bar charts, she is able to see what factors contribute to the ranks and how they are weighted. The QS World University Ranking is based on six attributes: *academic reputation* (weighted

40%), *employer reputation* (10%), *faculty/student ratio* (20%), *citations* (20%), *international faculty ratio* (5%), and *international student ratio* (5%). Additionally, the authors publish performance data on five broad subject areas, such as *arts & humanities* and *life sciences*, which, however, do not influence the ranking. While university rankings try to capture the overall standing of institutions, Jane is a prospective undergraduate student and does not care much about research and citations but rather wants to emphasize teaching. Additionally, she wants to go to a university that has a renowned arts faculty and a strong international orientation, as documented by many exchange students and staff from abroad. To obtain the ranking that reflects her preferences, Jane wants to combine these attributes and adjust the weights accordingly. In order not to lose the original ranking, she takes a snapshot of the original attribute combination. In the new snapshot she removes *employer reputation* and *citations* from the combined score and adds *arts & humanities* to the weighted attributes. Next, she adjusts the weights by interactively resizing the width of some of the columns. The immediate feedback through the animated transitions and the changed color coding help her to get a feeling of how sensitive the ranking is to changes in the attribute weights. She then refines the weights according to her preferences. The slope graph between the original ranking (stored in the snapshot) and the ranking based on the copied attribute combination clearly indicates that there are significant differences between the rankings. Jane realizes that she actually wants to find a university that not only matches her criteria but also has a high rank in the QS World University Ranking. To do this, she nests the original combined QS World University Ranking score, stored in the snapshot, within the customized combination. The result of this nested combination is shown in Figure 4.8. As a final step, she wants to make sure to only apply to universities that do not show a downward trend over the last 3 years. By following the slope graphs over time, she then picks the five universities that best fit her preferences and to which she will apply.

4.6. Evaluation

LineUp is tailored to allow novice users to solve tasks related to the creation, analysis, and comparison of multiple rankings. As discussed in the Related Work section, there is no single technique or software that fulfills all requirements described in Section 4.2. However, it is indeed possible to successfully solve most, if not all, ranking-related tasks with off-the-shelf tools such as Microsoft Excel or Tableau Desktop. To confirm this, we ran a pre-study where we asked an expert Excel user and an expert Tableau user to complete the ranking tasks described in Section 4.6.2. This informal test confirmed that solving these tasks with generic analysis and visualization tools is possible but tedious and time-consuming.

Also, tools such as Tableau and Excel require considerable scripting skills and experience to solve complex ranking tasks. In contrast, our technique aims to empower novice users to achieve the same results with very little training. A formal comparative study with a between-subject design where experts use Excel or Tableau and novices use LineUp would be unable to confirm this, as it would be impossible to tell whether the observed effects were caused by the difference in the subjects' backgrounds or between the tools. Also, a within-subject design that uses either experts or novices would be highly problematic. The first option would be to use Excel and Tableau experts and compare their performance using each tool. However, the experts would not only be biased because of their previous training in the respective tool, but, even more importantly, they are not the target audience of our technique. The second option, a within-subject design that tests how novices would complete the tasks using the different tools is not possible either because of the level of experience and knowledge necessary to perform the tasks in the other two tools. Consequently, we believe that it is more meaningful to show the effectiveness of the LineUp technique in a qualitative study.

4.6.1. Study Design

For the qualitative study we recruited eight participants (6 male, 2 female) between 26 and 34 years old. They are all researchers or students with a background in computer science, bioinformatics, or public health. Half of them indicated that they have some experience with visualization, one of them had considerable experience. In a pilot study with an additional participant, we ensured that the overall process runs smoothly and that the tasks are easy to understand. Prior to the actual study, we checked the participants for color blindness. We then introduced the LineUp technique and the study datasets. After the introduction, the participants had the opportunity to familiarize themselves with the software (using a different dataset than those used during the actual study) and to ask questions concerning the concept and interactions. Overall, the introduction and warm-up phase took about 25 minutes per subject. We advised the participants to “think aloud” during the study. In addition to measuring task completion time for the answers, we took notes on the participants' approaches to the tasks and what problems they encountered. After each task, participants had to fill out the standardized NASA-TLX questionnaire for workload assessment [Har06]. After they had finished all tasks, we gave the subjects a questionnaire with 23 questions, which evaluated the tool on a 7-point Likert scale. It included task-specific and general questions about the LineUp technique and questions making comparisons with Excel and Tableau (which they were asked to answer only if they were sufficiently experienced in using one of the tools). Additionally, we concluded every session by asking open questions to collect detailed feedback and suggestions for improvements.

4.6.2. Results and Discussion

We designed 12 tasks that the participants had to perform using LineUp. The tasks covered all important aspects of the technique concerning the creation, analysis, and comparison of rankings. Detailed task descriptions, study results, and questionnaires can be found in the supplementary material of [GLG⁺13].

Although we aimed to formulate the tasks as atomically as possible, they intentionally had different levels of complexity. This is also apparent in the results of the NASA-TLX workload assessment questionnaires. We measured task completion times to approximate the potential for improvements in the interface and to identify disruptions in the workflow. In general, participants were able to fulfill the tasks successfully and in a short period of time. Two outliers, where users needed more time, were noticeable: Task 7, in which participants were asked to filter data and evaluate the change, and Task 12, a comprehensive example with subtasks. As Task 12 comprised multiple aspects, we expected it to take longer. The long task completion times for Task 7, on the other hand, were unexpected. While most users solved the task in reasonable time, some needed longer because they tried to evaluate the changes in-place in the table, while we had assumed that they would use the snapshot feature, which would make the task significantly easier.

In the questionnaire, which is based on a 7-point Likert scale ranging from strongly agree (1) to strongly disagree (7), the majority of participants stated that the technique is visually pleasing (mean 1.6), potentially helpful for many different application scenarios (1.3), and generally easy to understand (2.4).

In the questionnaire, participants were asked about their experience level in Excel and Tableau. Although the participants used only LineUp in the study, we wanted to know (if they had any experience in Excel or Tableau) whether (a) the task could be solved in one of the other tools and (b) whether this could have yielded more insight. The average level of experience in Excel was 3.8 on a 7-point scale ranging from novice (1) to expert (7). None of the participants were familiar with Tableau. Participants rated the expected difficulty for doing the same tasks with Excel 4.4 on average. Most of them were convinced that LineUp would save time (1.6) and allow them to gather more insights (1.6).

In addition to evaluating the general effectiveness of our solution, we also wanted to find out if users are able to understand the mapping editor and which layout they prefer. The study showed that the participants found the parallel editor easy to understand (1.8) but that they were skeptical about the orthogonal layout (4.4).

In the open-ended feedback session, participants particularly valued the interactive approach combined with immediate feedback. Some of them stated that using drag-and-drop

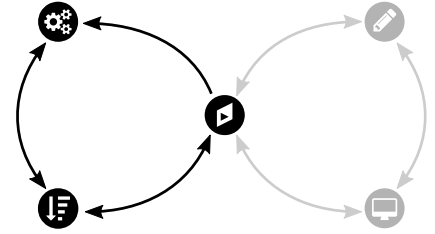
to create complex rankings is much more intuitive than typing formulas in Excel. Also, snapshots for comparing rankings were positively mentioned several times. In addition to the positive aspects, participants provided suggestions for improvements and reported minor complications in their workflow. They mentioned, for instance, that the button for creating a combined score is hard to find and suggested introducing a mode in which users can hold a modifier key and then select the attributes they want to combine. Also, some participants said that the rank change encoding disappears too fast to keep track of the changes. However, after reviewing the notes taken during the study, it was obvious that the participants who mentioned this did not use the snapshot feature, which provides better support for tracking rank changes than the transient change indicators.

The Tableau and Excel experts from the pre-study were asked to complete the same tasks as the regular participants. As previously mentioned, both were able to perform most of the tasks correctly in the pre-study. However, their completion times suggest that novice users are considerably faster in solving the tasks with LineUp than the experts using Tableau or Excel. We did not formally measure the task completion time in the pre-study, as the goal of the pre-study was not to collect performance data that can be used for comparison, but to get an impression if the tasks are possible in general and how difficult it is to solve them. Simple tasks that required users to filter, search, or sort to a certain attribute had about the same performance in all tools. However, the pre-study revealed that the experts in both tools had problems to solve tasks that involved “What if?” questions. For instance, it was difficult for them to change weights or mappings and evaluate changes, as these tasks benefit significantly from interactive refinement with immediate visual feedback. The Tableau expert even mentioned that “there is only trial-and-error or manual calculation. I do not know an elegant way of doing that”.

4.7. Conclusion

In this chapter we introduced LineUp, a technique for creating, analyzing, and comparing multi-attribute rankings. Initially, our goal for the technique was to enable domain experts to formulate complex biological questions in an intuitive and visual way. When we realized that the technique can be applied in many scenarios, we chose to generalize it to a domain-independent universal solution for rankings of items.

Our evaluation shows that major strengths of LineUp are the interactive refinement of weights and the ability to easily track changes. While this is valuable in many cases, it still requires users to actually perform the changes in order to see their result. We validated the *algorithm design* and *encoding/interaction design* aspects of LineUp according to Munzner’s model [Mun09] in an evaluation. What remains to be done is to observe actual users applying our tool in real-world analyses and to observe adoption rates.



5 | Guided StratomeX

Guided Visual Exploration of Genomic Stratifications in Cancer

Contents

5.1. Introduction	56
5.2. Application Areas	57
5.3. Method	59
5.4. Case Study	64
5.5. Conclusion	66

This chapter introduces a guidance process for selecting data subsets from a dataset collection, which covers the left part of the SPARE model described in Section 1.2. The process is illustrated and motivated by a problem from the biomedical domain, namely the visual exploration of genomic stratifications in cancer. We propose the *Guided StratomeX* tool, which is an extended version of the existing visualization technique *StratomeX* combined with our guidance process in order to support users in selecting the next column, i.e., data subset. An introduction to the *StratomeX* technique is given in Section 3.1. A central element of the guidance process is a wizard interface that acts as a placeholder for a column, which helps the user to formulate queries for a data subset. The query is used as input to an algorithm that scores available data subsets in the dataset collection. The resulting scores are visualized with the *LineUp* visualization technique introduced in the previous chapter. Using *LineUp*, users can rank data subsets based on various attributes, and select individual ones that match the initial query, thus closing the guidance process loop. A detailed use case study demonstrates the usage of the proposed method and application. More information about Guided StratomeX can be found at <http://stratomex.caleydo.org>.

5.1. Introduction

Cancer is a heterogeneous disease, and molecular profiling of tumors from large cohorts has enabled characterization of new tumor subtypes. This is a prerequisite for improving personalized treatment and ultimately better patient outcomes. Potential tumor subtypes can be identified with methods such as unsupervised clustering [Vea10] or network-based stratification [HSC⁺13], which assign patients to sets based on high-dimensional molecular profiles. Detailed characterization of identified sets and their interpretation, however, remain a time-consuming exploratory process.

To address these challenges, we propose *Guided StratomeX*, an interactive visualization tool, to efficiently compare multiple patient stratifications, to correlate patient sets with clinical information or genomic alterations, and to view the differences between molecular profiles across patient sets. Although we focus on cancer genomics here, Guided StratomeX can also be applied in other disease cohorts. Guided StratomeX is an extension to and superset of the previously published StratomeX [LSS⁺12] visualization technique that is introduced in Section 3.1.

Thousands of patient stratifications can be derived from large cancer genomics datasets. This space of patient stratifications—which we call the ‘stratome’—contains stratifications based on, for example, clustering of mRNA, microRNA, or protein expression matrices; the mutation or copy number status of genes; or on clinical variables. Due to the size of the stratome and the heterogeneity of the underlying datasets, integration of computational and visual approaches is indispensable to the analyst in identifying biologically or clinically meaningful stratifications, as well as clinical parameters and pathways that together provide a comprehensive view of each patient set.

Guided StratomeX also integrates a computational framework for query-based guided exploration of the stratome directly into the visualization (Figure 5.1), enabling discovery of novel relationships between patient sets and efficient generation and refinement of hypotheses about tumor subtypes. A ‘query wizard’ provides step-by-step instructions for defining queries, and a range of computational methods are used to generate rankings. Queries score stratifications, for example, based on their overlap with a particular patient set, or based on their overall similarity to a selected stratification. Furthermore, the analyst can query the collection for stratifications that contain patient sets that exhibit differences in survival or differential regulation of pathways. We use *LineUp*, a multi-attribute ranking technique, to visualize the results of these queries and to show which stratifications or pathways score high. The tight integration between the StratomeX and LineUp views, as well as the dynamic computation of scores, is essential for rapid identification of meaningful relationships between stratifications, clinical parameters, and pathways.

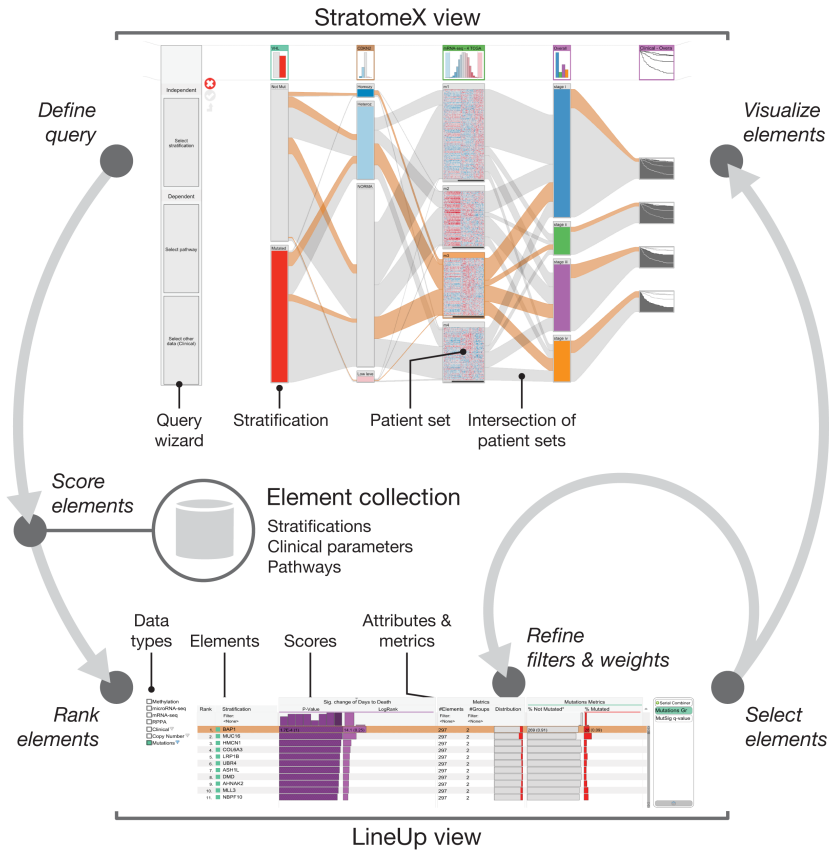


Figure 5.1.: Seamless integration of visual and computational components in the Guided StratomeX tool. In the StratomeX view (top) columns represent different stratifications, and the bands between the columns show the intersection between the subsets. The results of queries are lists of elements ranked by a score, which are shown in the LineUp view (bottom). Elements selected in the LineUp view are immediately visualized in the StratomeX view, enabling analysts to rapidly explore the results of queries.

5.2. Application Areas

Data analysis and visualization methods for cancer subtype analysis have three distinct application areas: (1) data exploration, i.e., discovery of novel insights, (2) hypothesis confirmation, i.e., finding supporting evidence for or against a working theory, and (3) presentation, i.e., communicating findings to others. Unlike other approaches, our guided visual exploration approach aims to address all three areas, with a focus on data exploration and hypothesis confirmation.

Data Exploration Within the data exploration area, we identify three primary tasks: (a) the creation of novel and improved stratifications, (b) judging the quality of stratifications, and (c) reasoning about stratifications. Stratifications are created using, e.g., clustering algorithms based on mRNA patterns [Vea10] or network based stratification [HSC⁺13]. Our approach employs such methods, i.e., enables analysts to run various clustering algorithms, or to import the result of such algorithms. In addition, StratomeX enables analysts to manually refine stratifications, e.g., by splitting clusters based on a clinical variable.

The quality of stratifications can be judged based on algorithmically derived measures, such as Dunn’s index [CSC⁺12], or silhouette values [TMH⁺13], or visually, either by visualizing the content of clusters in, e.g., cluster heatmaps [ESBB98], or by visualizing differences between alternative clustering results [LSP⁺10]. Our approach is the first to integrate all of these methods: scores can be loaded as supplemental data for stratifications, which can then be used to judge and rank stratifications. More importantly, Guided StratomeX integrates both, the visualization of cluster content and the analysis of cluster differences in a single concise visualization. Finally, our approach enables analysts to reason about stratifications, e.g., to identify supporting evidence in clinical or other data, by dynamically exploring the whole space of the stratome using targeted queries. This makes it easy for analysts to quickly check large quantities of candidate stratifications for mutual support.

The deep integration of analytical methods and visual exploration distinguishes the method described here from our previously published visualization-only approach [LSS⁺12]. The original method enables only a *knowledge-driven approach*, i.e., the confirmation and communication of existing hypothesis based on the analyst’s knowledge of the dataset. By integrating methods to identify and rank stratifications, clinical variables, and pathways, we enable a *data-driven approach* that does not rely on an analyst’s prior knowledge of the dataset to cancer subtype analysis. Such a data driven approach is necessary for data exploration in large datasets to discover novel insights.

Confirmatory Analysis The data-driven approach, however, also plays a major role in confirmatory analysis. Guided StratomeX makes it possible to efficiently put candidate stratifications in context of other data types, such as clinical outcomes, to judge effects of different stratifications, or pathways, to speculate about causes and effects of a particular cancer subtype. While we employ algorithms such as gene set enrichment analysis [STM⁺05] to identify pathways, and logrank tests to identify interesting stratifications based on clinical variables, it is the deep integration of these analytic processes with the interactive visualization that accelerates the analytical workflow. This enables analysts to explore a larger number of hypotheses in less time and allows them to perform a deeper analysis of the data than possible with other approaches in the same amount of time.

Presentation Finally, Guided StratomeX is also well suited for the presentation of results. While it is not the goal of Guided StratomeX to produce publication-ready figures, our visual representation is suitable to efficiently convey important characteristics of candidate subtypes. The visual encoding used by StratomeX is easy to understand and visually appealing. Also, Guided StratomeX can be used to communicate among distributed teams, either by exporting figures from Guided StratomeX, or by passing along project files that contain all the data as well as the analysis setup. In Chapter 7, we present an advanced model for presenting visual analysis results based on record provenance data.

5.3. Method

Figure 5.2 illustrates the proposed guidance process for iteratively selecting data subsets from a dataset collection. Users perform the main visual analysis within the **Exploration** stage. Employing integrated tools such as wizards, users can formulate a query to the guidance system. The available query methods are specific to the visualization technique and the data domain. Within the **Scoring** stage, the defined query is transformed into a scoring algorithm that is used to compute a score for each subset in the dataset collection. The data subsets along with their resulting scores are then presented in an interactive ranking interface in the **Ranking** stage. By combining scores and refining filters, users can adapt the ranking to their needs. The resulting ranked list then drives the selection of the data subset that best matches the user’s query, which is added to the main visualization technique. This closes the loop by returning to the Exploration stage, in which users can continue their exploration and formulate new queries.

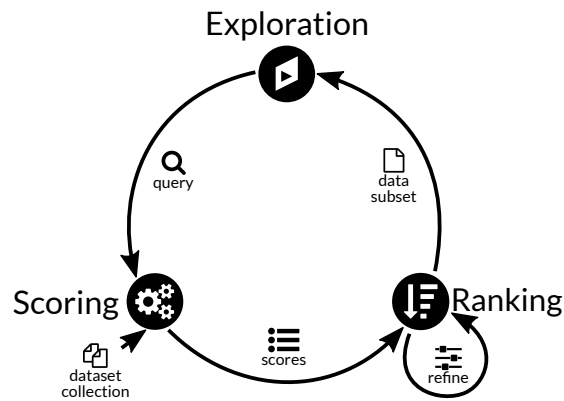


Figure 5.2.: Proposed guidance process, consisting of three stages: Exploration, Scoring, and Ranking.

The proposed guidance process is widely applicable in various problem domains. *Entourage* [LPK⁺13] use a similar combination of scoring a pathway collection and presenting the result as a ranking for guiding users to potentially interesting related pathways. Behrisch et al. [BBH⁺17] propose with Magnostics a set of feature descriptors for networks represented as matrix views that can be used to guide users to interesting subsets in large network collections in a similar fashion.

A variant of the proposed guidance process is that the *Exploration* and *Ranking* stages are combined in cases in which the exploration of the ranked items are of main interest. *Pathfinder* [PGS⁺16] is visualization technique for querying and exploring ranked paths in large networks. Tatu et al. [TMF⁺12] propose a workflow for guiding to interesting subspaces of large high-dimensional datasets. Both focus on the ranked and scored item collections with additional support views showing details about individual items.

Figure 5.1 summarizes the guidance process as applied in Guided StratomeX. It consists of three components: (1) A *main visualization exploration technique*, such as StratomeX [LSS⁺12], (2) a *query wizard* along with data-mining algorithms, and (3) a visual ranking interface—LineUp—for inspecting the query results. The process itself is independent of the visual exploration technique applied. Each component is described in more detail below.

5.3.1. StratomeX View

As described in the introduction of this chapter, we use an extended version of StratomeX as the main visual exploration technique in the Guided StratomeX tool. Figure 5.3 illustrates the basic user interface and components of StratomeX. A detailed introduction to the original StratomeX can be found in Section 3.1.

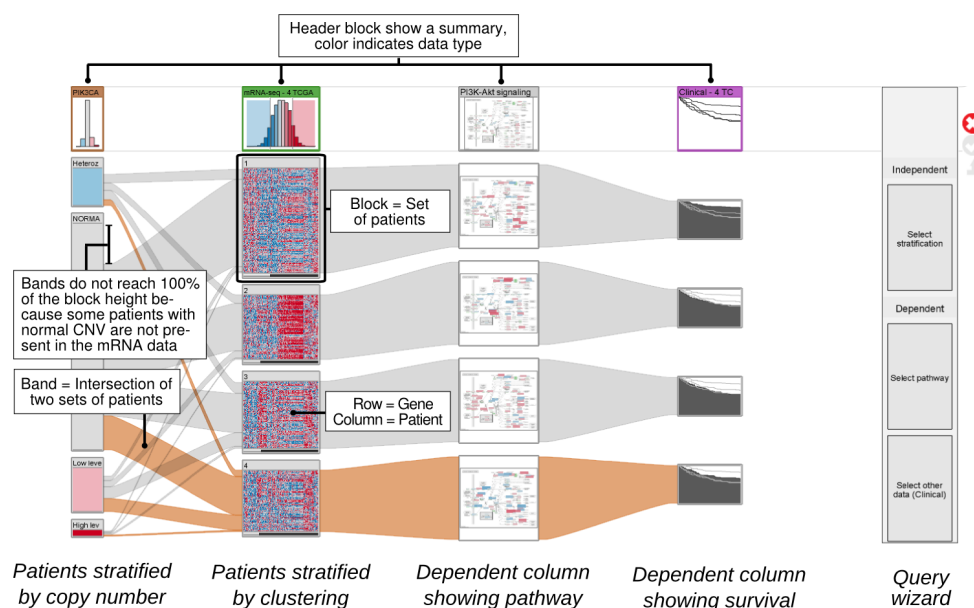


Figure 5.3.: Guided StratomeX user interface. Stratifications are represented as columns, showing additional data per block. Bands indicate set relationships between two neighboring columns. In addition, on the right side the query wizard is visible, acting as the starting point for the guidance process.

5.3.2. Queries Wizard and Scores

The column on the right of Figure 5.3 shows the ‘query wizard’ which is an assistive user interface that supports users in the process of adding new columns to Guided StratomeX and is an extension to the original StratomeX [LSS⁺12]. In addition to basic browsing, filtering, and ranking of stratifications, pathways, and clinical variables, Guided StratomeX supports a series of advanced query methods to find additional stratifications and pathways based on patterns identified in the StratomeX view.

Figure 5.4 summarizes the possible options along with their algorithmic counterpart that are available in a flow chart. Diamond nodes denote decision points within the wizard that require user input. Edges describe the options the user can choose from. Gray boxes represent actions taken by the user in the StratomeX view, white boxes with gray outlines mark actions executed by the system, and colored boxes indicate actions taken by the user in the LineUp view. Depending on the current step, the user either needs to select an option from a list, or is instructed to take actions in the user interface, for example, to select a stratification (column) or a set of patients (block) in StratomeX or to perform

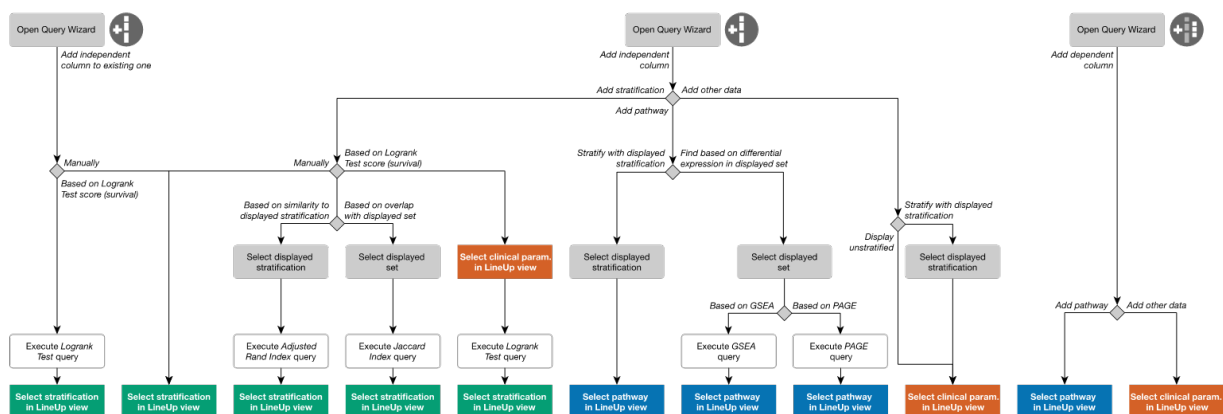


Figure 5.4.: Flow chart of query options and the query wizard menu items used to trigger them

actions in the LineUp view. Once this workflow is successfully completed, a new column will be added to StratomeX.

Some of the queries implemented in Guided StratomeX are based on hypothesis tests for which p-values are provided along with the test scores. The results of these queries, however, must not be interpreted as statistically reliable results, since correction for multiple hypothesis testing is not provided in the current implementation although some queries involve thousands of tests. Generally, the scores are provided to guide the user to stratifications or pathways that provide additional insight into patterns observed in the StratomeX view and to generate new hypotheses. Following scores are implemented in the proposed Guided StratomeX tool:

Scoring stratifications based on similarity to a selected stratification This query is useful for finding stratifications that are similar to a currently displayed stratification. The Adjusted Rand Index [HA85] is used to compare each stratification in the collection against the query stratification selected by the user.

Scoring stratifications based on overlap with a selected patient set In contrast to the Adjusted Rand Index, which quantifies similarities between stratifications, this type of query is designed to identify stratifications that contain sets similar to a query set in a displayed stratification. The score for a set is the Jaccard Index describing its similarity to the query set and computed for all sets in every stratification in the collection of stratifications, but only the best score for each stratification will be reported. In addition,

if the query is triggered from a binary stratification, such as mutations, a mutual exclusivity score is computed per set, which can be used to identify genes that are mutated in non-overlapping sets of patients.

Scoring stratifications based on logrank test for patient survival This query identifies sets of patients that exhibit altered survival times compared to the rest of the patients in the same stratification. It uses the logrank test (Mantel-Haenszel test) to score the stratifications and assigns larger scores to more extreme differences in survival. Similar to the previous method, the score is computed for each considered set of stratifications and the best result per stratification is presented to the user. A p-value for the best result is provided as guidance.

Scoring pathways based on gene set enrichment for a selected patient set This type of query is designed to identify pathways that are over- or underexpressed in a patient set relative to the rest of the cohort. It takes a set of patients as input and computes differential gene expression levels for patients in the query set against the rest of the patients in the same stratification. The differential expression levels are used to score pathways using either Gene Set Enrichment Analysis (GSEA) [STM⁺05] or Parametric Assignment of Gene Set Enrichment (PAGE) [KV05]. Additional meta-information, such as the number and percentage of mapped genes, are shown to allow filtering operations, such as exclusion of pathways with too many or few genes with expression levels.

Importing externally computed scores Any external score associated with stratifications or pathways can be imported using the data import wizard, and used for exploration of the data.

5.3.3. Ranking Interface

The computed scores for each item in the dataset collection are visualized using LineUp. Figure 5.5 explains the user interface of LineUp as used in Guided StratomeX in more details. In this example, each row corresponds to a gene for which a series of attributes are available. Attributes can be (1) simple general metrics such as the number of groups or patients in a stratification, (2) data specific metrics, such as the mutation rate, (3) computed scores based on queries triggered by the user, such as the similarity between two patient subsets, or (4) imported scores or groupings that have been computed using an external tool, such as the Mutation Significance (MutSig) q-value of the genes. By



Figure 5.5.: LineUp view with multi-attribute ranking and filtering. The interactive ranking visualization allows users to filter and order stratifications according to a single attribute or a combination of multiple attributes, such as the sum or the maximum of attributes, associated with the stratifications.

selecting a row, the stratification will be added to the StratomeX view as a new column. Additionally, analysts can search for stratifications of interest by typing their name. The list of available datasets on the left allows analysts to select the subset of stratifications that will be scored by the query and incorporated into the ranking. In this example, only gene mutation calls are included. The ‘memo pad’ on the right allows analysts to store attributes that are not of immediate interest. The general visualization technique itself is presented in Chapter 4 of this thesis.

5.4. Case Study

To demonstrate that StratomeX supports the generation and refinement of novel hypotheses using a combination of data-driven and knowledge-driven queries, we explored the clear cell renal carcinoma (ccRCC) data set published by The Cancer Genome Atlas (TCGA) consortium [The13].

We started by confirming the hypothesis from the study that patients with mutations in *BAP1* have much worse outcomes compared to patients without such mutations. A query was triggered by using the StratomeX query wizard, looking for high log-rank scores in the ‘Days to Death’ variable in all mutation data subsets. The result of the query, shown in the LineUp view at the bottom, is a ranking of 19 significantly mutated genes based on the log-rank score (purple bars) and corresponding p-values (dark purple bars,

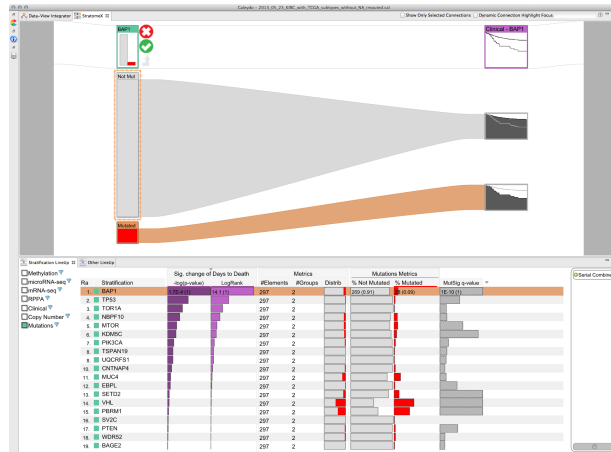


Figure 5.6.: Illustration of Guided StratomeX. The clear difference in the Kaplan-Maier plots shown on the right (one for patients with and one for patients without mutation) confirms the hypotheses from the study that patients with mutations in *BAP1* have much worse outcomes than patients without such mutations.

diagonal hatch pattern indicates ‘inverted’ mapping with $-\log(p)$), obtained by applying the corresponding stratifications to the ‘Days to Death’ variable. The minimum patient set size for which scores were computed was 10. Additional columns show common metrics for mutated genes, and the MutSig q-values ranging between 0 and 0.1. Figure 5.6 shows the final outcome and confirmation. The top hit *BAP1* was selected in the LineUp view and is shown in the StratomeX view (left column) next to the Kaplan-Meier plots for ‘Days to Death’ stratified by *BAP1* mutation status (right column). The plots indicate that patients with a mutation in *BAP1* have worse outcomes than those without a *BAP1* mutation.

While this observation was made in sequence-level data, we were also interested in whether we could find such patient sets based on patterns in functional data, such as the expression levels of microRNAs, mRNAs, or proteins. We queried a total of 51 clustering results - 16 for mRNA-seq data [Bro13e, Bro13d], 14 for microRNA-seq data [Bro13c, Bro13b], 14 for protein expression data (reverse-phase protein array, RPPA) [Bro13g, Bro13f] and 7 for DNA methylation data [Bro13a] - obtained from the 23 May 2013 Firehose analysis run against the ‘Days to Death’ survival variable with the ‘logrank query’ of the query wizard.

The top result (logrank test score = 68.6, p-value = 1.1×10^{-16} ,) is a clustering of RPPA data into 8 clusters found using a consensus non-negative matrix factorization clustering approach [Bro13g]. The Kaplan-Meier curves indicate that the 57 patients in cluster 8

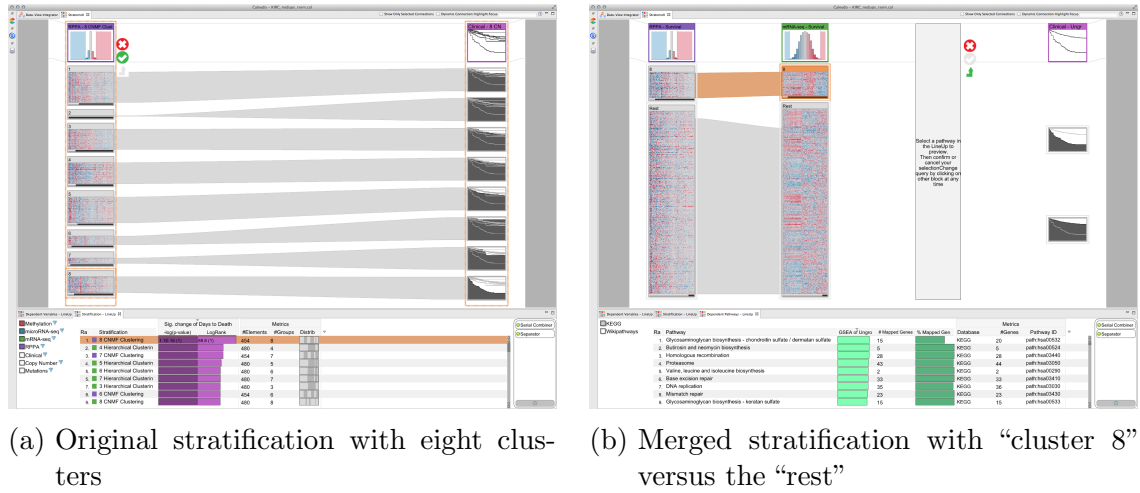


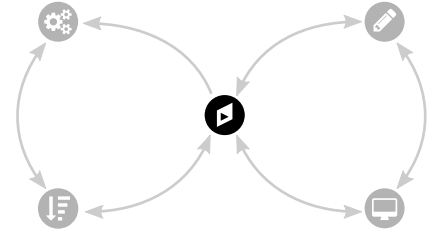
Figure 5.7.: Screenshots of Guided StratomeX, illustrating that patients in “cluster 8” (bottom right in 5.7a and top right in 5.7b) have fewer days to death than other patients.

have notably poorer outcomes (see Figure 5.7a). Since the outcomes of the patients in the other groups are all fairly similar, we decided to study the patients in cluster 8 ($n = 57$) relative to the remaining patients ($n = 397$) and created a new stratification of the RPPA data with only two groups: “cluster 8” and “rest” (see Figure 5.7b).

In summary, we were not only able to quickly confirm a published finding of the study, we also highlighted a potential additional finding. A more detailed case study of how Guided StratomeX is applied to this cancer study was published by Streit et al. [SLG⁺14].

5.5. Conclusion

In this chapter we presented a guidance process for supporting users in characterizing cancer subtypes. We used and extended StratomeX and integrated a guidance process. Users can flexibly define queries that are converted into scores and yield a ranked list of matching data subsets. In addition, we showed in our case study that we can not only quickly reproduce published findings, but also make new ones. The proposed guidance process is not restricted to the presented domain problem, but can be applied to any domain problem by adapting or replacing the main visualization technique and defining the queries and scores that are useful for the given domain problem.



6 | Domino

Extracting, Comparing, and Manipulating Subsets across Multiple Tabular Datasets

Contents

6.1. Introduction	68
6.2. Domino Visualization Technique	69
6.3. Spectrum of Supported Visualizations	78
6.4. Interface and Interaction Design	78
6.5. Use Cases	83
6.6. Discussion	87
6.7. Conclusion	89

This chapter describes *Domino* a novel multiform visualization technique for effectively representing subsets and the relationships between them. By providing comprehensive tools to arrange, combine, and extract subsets, *Domino* allows users to create both common visualization techniques and advanced visualizations tailored to specific use cases. It provides powerful toolset for guidance in exploration based user tasks, including innovative interactive features such as placeholders and live previews that support rapid creation of complex analysis setups. Besides the concept itself and an example set of existing visualization techniques that are covered, the interface and interaction design of the prototype is discussed. The practical applicability is shown via two use cases. This chapter concludes with a discussion of the concept, its advantages, and limitations. A prototype is available at <http://domino.caleydo.org>.

6.1. Introduction

Common heterogeneous data visualization methods show data associated with a single shared item type. The popular cars dataset [HV81], for example, is defined over the item type *car* and contains multiple observations across several data types for each car. A parallel coordinates plot could be used to visualize such data. Multi-dataset visualizations typically follow the same pattern: they refer to a single, shared item type. A medical dataset may contain data about patients, such as gender, birth date, or height, while a blood test dataset for the same patients will contain measurements such as blood type or white blood cell count. These datasets follow a relational data model where each dataset contains one or more observations for a given item type.

In practice, however, an increasing number of domains contain rich data for multiple item types. Figure 6.1 shows a simple illustrative example from the music industry¹ that includes multiple datasets defined in terms of the item types *artist* (e.g., count of number-one hits, gender, origin) and *country* (e.g., number-one hits per artists, total number of albums sold, sales normalized to population).

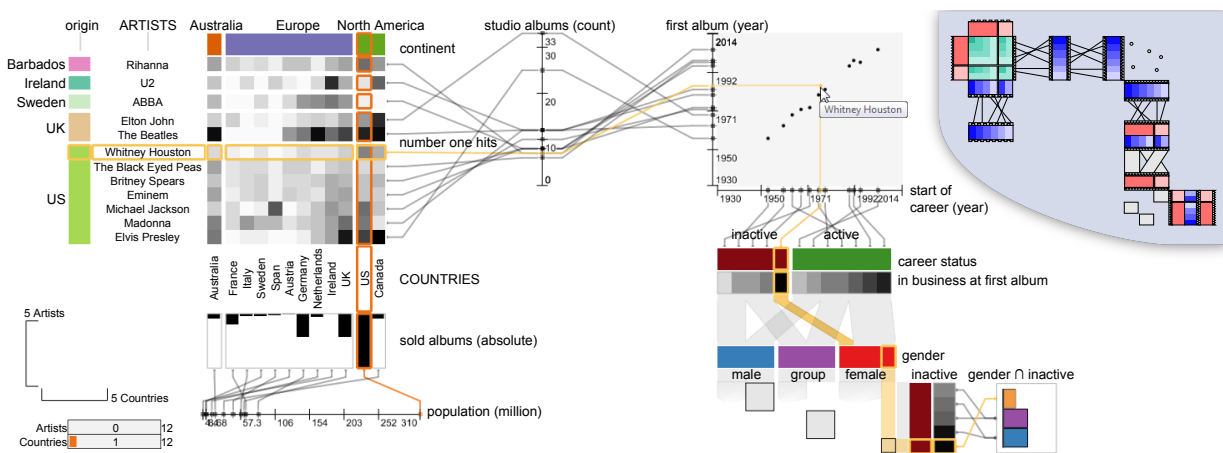


Figure 6.1.: Domino showing relationships between subsets of a music charts dataset. The visualization illustrates that *Whitney Houston* is a female, inactive artist who has had many number-one hits in English speaking countries but produced fewer than 10 studio albums. The schematic illustration in the top-right corner shows the setup using a graphical notation.

¹The dataset was collected from multiple web sources. References and the dataset are available at <http://domino.caleydo.org>.

This chapter introduces *Domino*, a novel visualization technique that enables analysts to explore a mix of numerical and categorical data connected via various item types. This is achieved by allowing users to freely arrange, combine, and manipulate subsets (blocks), visualize associated data, and explicitly represent the relationships between these subsets. Domino enables analysts to rapidly assemble both established visualization techniques and novel combinations specifically tailored to the task at hand. In addition, we present a prototype implementation that enables users to exploit the wide spectrum of possibilities that the technique offers. To assist users in the process of adding new subsets to the setup and combining previously defined ones, we introduce *placeholders* that indicate possible options for placing a subset, and *live previews* that show possible visual encodings of the associated data.

We demonstrate the utility and flexibility of the system by means of two use cases: the first discusses a small music charts dataset, while the second uses a collection of datasets from cancer genomics.

6.2. Domino Visualization Technique

Domino is a meta-visualization technique that enables analysts to arrange, extract, and manipulate subsets of interest and show the relationships between them at multiple levels of detail. The approach is designed for scenarios that include multiple heterogeneous datasets which are connected by shared identifiers (artist, country, etc.). The goal of Domino is to effectively represent **subsets**, the **data associated with the subsets**, and the **relationships between the subsets**.

The basic visual elements of the Domino technique are **blocks**, similar to tiles in a dominoes game, and **block relationships**, which connect the individual blocks. Figure 6.1 shows a snapshot of an analysis performed with the previously introduced music charts dataset. Blocks represent one or multiple combined subsets, where subsets can be numerical (e.g., *population* of countries), partitioned (artists' *gender*), or matrix data (*# of number-one hits* by artists in various countries). Depending on the data and the analysis task, various visualization techniques can be used to represent the subsets. While the matrix, for instance, is represented as a heatmap, the population of countries is visualized as a 1D scatterplot. Blocks can be arbitrarily positioned and rotated. The visual representations of the relationships between blocks are automatically displayed if the item types of two neighboring blocks match, analogously to matching dominoes. We consider two blocks as neighbors if they can be connected by a line without cutting across another block. The rows in the matrix that correspond to the *artists*, for example, are visually linked to the 1D scatterplot showing the *# of studio albums*.

In Domino, users are able to switch between three levels of granularity at which the relationships can be represented—focusing on the overlap between whole blocks, groups within blocks, or down to the lowest level of individual items. Depending on the arrangement of the blocks and the chosen granularity level, relationships are represented using lines, bands (bundles of lines), points, or rectangular regions (see Figure 6.1). The flexibility of freely arranging and associating blocks in Domino allows analysts to both quickly produce established visualization techniques, such as parallel sets, parallel coordinates, 2D scatter-plots, and to create tailored, complex visualizations to address the needs of a particular use case.

6.2.1. Blocks

The basic visual unit of the Domino technique are **blocks**, rectangular regions that represent subsets. We define i as an item that is part of the set I . I contains only items of the same type (e.g., *country*). A subset S is defined as a mathematical subset of $S \subseteq I$ (*countries in Europe*). We distinguish between three functions that determine the possible types of blocks in Domino (see Figure 6.2):

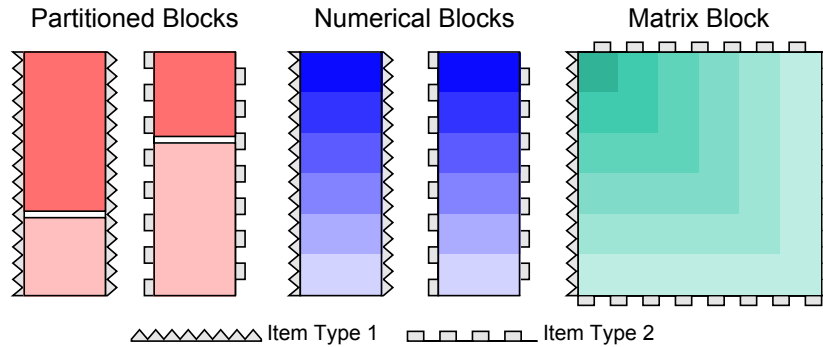


Figure 6.2.: The three block types in Domino: partitioned (red), numerical (blue), and matrix blocks (green). The border style indicates the type of the items contained in the subset. While partitioned and numerical blocks are defined by one item type, matrix blocks have two, one in each direction. Only blocks with matching item types can be combined.

A **partitioned block** is defined by the function f_p that maps items to groups that are associated with unique natural numbers (e.g., $\{Africa(0), Europe(1), \dots\}$, $\{male(0), female(1)\}$):

$$f_p(i \in I) \mapsto g \in P,$$

where P is a partition of items I . The subset S_g of one specific group $g \in P$ is defined as $S_g = \{i | f_p(i) = g\}$. A partitioned block can be viewed as a grouping of items without any associated data. The grouping can be the output of a clustering algorithm, derived from a categorical data attribute, or a quantization of a numerical block.

A **numerical block** is defined by the function f_n that maps items to numerical values (*population*, *# of studio albums*, or *retail value*):

$$f_n(i \in I) \mapsto v \in \mathbb{R}$$

A **matrix block** is defined by the function f_m that maps the product set of two subsets of items to numerical values:

$$f_m((i, j) \in I_1 \times I_2) \mapsto v \in \mathbb{R}$$

In contrast to partitioned and numerical blocks, in which all items are part of the same item set, a matrix block consists of two item sets (I_1 and I_2). The item sets can be identical (e.g., in a *city distance matrix*) or different (*number-one hits for countries* or *gene expression of cancer patients*). However, the item sets themselves need to be homogeneous with respect to the item type. According to our definition of blocks, the prominent cars dataset [HV81], for example, is not a single matrix block, but a collection of individual numerical and partitioned blocks for each variable (*horse power*, *year of construction*, etc.)

Figure 6.2 illustrates the three block types. The border style of the blocks encodes the different item types (e.g., *country* and *artist*). The sorting of items within blocks is essential for users to see patterns, for instance, in a clustered tabular dataset, or to see the distribution of items. For our technique, the sorting is also relevant because it determines the relationship representations that connect the blocks, as described in Section 6.2.2. From a mathematical point of view, sets do not define an order on the items that are contained in them. In Domino, items in blocks are sorted according to the functions defined above. In numerical blocks, items are sorted according to the associated data value. In partitioned blocks, the groups themselves are sorted according to the assigned natural number, while the individual items inside the group have no order. In matrix blocks, the two dimensions (horizontal and vertical) are sorted independently according to the mean value of the items. Sorting within blocks is illustrated by varying levels of brightness, as shown in Figure 6.2.

Multiform Visualization

An essential characteristic of blocks in Domino is that they can be used to visualize subsets using multiform visualization, i.e., that users can switch between various visualization techniques.

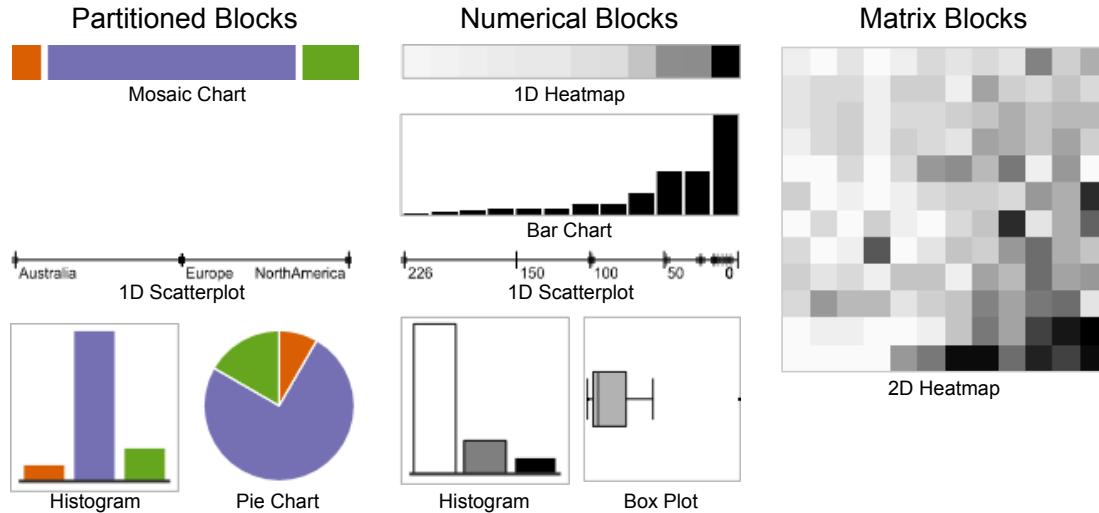


Figure 6.3.: Block visualization techniques categorized by block type (partitioned, numerical, and matrix) including 1D and 2D heatmaps, mosaic and bar charts, 1D scatterplots, histograms, pie charts, and boxplots.

Domino is not bound to a predefined set of visualization techniques. Throughout this chapter, however, we use a set of standard techniques, which can be seen in Figure 6.3, from which the analyst can freely choose. Depending on the type of the data represented by the block, only certain visualization types are suitable to encode data. While a heatmap representation can be applied to all three block types, 1D scatterplots and bar charts are available for partitioned and numerical blocks. Histograms and box plots are provided to represent numerical subsets, and pie charts can only be used to encode partitioned blocks. Although our current implementation is focused on these visualization types, the set of multiform techniques supported can be extended arbitrarily, depending on use case and application context. For the cancer genomics use case described in Section 6.5.2, for instance, we added Kaplan-Meier plots [RNP⁺10] to encode patient survival data.

Combined Blocks

Blocks that share the same item type, as indicated by the same border style in the illustrations, can be attached to each other, to form a **combined block**. The resulting combined block contains the union of the two item subsets. Figure 6.4 demonstrates by means of the music charts example how the stitching of blocks works. The analyst starts with the partitioned block *career status*, which separates active from inactive artists. In a second step, the analyst adds the numerical block *# of studio albums*, whose items are automatically partitioned according to the artists' *career status*. The two blocks are then combined with the matrix block that holds the absolute *# of number-one hits* for 12 countries. Again, as only one partitioning can drive the grouping of items in a combined block, the matrix block also inherits the *career status* partitioning. Finally, the analyst attaches the partitioned block *continent*, which splits the countries into three groups (Australia, Europe, and North America).

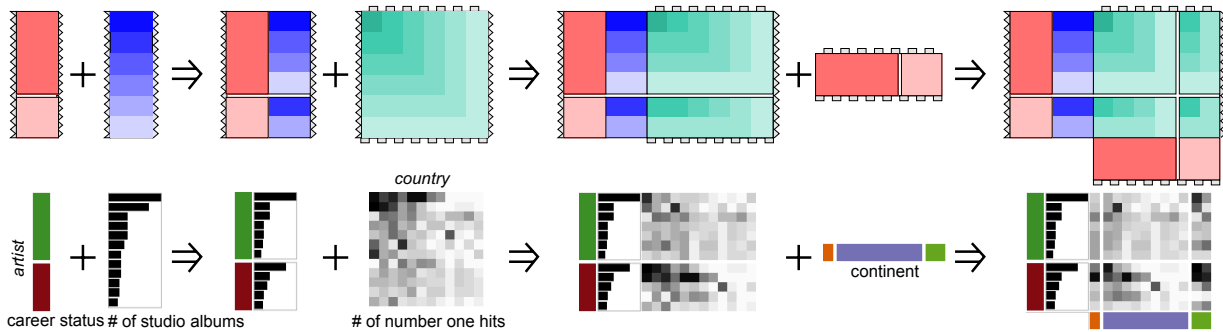


Figure 6.4.: Example sequence demonstrating the creation of a combined block. The user starts by attaching a partitioned block to a numerical block, which inherits the partitioning. Then the user adds a matrix block that is also partitioned accordingly. In the final step, the matrix column items are partitioned based on a partitioned block matching the column item type.

As described in Section 6.2.1, items within blocks are sorted. In the case of a combined block, the item order is determined by a hierarchical sorting strategy. For instance, if the user defines a primary and a secondary sorting criterion, the latter is only used if the order cannot be resolved using the former.

6.2.2. Block Relationships

In addition to blocks, the second class of visual elements in Domino are explicit representations of relationships between blocks. We consider blocks to be related to each other if their subsets contain items of the same type. In the previous section, we discussed that a partitioned block can only be subdivided into groups according to one primary partition, and a numerical block can only be sorted according to one primary sorting criterion. In the case of matrix blocks, these properties apply equally to each dimension. Consequently, the ability to create combined blocks by attaching multiple separate blocks requires re-sorting and/or re-distributing of the items in one or multiple blocks. Applying multiple sorting criteria hierarchically does not solve the issue, as only one can be the primary sorting criterion. However, in many analysis scenarios, users want to investigate the relationships between multiple blocks without giving up their designated sorting or partitioning. To address this issue, we introduce four degrees of relationships.

Relationship Degrees

A relationship degree defines the strength of the relationship between two blocks. We distinguish between four degrees: none, weak, medium, and strong, as shown in Table 6.1. The stronger the relationship, the more properties the blocks share. In Domino, the relationship degree is reflected by the proximity of two blocks in the layout. Users can freely switch between the four relationship degrees by moving blocks in the layout using drag-and-drop operations.


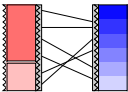
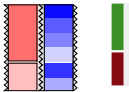
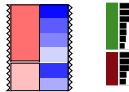



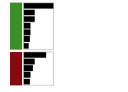
Shared ...	None	Weak	Medium	Strong
Item Type	—	•	•	•
Partitioning	—	—	•	•
Sorting	—	—	—	•
				
				

Table 6.1.: Properties of the possible block relationship degrees (none, weak, medium, and strong) along with both a schematic diagram and an application example of each relationship degree. The stronger the relationship, the more properties two related blocks share.

Two blocks have no relationship if they do not share the same item type (**none**). In the example used in Table 6.1, the artist’s *career status* cannot be matched with the *population* of countries. If the blocks contain items of the same type but retain their individual sorting or partitioning, their relationship is **weak**. For instance, the artists in the *career status* block are assigned to the *active/inactive* group and are then weakly connected to the corresponding bar in the *# of studio albums* bar chart, which is sorted by the height of the bars. Blocks that have a **medium** relationship degree have the same partitioning but do not require the same sorting or visualization technique (see Section 6.2.1), which means that items are aggregated on a per-group basis. In the example, the same subsets are used as in the weak case, but this time the *# of studio albums* block inherits the partitioning from the artists’ *career status* block. However, since both blocks can retain their individual sortings, different visualization techniques can be used. Finally, in the **strong** case both blocks also have to use the same sorting and must therefore apply the same visualization technique. In the example, the bars encoding the *# of studio albums* are sorted within the *career status* group.

As blocks with a strong relationship are directly attached to each other (see Section 6.2.1 on combined blocks), no explicit visual representation of the relationship is needed. Consequently, the following discussion on how to represent relationships only applies to medium and weak relationship degrees.

Relationship Representation

We characterize the visual representation of relationships by two aspects: the **direction** of blocks—either **parallel** or **orthogonal**—and the **granularity** of the relationship. Depending on the user’s task, we differentiate between three levels of granularity: **block**, **group**, and **item** relationships. Figure 6.5 illustrates the possible combinations using our graphical notation introduced in Figure 6.2.

At the finest granularity level for visualizing relationships between items, we connect the items by drawing lines between parallel blocks or by drawing points at the hypothetical intersection point of related items of orthogonal blocks. While the former results in parallel coordinate-like setups, the latter results in scatterplots. The examples in the bottom row of Figure 6.5 show the correlation between the numerical 1D scatterplot *# of studio albums* and *year of first album*. The positions of the points in the scatterplot as well as the starting and end points of lines in the parallel coordinates plot depend on the currently applied visualization technique in the two blocks involved. For histograms, for example, the bin position is used.

To represent relationships at the coarser granularity levels, between groups and whole blocks, we draw bands and rectangular regions, respectively, depending on the arrangement of the blocks. The width of a band or the size of a region is proportional to the overlap between the subsets. In earlier work [LSS⁺11, LSP⁺10, LSS⁺12], we already made use of bands to relate groups between partitioned datasets. This includes the visualization technique StratomeX [LSS⁺12] that is described in more detail in Section 3.1. In the parallel case, blocks connected by bands produce the parallel sets technique [KBH06]. The “parallel-group” example in Figure 6.5 shows the relationships between the groups of the partitioned blocks *gender* and *career status*, which are arranged in parallel. In the orthogonal case, the size of the regions is proportional to the number of shared items.

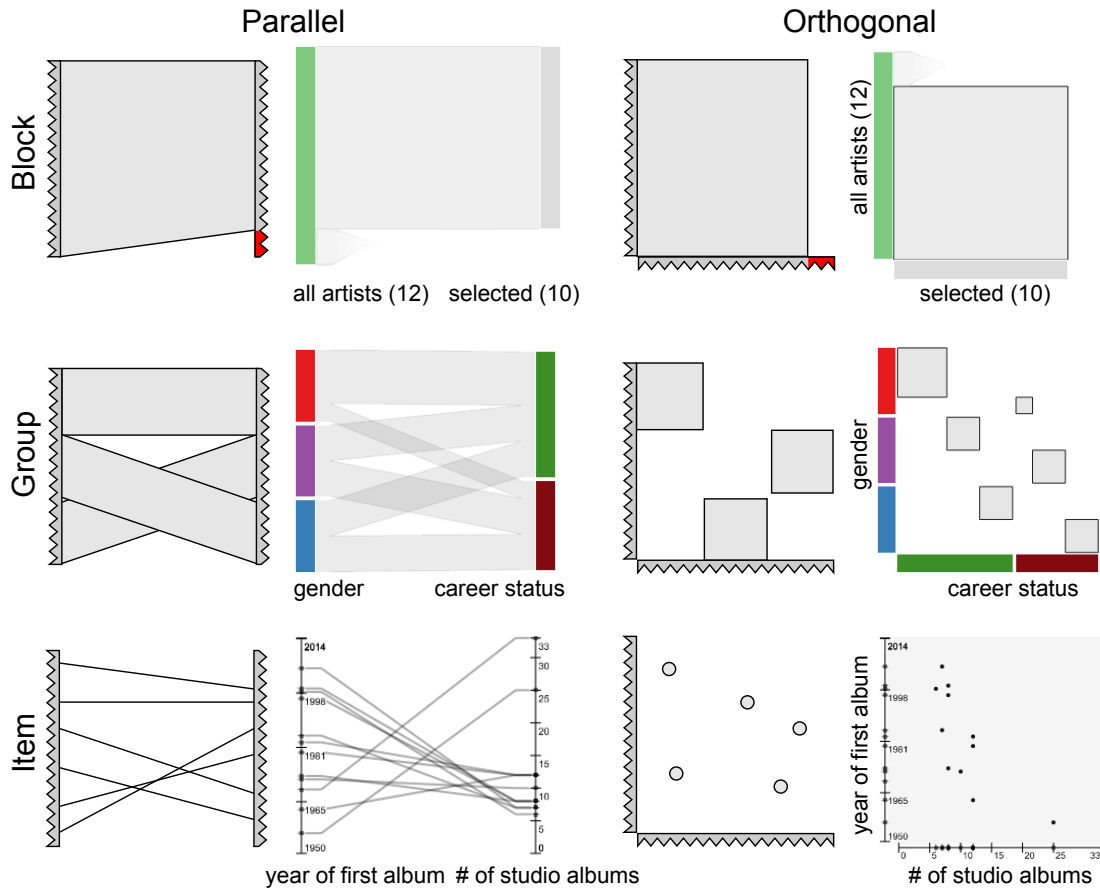


Figure 6.5.: Overview of possible relationship representations characterized by the granularity of the representation (block, group, and item) and the orientation of the block arrangement (parallel, orthogonal). All combinations are illustrated using the corresponding relationship representation and an example from the music charts dataset.

Subsets that do not fully overlap result in unshared items $S_1 \setminus S_2$ and $S_2 \setminus S_1$. In the “block” examples in Figure 6.5, for instance, the relationships of “all artists” (green) to a subset of ten selected artists (gray) are shown. In the illustrations, unshared items are indicated in red. In our implementation a fading band is used to represent them.

Note that since the positioning of blocks in Domino is not restricted to perfect vertical or horizontal alignments, all relationship representations can be sheared.

6.2.3. Subset Extraction and Manipulation

Up to this point, we have focused our discussion on the exploration of predefined subsets, relationships between these subsets, and the different ways of visualizing them. Beyond predefined subsets, Domino also supports users in the process of creating new subsets. Users can define new subsets either by combining existing ones using logical operations (union, intersection, set difference) or by extracting parts of subsets based on selected items or groups. Depending on the visual representation of a block or relationship, users can, for example, extract items above a certain threshold in a numerical 1D scatterplot, items aggregated in a single bin of a histogram, or a collection of points in a scatterplot that is formed by orthogonally arranged numerical blocks. In Figure 6.10, for instance, we selected the group *G-CIMP* in the matrix block and extracted all items to a separate block.

Enabling analysts to extract and manipulate subsets as needed during the analysis is a powerful approach in many scenarios. Ensuring that the user is always aware of the origin of the created subset is essential. In general, this provenance information can be provided either in a *temporal* or in a *spatial* manner. Domino offers both options.

In order to be able to present the provenance information, the user’s actions must be tracked. To understand what steps lead to the current subset configuration, users can then navigate through the history by using animation, or also replay the actions. Alternatively, provenance information can be encoded directly in the visualization, which we call *spatial* provenance representation. One example of such a spatial provenance presentation is the technique by van Elzen and van Wijk [vdEvW13], where they present previous analysis steps in a linear fashion, reminiscent of a filmstrip. Other examples are *VisTrails* [BCS⁺05], *ExPlates* [JE13], and *GraphTrail* [DHRL⁺12].

Instead of arranging the steps along a time line, the provenance information is contained implicitly in the design of the Domino technique. The relationship representation between subsets (blocks) and the colors of blocks allow analysts to track from which original subsets

new subsets have been derived. In Figure 6.10, for instance, it can be seen that we extracted all items in the *G-CIMP* group to a separate block. Further, if a newly extracted subset turns out to be irrelevant, users can go back to the original subset and follow a different path in the analysis. In the use cases presented in Section 6.5, we demonstrate how analysts make use of the temporal replay capabilities and the implicit encoding for tracking the evolution of the current subset configuration.

6.3. Spectrum of Supported Visualizations

The modular block concept of Domino enables analysts to create sophisticated visualization setups. In addition to using the inherent visualization techniques realized in multiform blocks (see Section 6.2.1), analysts can easily assemble standard techniques such as scatterplots, scatterplot matrices (SPLOMs) [CLN86], parallel coordinate plots [ID90], parallel sets [KBH06], mosaic plots [HK81], Sankey diagrams [RHF05], GPLOMs [IML13], Flexible Linked Axis [CvW11] (although our current implementation does not support free rotation), ConnectedCharts (with the exception of stacking and nesting) [VM12], Table Lens [RC94], and scattering points in parallel coordinates [YGX⁺09]—to name but a few. Our technique also conceptually subsumes our previously published Matchmaker [LSP⁺10], VisBricks [LSS⁺11], StratomeX [LSS⁺12], and Furby [SGG⁺14] techniques. Figure 6.6 shows a collection of techniques created with Domino. Each figure contains a schematic illustration together with an example generated with the music charts dataset.

However, in addition to the flexible composition of well-known techniques, another strength of Domino is that these techniques can be combined arbitrarily to form a wide spectrum of new hybrid visualizations which the analyst can tailor specifically to the task at hand, as demonstrated in the music charts visualization shown in Figure 6.1 and the example setup from our cancer genomics use case presented in Figure 6.10.

6.4. Interface and Interaction Design

The Domino technique opens up a wide range of possibilities for exploring, relating, and manipulating subsets. To ensure that users can apply this generic approach efficiently, a well-designed user interface combined with a specifically tailored interaction concept is essential. To assist users in the process of adding new blocks to the setup and combining existing blocks, we introduce *placeholders* that indicate the various positions for placing a subset and *live previews* for showing possible visualization outcomes (see Section 6.4.1).

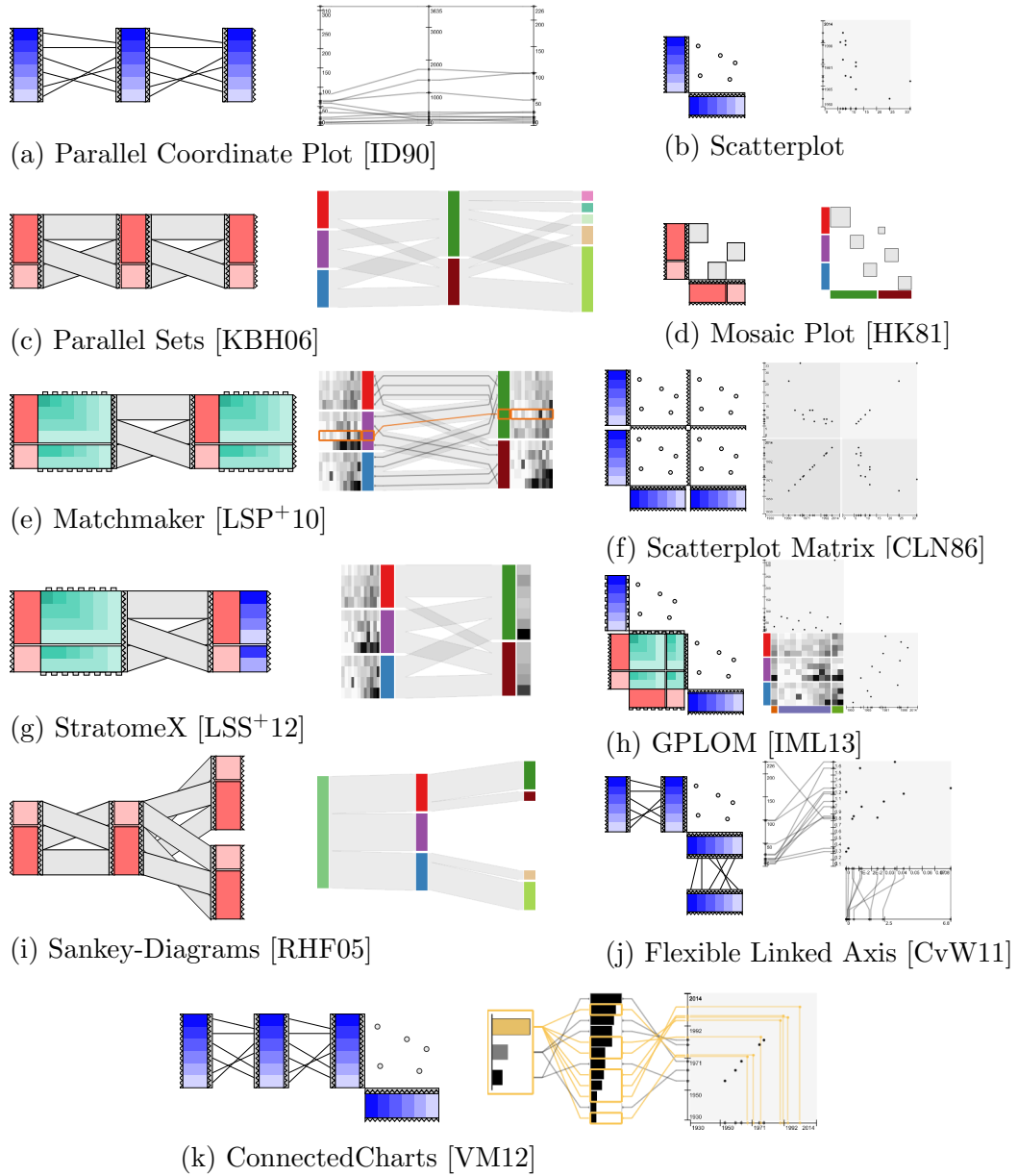


Figure 6.6.: Examples of supported visualization techniques demonstrating the flexibility of the Domino approach.

Various interaction modes let users select and relate blocks, groups, or individual items (see Section 6.4.2). Interactions for block scaling (see Section 6.4.3) ensure that users can effectively handle scenarios of different complexity and scale—from small datasets like the music charts dataset with only 12 items to large scenarios including a multitude of datasets such as that discussed in the cancer use case described in Section 6.5.2.

As shown in Figure 6.7, the main user interface consists of two parts: the *Domino Board* and the *Block Browser*. In addition to the two main interfaces, support views provide information about the data and current selections.

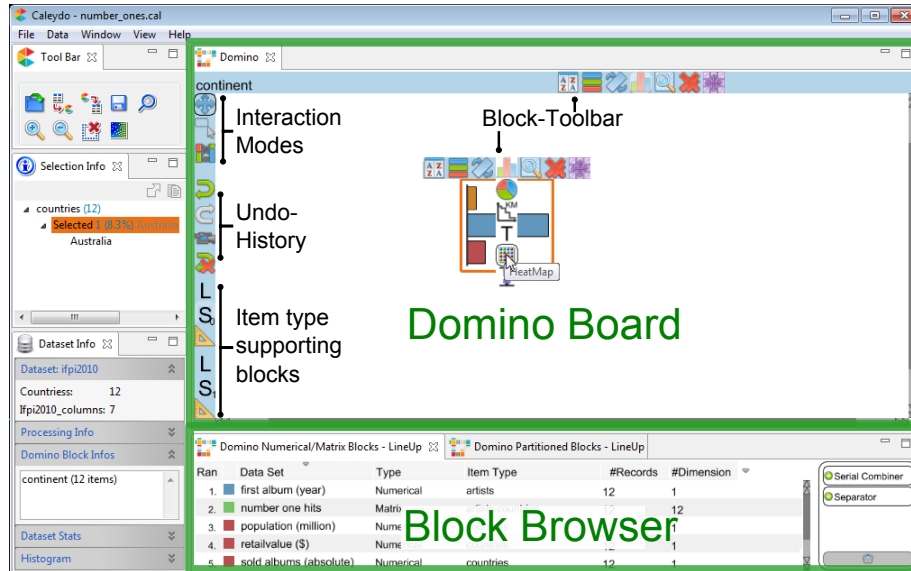


Figure 6.7.: Interface of the Domino prototype. The *Block Browser* presents lists of possible subsets organized by type. Users can drag subsets to arbitrary positions on the *Domino Board*, where they appear as a new block. Support views on the left show selections and meta-information.

The *Domino Board* is the central visualization and interaction space of Domino, in which blocks can be arbitrarily positioned, manipulated, and combined using drag and drop. To handle setups with many related subsets, the board is realized as an infinite canvas.

The *Block Browser* is a list-based interface for adding subsets to the board. Users can switch between two different lists: the first presents subsets that are associated with data (numerical and matrix subsets), and the second holds the partitioned subsets. Depending on the type of the subset, additional columns present meta-data, such as size and item type. Users can add blocks to the board by dragging entries from the list to an arbitrary location on the board. The block browser is based on the LineUp [GLG⁺13] described in Chapter 4.

Block-specific and global toolbars allow the analyst to trigger actions (e.g., sort, transpose, remove), switch between interaction modes (see Section 6.4.2), and provide access to a linear undo-history. In addition, item-type-specific supporting blocks are available, such as labels and rulers (Section 6.4.3).

6.4.1. Placeholders and Live Previews

When adding a new block to the *Domino Board*, the user needs to decide where to place it and how to represent it. To support this process, we make use of placeholders and live previews, as illustrated in Figure 6.8. Placeholders assist analysts in the process of combining blocks by indicating the various potential positions for placing a subset in the current Domino setup. When a user hovers over the placeholder, live previews [vdEvW13] of the possible visualization outcomes are presented to help the user choose the appropriate technique for the subset.

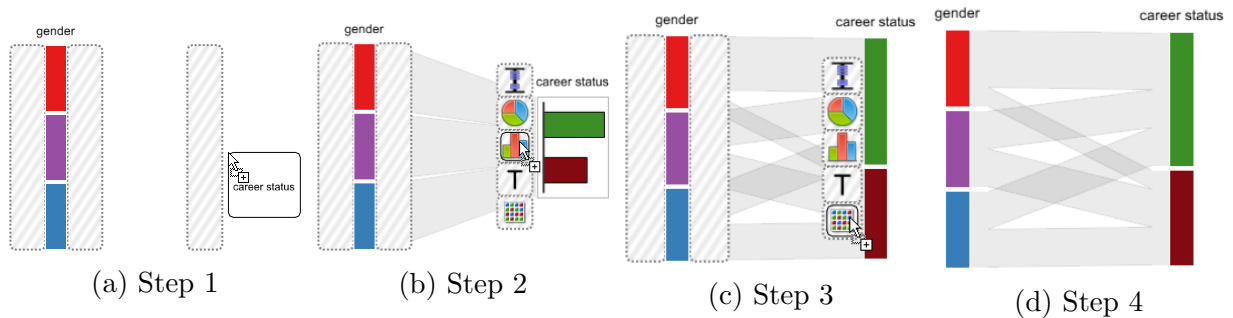


Figure 6.8.: Usage of placeholders and live previews to add new blocks to Domino. When a block is dragged, placeholders (hatched areas) indicate possible drop locations. Live previews help users to choose a visualization technique.

While a block is dragged, all possible drop locations with a strong or weak block relationship are highlighted, as shown in Step 1 of Figure 6.8. The proximity of the placeholder to the block indicates the relationship degree, as introduced in Section 6.2.2. When the user drags a block or subset onto a placeholder, the applicable multiform visualization techniques are presented (Step 2). Hovering over one of the icons representing the techniques renders a live preview of the resulting visualization (Step 3). Dropping the block onto a specific icon confirms the selection and removes all other placeholders (Step 4).

6.4.2. Interaction Modes

In our implementation of the Domino technique, analysts can switch between three different interaction modes by using the toolbar, as indicated in Figure 6.7. The changing background color of the toolbar area emphasizes the active interaction mode.

As blocks and their relationships are the primary visual elements in Domino, the **block interaction mode** is designed for adding, moving, combining, and removing blocks.



Figure 6.9.: Relationship interaction mode. a To reduce visual clutter, relationships between blocks that intersect with another block are culled. b By actively selecting blocks, users can reveal the otherwise hidden relationships. Relationships to unselected blocks are faded out.

In the **item interaction mode**, users can select and explore individual items within the blocks and within block relationships. Depending on the visualization technique employed, the analyst can directly select items, such as columns or rows in a heatmap, an aggregation of items represented by a bin in a histogram or bar chart, or parts of relationship representations.

Selected items are highlighted in all matching blocks and block relationships. While item block relationships highlight the whole matching line or point, the coarser granularities highlight only a portion, depending on the represented subset. In Figure 6.1, for example, we selected the artist *Whitney Houston*, highlighted in all blocks and block relationships.

Weak block relationships are shown for all blocks with matching item types. However, by default, a block relationship is culled automatically if it intersects with any other block or an artificially introduced separator. On the one hand, this avoids clutter and can be helpful to create independent sections on the board. On the other hand, this also prevents analysts from investigating block relationships of non-neighboring blocks. Therefore, the purpose of the **relationship interaction mode** is to let users explore these hidden relationships between blocks. In this mode, all block relationships between selected blocks are shown, regardless of whether they intersect with any other blocks. To reduce clutter, all unselected blocks are faded, as illustrated in Figure 6.9.

6.4.3. Scaling and Ruler

Accommodating a range of dataset sizes on various screen resolutions is challenging. Domino addresses scalability by its flexible zoom capabilities and initial scaling heuristics, which make blocks with various sizes fit on the board. In our implementation, blocks

can be scaled freely in the horizontal or the vertical direction. When the user hovers over a block while performing a scaling operation, only this block and strongly related blocks are affected by the operation; otherwise, all blocks on the board are scaled.

While the scaling approach ensures flexibility to deal with datasets of various sizes, it hampers the user’s ability to compare the sizes of multiple independent blocks, since they may have different scaling factors. To address this problem, we provide *rulers* that illustrate how much space on the board corresponds to how many items. They also serve as global scaling factors for a specific item type. In Figure 6.1, rulers for *artists* and *countries* are shown in the bottom left corner. When a scaling operation is applied to a ruler, all 1D and 2D heatmap and bar chart blocks are scaled simultaneously.

6.4.4. Subset Manipulation

Subsets can be created and manipulated in several ways. Set operations between blocks can be triggered by dragging a block onto another one. The desired operation (union, intersection, or set difference) is determined by dropping the block on an icon, similarly to choosing the visualization type as described in Section 6.4.1. Since relationships at the group and block granularity levels already encode set operations between two blocks, they can also be extracted into a new block by dragging the relationship representation to an open area on the board [KRL97]. In a similar fashion, individual groups of a partitioned block can be extracted into a new block. Finally, we use a small widget for extracting selected items as new blocks, as shown in the bottom left corner of Figure 6.1. The selected items of the corresponding item type are represented by a bar whose size corresponds to the number of selected items. By dragging the bar to an arbitrary position on the board, the selected items are added as a new block.

As introduced in Section 6.2.3, Domino distinguishes between temporal and spatial provenance. While spatial provenance is encoded directly by the Domino visualization, temporal provenance is implemented by storing all operations in a linear history. Users can undo and redo operations, revert all operations at once, and replay them as an animation.

6.5. Use Cases

We demonstrate the utility of Domino in two use cases. In the first scenario, we utilize our technique to explore the music charts dataset introduced at the beginning of this chapter (see Figure 6.1). In the second use case, we use Domino to characterize tumor subtypes.

6.5.1. Music Charts

At the beginning of the analysis, we want to know whether there are any exceptional correlations between the count of number-one hits in a country and the artists' country of origin. We therefore begin by adding the *number-one hits* matrix block and partition the artists by their country of *origin* while partitioning the countries by *continent*. To inspect individual items, we switch to the item interaction mode. Looking at the partitioned matrix block, we observe that the band *U2* has the most number-one hits in their home country *Ireland*, while *ABBA*, for instance, is far more popular in *Germany* than in their home country *Sweden*. To understand the relevance of a number-one hit in a specific market, we add the *sold albums* numerical blocks as a bar chart below the matrix block and observe that the *US* leads by a wide margin. However, we assume that the absolute number of albums sold depends on the population size of a country. To confirm this, we add the *population* of the countries as an additional 1D scatterplot.

Another interesting question we seek to answer is how the count of *number-one hits* depends on the count of published *studio albums*. By adding the corresponding numerical blocks, we find that *Elton John* has released an impressive 33 albums. When looking at this number in the context of the years of his *first album* and his *start of career*, it becomes obvious that he has been successful in the industry for a long time. By looking at the other artists, we see that only *Elvis Presley* published an album earlier than Elton John. However, as Elvis Presley (presumably) passed away, we continue to look for additional artists that are no longer active. According to the *career status* block, five out of twelve are inactive. To examine the inactive artists more closely, we correlate the status with the artists' *gender*. We further investigate the distribution of the gender in the inactive group by performing a logical intersection operation, which shows us that only a single female artist is no longer active: *Whitney Houston*. Figure 6.1 presents a screenshot of our analysis, in which *Whitney Houston* as well as the *US* are selected and highlighted across all blocks.

6.5.2. Cancer Subtype Analysis

A second dataset we studied using Domino is the glioblastoma multiforme (GBM) dataset generated by TCGA project and described in several publications by that consortium [BTe13, Net08, Vea10]. Glioblastomas are aggressive brain tumors, and patients with this type of cancer have a median expected survival time of around 12 months after diagnosis [Net08]. There are, however, different subtypes of GBM, which are driven by different molecular changes and associated with different patient survival times. A introduction into cancer subtype characterization is given in Chapter 2.

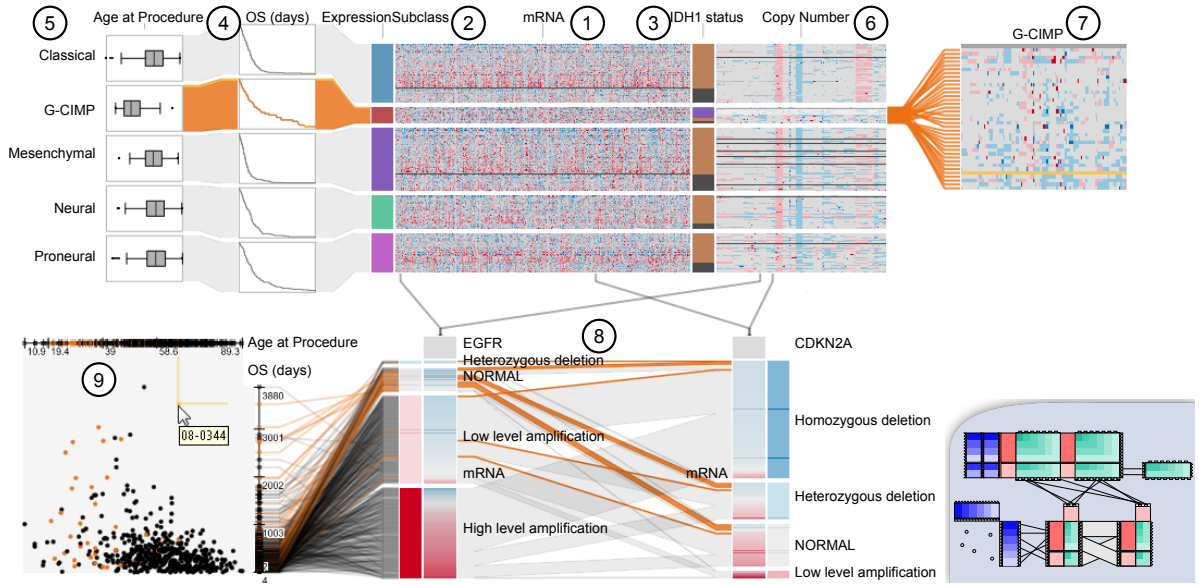


Figure 6.10.: Key findings on subtypes in glioblastoma multiforme. Orange highlighting indicates patients with the *G-CIMP* subtype. (1) The mRNA gene expression matrix. In this heatmap, red and blue indicate expression levels higher and lower than the cohort average, respectively. Gray represents the average. (2) Partitioning block representing the tumor subtypes defined by Brennan et al. [BTe13]. (3) The mutation status of the *IDH1* gene. Purple represents mutations; orange represents patients without mutation, and gray represents missing data. (4) Patient survival illustrated by Kaplan-Meier plots. (5) Patient age as boxplots. (6) Copy number for each patient and gene sorted by genomic location. Blue and red hues represent copy number losses and amplifications, respectively. Gray represents the normal state of two copies per gene. (7) Block extracted from the copy number heatmap to highlight uncommon copy number patterns observed in the *G-CIMP* subtype. (8) Copy number and gene expression levels for *CDKN2A* and *EGFR*. (9) Scatterplot showing the relationship between patient age and overall survival time.

Our goal for this case study is to recapitulate some of the findings of the most recent TCGA study on GBM [BTe13], which are presented in Figure 6.10. We start by adding the gene expression matrix (*mRNA*) to the board as a heatmap (1). This matrix contains data for over 12,000 genes and 528 patients, and each cell of the matrix corresponds to the activity of a given gene in the tumor sample from a given patient. Without clustering or sorting, the heatmap does not reveal any particular patterns. Using the gene expression subtypes published by Brennan et al. [BTe13], we partition the heatmap into five groups: *Classical*,

G-CIMP, *Mesenchymal*, *Neural*, and *Proneural* (2). We further combine the mutation status of the *IDH1* gene with the expression subtype block (3). Mutations in *IDH1* are known to play a role in GBM and for their association with the *G-CIMP* subtype. As Figure 6.10 illustrates, this relationship becomes immediately evident in Domino.

In order to compare patient survival times in the five gene expression subtypes, we drag the overall survival variable (*OS (days)*) from the block browser onto the Domino board and create a medium relationship between this and the partitioned gene expression matrix so that the survival times are partitioned by expression subtype (4). We select the Kaplan-Meier survival plot as the block visualization, which shows that the overall survival times for the *G-CIMP* group seem to be better than for the other groups. This is a known characteristic of both the *G-CIMP* subtype and the *IDH1* mutation [BTel3]. Another finding related to this subtype is that the patients in this group tend to be younger. We confirm this by visualizing the age of the patients in each expression subtype as a box plot next to the Kaplan-Meier plots, and observe a notable difference between the *G-CIMP* group and the four other groups (5).

Next, we study the copy number changes of genes in the patient genomes. We drag the copy number matrix for 560 patients and over 24,000 genes onto the board, creating a strong relationship between this block and the existing gene expression matrix block (6). Sorting the genes by their location within the genome reveals a number of common large scale copy number losses and amplifications in GBM tumor genomes. As the copy number matrix is partitioned by gene expression subtypes, it also reveals that the copy number pattern for the *G-CIMP* is remarkably different from the other groups. We study these patterns more closely by extracting and enlarging the corresponding block of all genes and *G-CIMP* patients, noting what appears to be two different types of copy number patterns in this patient group (7). Two genes that are known to be affected by copy number changes in GBM are the tumor suppressor gene *CDKN2A* and the oncogene *EGFR*. We extract these genes from both the copy number and the gene expression matrix and create a combined block for each gene (8). We find that the copy number of *EGFR* is in fact highly amplified in close to 50% of the patients, and that this copy number change has an impact on the corresponding gene expression levels. For *CDKN2A* we observe that frequently either one copy is deleted (*heterozygous deletion*) or both copies are deleted (*homozygous deletion*), with the expected effect on the gene expression levels.

We are particularly interested in the patients with the *G-CIMP* subtype of GBM. In order to see their copy number status and gene expression levels for *CDKN2A* and *EGFR*, we select this patient group in the partitioned gene expression matrix (*mRNA*). Due to the highlighting of the patients in all blocks and bands, it is evident that overall the *G-CIMP* patients tend to have no or low-level copy number changes in the two genes. This indicates

that the tumors of these patients are most likely driven by a mechanism that does not involve molecular alterations of *EGFR* or *CDKN2A*.

Finally, due to our earlier observation that patients in the *G-CIMP* group tend to be younger and have longer survival times, we are interested in whether there is an overall correlation between these two variables. We drag both the overall survival and patient age variables from the block browser onto the board. By visualizing them as axis plots with an orthogonal alignment, we create a scatterplot of the two variables (9). The scatterplot indicates that there might be a weak correlation between younger age and longer survival times. The selected *G-CIMP* patients are highlighted in orange, emphasizing the distinct age distribution in that group.

6.6. Discussion

Visualization Grammar and Templates Domino provides a comprehensive toolset to assemble both established visualization techniques and novel combinations. The concept could be generalized to a visualization grammar [Wil05] for the exploration of tabular datasets. The grammar could be utilized to define templates that serve as a blueprint for the creation of sophisticated visualizations. This would make it possible to customize a specific implementation of the Domino concept according to the template. In this approach, the feature set of Domino could be restricted to certain operations in order to allow users to create only specific kinds of composite visualizations, such as parallel coordinates or parallel sets, or more complex ones, depending on the task. Furthermore, a visual editor could be created that enables users to interactively define templates for specialized visualizations.

Use of colors Assigning unique colors to partitioned blocks is a challenging task. By default, we automatically assign colors by using predefined color schemes provided by ColorBrewer [HB03]. However, users can manually override the chosen colors if desired. The automatic color selection guarantees that groups belonging to the same partitioned block have different colors. However, depending on the number of partitioned blocks and contained groups, it might not be possible to choose different colors for all groups in all partitions. Assignment of the same color to semantically unrelated groups can cause confusion. A compromise would be to at least ensure that colors within a combined block are unique. However, this could result in situations where the color assignment to blocks needs to be adapted after the user interactively changes the configuration of combined blocks. As this change of colors would destroy the user's mental map, we opted to allow duplicate colors instead.

Crossing lines and bands As defined at the end of Section 6.2.1 on blocks, the item order within a partitioned block is undefined. When visualizing relationships between blocks, this can lead to unnecessary line crossings between two weakly related blocks that use the parallel item block relationship granularity. In the current implementation, one has to strongly relate a copy of the opposite block and apply it as secondary sorting criterion. This will reduce line crossings, since the item order within a group will match the item order of the opposite block. To remedy this issue, more sophisticated reordering operations could be applied [PGU12, PWR04]. Relationships at the group granularity level can also suffer from a similar ordering problem, resulting in crossing of bands. Again, reordering strategies need to be applied to minimize the crossings.

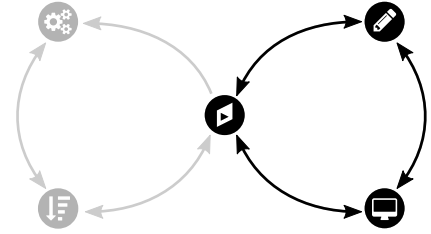
Partitioned matrix blocks The presented technique introduces three different block types: partitioned, numerical, and matrix blocks. Matrix blocks can be treated as the two-dimensional version of a numerical block. Therefore, a logical consequence would be to also introduce an analogous two-dimensional block type for a partitioned block. However, a partitioned matrix block is currently not supported in Domino, as the mapping of an item pair ($f((i, j) \in I_1 \times I_2) \mapsto g \in P$) to a group would result in a biclustering [CC00] in which one item of I_1 can be mapped to multiple groups. This prohibits splitting up the block horizontally or vertically, which is needed for propagating the partitioning to strongly related neighboring blocks. However, in Domino such scenarios can be handled by representing every group as a single block in the layout, where overlapping items between blocks are visualized using lines or bands [SGG⁺14].

Scalability and Performance We ensure visual scalability through the flexible zooming concept described in Section 6.4.3. Regarding data scalability, the size of matrix blocks is the critical issue. In the cancer subtype analysis case study presented here, matrix blocks have up to $560 \times 24,000$ items, demonstrating Domino’s applicability to larger datasets. Item relationships for large subsets are, however, a performance issue, since individual lines for each data item need to be drawn. This issue can be addressed by using a higher relationship granularity.

An alternative approach is using a focus+context concept in which detailed information about the relationship is provided on demand. An example is *LayerCake* [CBS⁺15] which is visualization tool for large scale comparison of sequencing data. Encoding relationships between two or more dimensions in an interleaved visualization is another possible approach to tackle scalability. Alexander and Gleicher [AG16] propose a visualization technique for the large scale comparison of topic models in document collections by encoding relationship to topic models in horizontal position and color of a document visualization.

6.7. Conclusion

We have presented Domino, a novel technique that visualizes subsets together with associated data, and relationships between them. Domino allows users to explore, extract, and manipulate subsets of multiple item types at various levels of granularity. Analysts can not only rapidly assemble common visualization techniques, but also create new combinations tailored to specific tasks and datasets. The prototype implementation uses live previews and placeholders to support and guide users in managing the wide range of possibilities that the technique offers.



7 | CLUE

From Visual Exploration to Storytelling and Back Again

Contents

7.1. Introduction	91
7.2. CLUE Model	93
7.3. Realizing the CLUE Model	95
7.4. Usage Scenarios	100
7.5. Discussion	103

The primary goal of visual data exploration tools is to enable discovery of new insights. To justify and reproduce insights, the discovery process needs to be documented and communicated. Common approaches are capturing visualizations as images or videos. However, they have several drawbacks. Most importantly, neither approach provides the opportunity to return to any point in the exploration in order to review the state of the visualization in detail or to conduct additional analyses. This chapter presents CLUE (Capture, Label, Understand, Explain), a model that tightly integrates data exploration and presentation of discoveries. CLUE resembles the right part of the SPARE model, as introduced in Section 1.2. Based on provenance data captured during the exploration process, users can extract the key steps, add annotations, and author “Vistories”, visual stories based on the history of the exploration. These Vistories can be shared for others to view, but also to retrace the original analysis and extend it. After introducing the CLUE model the realization of this model is discussed and its applicability shown in two usage scenarios, available at <http://gapminder.caleydoapp.org> and <http://stratomex.caleydoapp.org>. This chapter concludes with a discussion of the current limitations.

7.1. Introduction

Scientific progress is driven by discoveries based on observations. Accurate and efficient documentation and presentation of how discoveries were made is essential, since the scientific method requires that findings are reproducible. The process from making a discovery in a visualization tool to communicating it to an audience is typically a process that does not allow users to switch back from presentation to exploration, as illustrated in Figure 7.1.

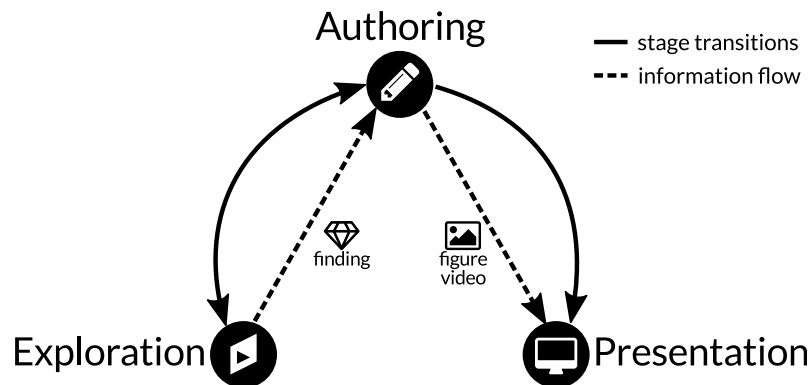


Figure 7.1.: Traditional workflow and information flow for visual data exploration and presentation of discoveries. Dashed edges indicate information flow, solid edges show transitions between stages. The information flow is sequential and different tools are used in each stage.

In the exploration stage of a data-driven research project, analysts apply interactive visualization and analysis tools to gain new insights. Then they document the discovery process for reproducibility and presentation. Presentation can be in the form of text or figures in a paper or slide-deck, or in the form of interactive visual data stories. Visual data stories are increasingly popular as they are engaging and can communicate complex subject matters efficiently [LRIC15]. Only in rare cases, however, can static figures or visual data stories be created straight out of the exploratory tool. Instead, an authoring stage, in which artifacts, such as screenshots, are sorted, edited, and annotated is necessary. In the case of interactive data stories, this process often consists of custom development of software. The final product is subsequently used to present a finding to a consumer. A consumer in this context can be, for instance, a reader of a news article, a reviewer or editor of a scientific publication, or a colleague of the analyst, trying to understand a finding and the process of its discovery. This workflow corresponds to the storytelling process introduced by Lee et al. [LRIC15]. The authoring stage described here includes their “make a story” and “build presentation” stages, since the story being told is a scientific discovery not requiring a distinction between scripter (author) and editor.

In a visual exploration process, findings are often captured by taking one or multiple screenshots of the visualizations, or by creating a screen recording that shows the steps that led to the discovery. Static images, however, cannot tell the story of the visual discovery, as they cannot convey information about the exploration process. Videos are difficult to create, edit and update, and also do not capture the full analysis process. Furthermore, the tools used for exploration are in many cases not suitable for authoring and presentation. Neither images nor videos allow an exploration to be continued, and both prohibit users from asking additional questions. Given the sequential information flow and the separation of tools, it is inefficient for the analyst and creator of the story – and even impossible for the consumer – to work back from an artifact used for presentation to the exploration stage. The lack of a back-link from the curated story to the exploration stage and the underlying data makes it impossible (1) to reproduce and verify the findings explained in a figure or video and (2) to extend an exploration to make new discoveries. In this work, we propose a comprehensive set of solutions to these problems.

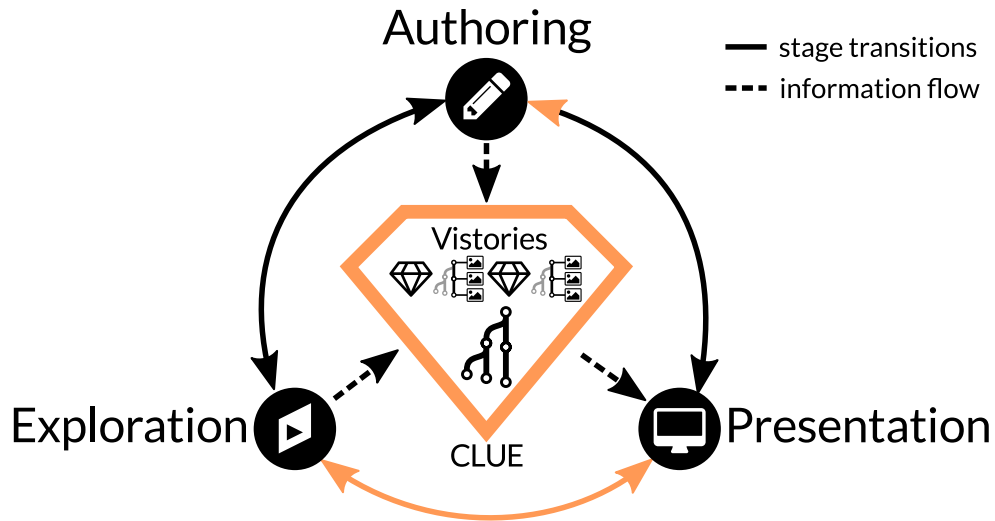


Figure 7.2.: Information flow and stage transitions using the CLUE model. The provenance graph of an exploratory session and Vistories (interactive visual stories) are in the center. Solid edges indicate possible stage transitions, dashed lines indicate information flow. In the exploration stage, provenance data is generated; in the authoring stage a Vistory is created by curating provenance data, which is then used in the presentation stage. Note that consumers of a Vistory can also switch to any other stage.

This chapter presents CLUE, a model for reproducing, annotating, and presenting visualization-driven data exploration based on automatically captured provenance data. In addition, it introduces Vistories – interactive visual data stories that are based on the

history of an exploratory analysis that can be used as an entry point to reproducing the original analysis and to launching new explorations. Figure 7.2 shows our proposed CLUE model. All information flow is routed through a central component, and all stages use the same universal tool. This allows the seamless stage transition indicated by the solid edges.

In addition, this chapter presents a prototype implementation and a discussion of how to integrate CLUE with other visual exploration tools. To demonstrate the overall CLUE model, we describe a Gapminder-inspired usage scenario based on public health data. In a second usage scenario, we apply CLUE to a more complex multi-step visual exploration of cancer genomics data.

7.2. CLUE Model

The CLUE model bridges the gap between exploration and presentation. Its backbone is a rich provenance graph that contains all actions performed during the exploration. This includes exploration paths that led to findings, but also the dead ends encountered by the analyst. By putting the provenance graph at the center of our model, we are able to break the strict sequential order of the exploration, authoring, and presentation stages (see Figure 7.1) that dominates traditional workflows. CLUE allows users to seamlessly switch between **Exploration Mode**, **Authoring Mode**, and **Presentation Mode**. This integrated and flexible process is illustrated in Figure 7.2, where solid lines indicate the possible transitions between stages, and dashed lines represent information flow.

Provenance data makes scientific findings more reproducible and can also provide the basis for authoring stories. In CLUE, users create stories by defining a path through the provenance graph. We call such a provenance-based story a ***Vistory***. States in the story can then be enriched with highlights, textual annotations, and, if desired, timed for automatic playback. Hence, the resulting story is not an artificial composition of visualizations, but a curated version of the actual exploration. Most importantly, this deep integration of the provenance graph introduces the back-link from presentation to exploration. Vistories can be shared and encourage collaborative visual data analysis. Consumers can step through a story, but also switch to the exploration mode and interactively build upon the previous analysis to gain new insights. Vistories also make the exploration process more efficient, as a user can revisit states and apply changes, without redoing all steps to reach a particular state. Therefore, Vistories are more than visual data stories as defined by Lee et al. [LRIC15], since they allow consumers to continue the visual exploration in-place and build new stories themselves.

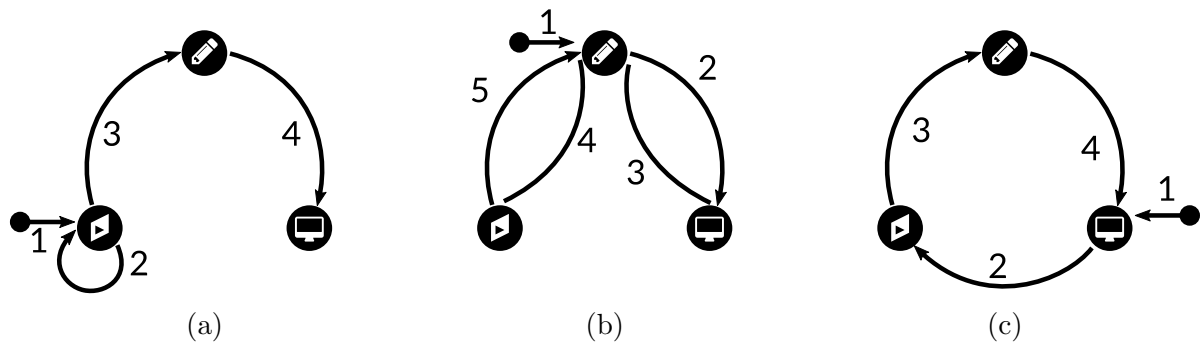


Figure 7.3.: Three examples of transitions within the CLUE model, highlighting different entry points. Numbers indicate the order in which the stages are visited by the user.

Figure 7.3 illustrates three scenarios that show how users can switch between modes. In the first example (Figure 7.3a), the process starts by investigating the data in exploration mode. After several iterations, the analyst discovers a finding worth presenting and switches to authoring mode to create a Vistory. Finally, the analyst previews the Vistory in presentation mode and makes it available to others. In the second example (Figure 7.3b), an editor creates a Vistory in the authoring mode, starting with an existing analysis session. After previewing the story in presentation mode and continued editing in authoring mode, the editor notices that the content of the story could be improved, and switches back to exploration mode in order to refine the visualization. Subsequently, the user returns to authoring mode and finishes the Vistory. In the last example (Figure 7.3c), a consumer starts by watching an existing Vistory. In that process, the consumer becomes curious about the consequences of adding another dataset to the analysis. The consumer switches to exploration mode and picks the relevant state, from where she can start her own analysis. Based on this new analysis, she creates a new Vistory that can be shared with collaborators.

These simple workflow examples illustrate that users can enter the process in any mode and switch freely between modes. Note that, from a conceptual point of view, switching from exploration directly to presentation is possible without going through an authoring step. While this can be useful when the only goal is to reproduce an analysis, authoring is required as a transition between exploration and presentation in practice to create an informative and concise story.

7.3. Realizing the CLUE Model

This section demonstrates the practical use of the CLUE model. We discuss how visual exploration tools can be extended by adding provenance capturing, authoring, and presentation capabilities. We also describe a prototype library for capturing and managing provenance and story data as well as a prototype that demonstrates the flexibility and efficacy of our model.

The library is used by a *Visual Exploration Application*. It is important to note that the Visual Exploration Application is not restricted to a specific visual exploration tool, but only has to comply with a set of basic requirements (see Section 8.4.2) and make appropriate calls to the library. The application is shown in all CLUE modes, although it is set to read-only during authoring and presentation.

The CLUE library consists of several building blocks. At its core is the provenance graph data model that forms the back end of CLUE that is used to store the captured exploration process. The other important components of the library are the provenance view and the story view.

The *Provenance View* provides a scalable visualization of the provenance graph. In exploration mode, a simplified version of the graph gives the analyst an overview of the provenance, by showing, for example, the states leading to the current one. When the system is in authoring mode, the provenance view is used for navigating and selecting the states of the exploration process that should be part of the story.

The *Story View* visualizes the elements of the story. Depending on the selected mode, the view shows different features. In presentation mode, it is essentially a stepper interface for the curated story. When the system is in authoring mode, it enables users to create, manage, and edit stories.

The visual components can be active in more than one mode and show different levels of detail depending on the mode. A switch between modes results in the addition, adaption, or removal of certain parts of the user interface. Animated transitions are applied to support users in maintaining their mental model during mode changes. In addition, the current exploration state along with the mode used are encoded in the URL of the visual web application. This allows users to conveniently share states and Vistories by exchanging links.

7.3.1. Provenance Graph Data Model

The provenance graph data structure used in CLUE consists of four node types: *state*, *action*, *object*, and *slide*. Figure 7.4 illustrates the relationships between the different node types. An *action* transforms one *state* into another by creating, updating, or removing one or multiple *objects*. A *state* consists of all *objects* that are active at this point of the exploration. A *slide* points to a state along with annotations and descriptions explaining the state. A *Vistory* is made up of a sequence of slides. Switching between slides triggers actions that transition to the state associated with the target slide.

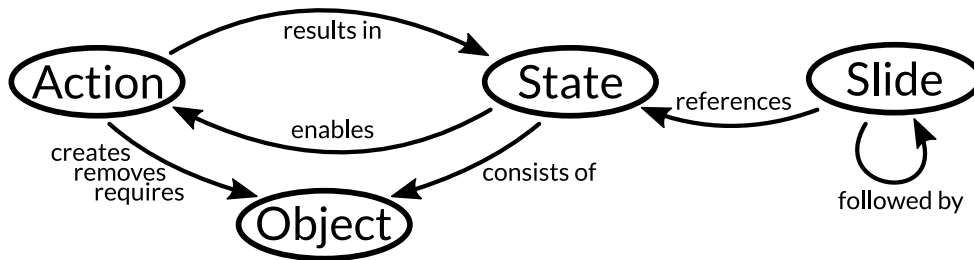


Figure 7.4.: The provenance graph data model consists of four different node types that are connected with each other by one or more edges.

Object and action nodes are generic, and refer to the application-dependent implementation. In order to improve the visualization, additional meta-data about objects and actions is stored. For objects we also store a *type*. We distinguish five types: data, visual, layout, logic, and selection. For actions we also store an *operation*: create, update, and remove.

Data actions deal with the addition, removal, or subsetting of datasets within the application. An example of a data action in our Gapminder usage scenario is the assignment of an attribute to an axis of the plot. Visual actions (e.g., switching an axis to logarithmic scale) manipulate the way datasets are shown to the user. Layout actions manipulate the layout of a visualization (e.g., manipulating the axes order in parallel coordinates or hiding the categorical color legend in a Gapminder plot). Logic operations, such as triggering a clustering algorithm, are concerned with the analytical aspect of the applications. Finally, selection actions encompass user-triggered selections of data in the visualization.

A state is characterized by the sequence of actions leading to it. Therefore, restoring a state is achieved by executing its corresponding actions. Jumping from one state to another is implemented by reverting the actions from one state to a common ancestor and executing the actions necessary to reach the target state.

In addition, for the purpose of transitions, the action sequence is compressed before being executed by removing redundant actions. A sequence of five selections, for example, will be replaced by the last one, since all the others are just intermediate states that do not influence the final selection. Similarly, when a create-and-remove action is associated with the same object, we remove both actions, as they neutralize each other. This compression avoids the execution of superfluous actions.

7.3.2. Provenance Visualization

As provenance graphs grow quickly during the exploration process, it is challenging to develop an effective visualization for it. Our provenance visualization is based on a node-link tree layout that we combine with a Degree-of-Interest (DoI) function to adapt the detail level of nodes [Fur86]. An example of the provenance visualization for one of our usage scenarios can be seen in Figure 7.7(c), and a close-up in Figure 7.5.

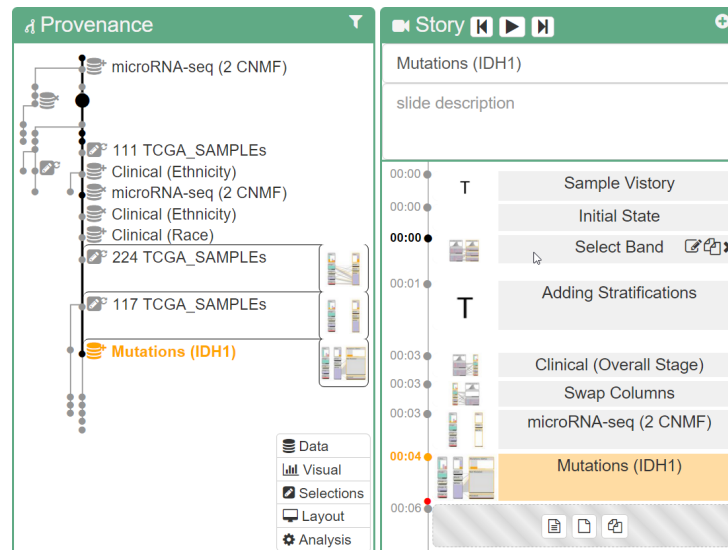


Figure 7.5.: Close-ups of the provenance and story views. Based on the DoI, exploration states are represented at different levels of detail in the provenance view. The structure of the Vistory, shown on the right, corresponds to a path through the provenance graph, which is shown as a thick black line in the provenance visualization. Both the active slide in the story view and the associated exploration state in the provenance graph are highlighted in orange.

Layout We use a vertical node-link layout as the basis for the provenance graph visualization. Nodes represent states, while edges represent actions transforming one state into another. However, instead of using a plain balanced tree, we reorder and skew it such that the currently selected node and all its ancestor nodes in the path to the root are aligned vertically on the right side. The remaining nodes and branches are then lined up on the left side. This strategy leaves space for details of the selected nodes and their ancestors, including labels describing the state and thumbnails previewing the state of the visualization. However, the layout needs to be updated when the user selects a different state. We use animated transitions to convey such changes.

Detail Level We assign each node a DoI value that is influenced by several factors, including the current state selection, whether the state is an ancestor of the selected state (the distance term of the DoI function), and several filtering options that can be defined by the user (the a priori interest term of the DoI function). The DoI computed for each node is then used to adapt its representation with a combination of semantic and geometric zooming. We distinguish between four levels of increasing node detail.

Level 1: the state represented as a bullet

Level 2: icon describing the action associated with this state

Level 3: label describing the action associated with this state

Level 4: thumbnail of the application at the given state

Nodes of all detail levels are shown in Figure 7.5.

Interaction Users can interact with the provenance graph visualization in several ways. Selecting a node will show the corresponding state in the application. At the same time the selection of a node triggers a re-layout of the provenance graph visualization, since the DoI values of the nodes change according to the selection.

The user can bookmark a state for later use, add tags, or add notes. Additional meta-data can then be used to filter the states in authoring mode, which enables a more efficient story editing process.

To ensure reproducibility, it is important to be able to prevent a user from modifying the provenance graph. However, when authoring, the editor might want to improve a previous state, for example, by changing an axis scale from linear to logarithmic. Rather than allow the user to change existing states, we create a new branch. However, when

starting a new branch, a user would need to redo all other actions that came after that was previously updated. We address this problem by allowing the user to apply a subbranch of a provenance graph to any other state. By dragging one node onto another, the actions are replayed automatically if possible. An example where this is not possible is when the user seeks to manipulate an object that has already been removed in one of the earlier states.

The consequence of preventing users from modifying the provenance information is that the graph grows rapidly. Although our provenance visualization supports multiple detail levels, the design and implementation of a truly scalable provenance visualization was not the main focus of CLUE and is therefore open for future research.

7.3.3. Story Editor

A story is composed of a sequence of slides. We distinguish between two slide types: *text slides*, which contain introductory text and captions, and *state slides*, which are associated with a specific state in the provenance graph. Both slide types can be annotated using multiple methods, such as styled text, scalable arrows, and freely adjustable boxes.

Layout We use a vertical layout to present the slides, where we use the y-axis as a pseudo-timeline. The higher a slide, the longer it will be shown when automatically playing the story. Similarly, the more space between two slides, the longer the transition between them. Both transition and slide duration can be manipulated by dragging the top and bottom border lines of the slide, respectively. We chose a vertical layout because of (1) the alignment of the story with the provenance graph, and (2) the better readability of horizontal labels.

Interaction Vistories can be created and edited in various ways. Editors can (1) start with a default text slide, which is useful if they already have an idea about the story they want to tell, (2) extract the currently selected state and all its ancestors, or (3) extract all bookmarked states.

Individual slides can be rearranged using drag-and-drop. In addition, dragging one state node in the story editor will wrap the state in a state slide when dropped, which allows a user to quickly create complex stories. Note that individual state slides need not be in sequential order in the provenance graph. The system automatically resolves the path between the states and plays all necessary actions, as discussed in Section 7.3.1. We

indicate the currently active story in the provenance graph by connecting its states using a thick line. The provenance graph is also fully linked with the story editor: selecting a state slide in the story editor highlights the corresponding state in the provenance graph, and vice versa.

Annotations Each slide can be enriched with annotations that are shown as an overlay on top of the visual exploration application. The library currently supports three different annotation types: text, arrow, and box. All of them are available in the movable annotation toolbar, which appears top left when a slide is selected in authoring mode. Figure 7.7(b) contains examples of all annotation types in blue. Annotations are positioned relative to anchor points in the visual exploration application. Anchor points represent important visual elements in the application such as data points of the scatterplot in Gapminder. By linking annotations to anchor points, their positions can be better adapted to layout changes due to different screen resolutions and aspect ratios.

7.4. Usage Scenarios

We demonstrate the utility of CLUE for a variety of applications in two usage scenarios. The first is inspired by Hans Rosling’s Gapminder¹. It illustrates the workflow of how users interact with and switch between different CLUE modes. The second scenario reproduces parts of our recent Nature Methods publication about Guided StratomeX [SLG⁺14], a tool for characterizing cancer subtypes also described in Chapter 5. It highlights CLUE’s reproducibility support and its applicability to scientific analysis and storytelling. We provide links to the interactive Vistories for both usage scenarios below the respective figures.

7.4.1. Gapminder

A historian based in Europe is interested in assessing the interplay between wealth and health over the last 215 years. In particular, he would like to visualize changes in European countries to present his findings to a colleague in America. To explore health versus wealth, he first assigns income per person to the x-axis and life expectancy in years to the y-axis. The size a mark in the scatterplot corresponds to the size of the population of a country for the currently selected year. Continents are color-coded; Europe is shown in Purple,

¹<http://www.gapminder.org>

America in red, Africa in blue, and Asia in brown. He applies a linear-to-log transformation to the wealth data and is ready to explore. Moving the slider on the timeline from 1800 to 2015, he observes an overall trend for Western countries: when wealth increased, people lived longer. The populations of most African countries, however, continue to have low GDPs and low life expectancies, as shown in Figure 7.6.

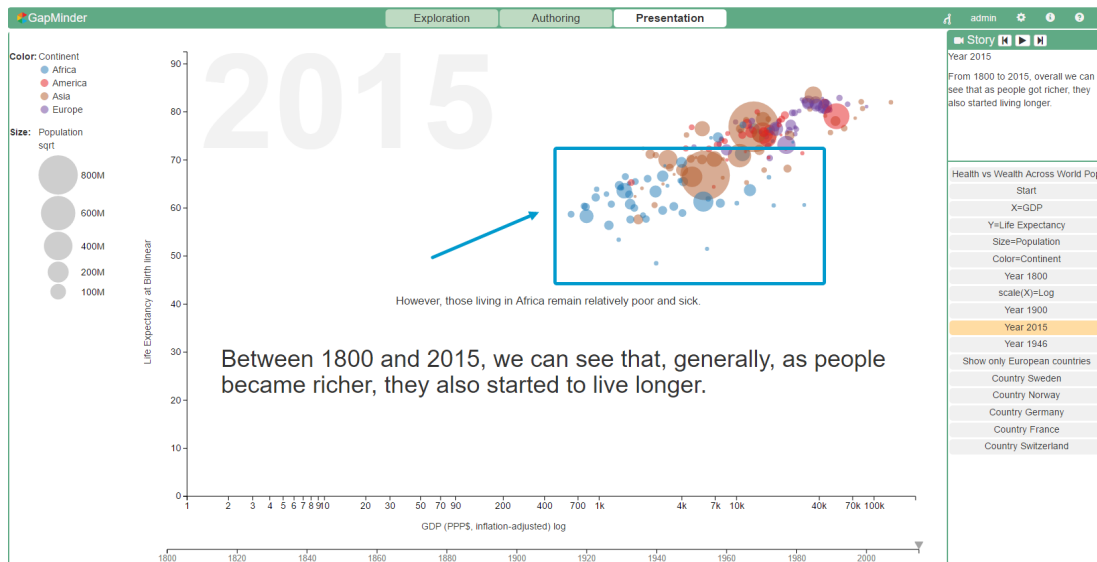


Figure 7.6.: Screenshot of a Gapminder-inspired application illustrating a Vistory in presentation mode. African countries are highlighted using annotations.

Vistory: <http://vistories.org/v/gapminder>.

To create a Vistory based on this finding, he switches to authoring mode, where he extracts all states captured in the provenance graph. In the story editor, he annotates the states using the text and arrow annotation tools and previews the story by clicking on the play control. Now, he would like to take a closer look at health and wealth in Europe in particular, so he switches back to exploration mode, where he uses the continent legend to select Europe from the chart. As a result, all countries in other continents are shown with reduced opacity. He goes to a pivotal year in European history: 1946, right after World War II. He evaluates the state of Europe regarding health and wealth, and extracts his findings to the story editor. In the story editor, he adjusts duration times and annotates key findings, moves to presentation mode to review, and finally shares a link to his Gapminder Vistory with his American colleague.

The American historian views the Vistory in presentation mode and decides that she would like to look at the state of Africa at the points chosen for Europe. To do so, she switches to exploration mode. Here she selects the node in the provenance graph where her European

colleague initially selected Europe, and selects Africa instead. Her exploration is now captured on another branch of the provenance graph, and she is free to extract her own Vistory. To reproduce the analysis done for Europe for her subset, she applies the original subbranch to her new branch.

7.4.2. StratomeX

We adapted Guided StratomeX [SLG⁺14] presented in Chapter 5 and extend it with our CLUE model. In the following section, we describe how CLUE can be used to partly reproduce use cases published in [SLG⁺14]. The initial exploration was illustrated in supplementary figures and a video. However, the video shows only the final curated story and not the exploration as a whole.

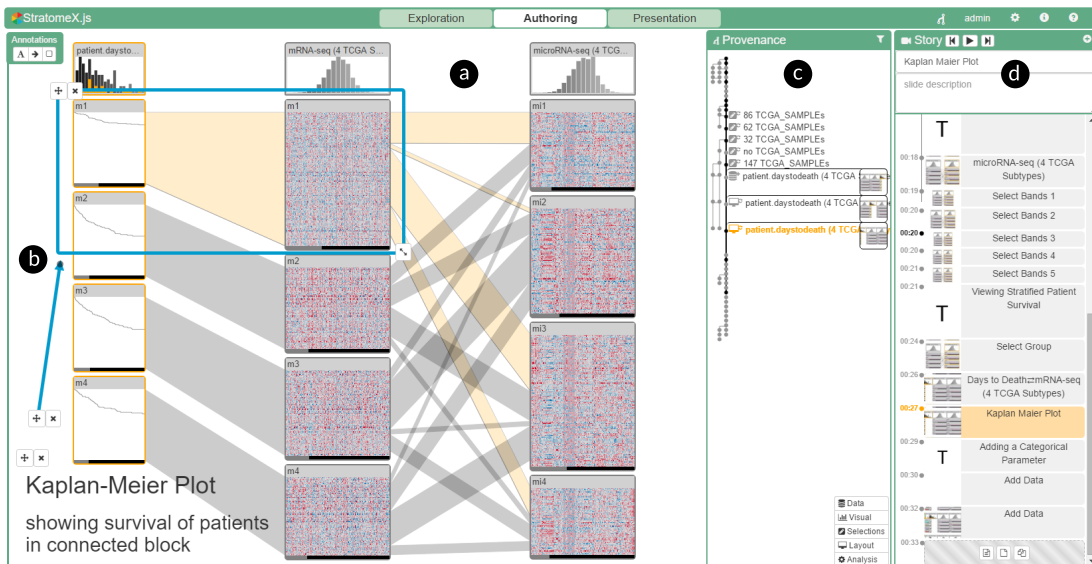


Figure 7.7.: Screenshot of CLUE applied to the StratomeX technique (a) in authoring mode. An annotation (b) highlights relevant aspects. The provenance graph view (c) and story view (d) show the history of the analysis and a Vistory being created.

Vistory: <http://vistories.org/v/stratomex>.

We started to reproduce the case study by following the figure captions and the video step by step. This included frequent switching between exploration and authoring mode, while simultaneously creating the corresponding Vistory. In the end, we successfully reproduced Supplementary Figures 6(a) and 6(b) of the original paper. Figure 7.7 shows an inter-

mediate screenshot of StratomeX in authoring mode, illustrating a central aspect of the technique. A similar picture is part of the original video. The left side shows StratomeX with annotations. On the right, the provenance graph of this analysis and the story view are shown.

With this CLUE-based implementation of StratomeX we were able to re-trace an analysis from a published paper and make this analysis reliably reproducible as a Vistory. This has all the benefits of the original research video, but was much easier and faster to produce. Moreover, consumers of the Vistory can go back to the analysis and, for example, check for confounding factors by adding other datasets, or start their own analysis to look for new findings. The Vistory is accessible at <http://vistories.org/v/stratomex>.

7.5. Discussion

Separation of Concerns The CLUE model contains three different modes: exploration, authoring, and presentation. Since each mode has a different focus, only the relevant information and visual elements relevant to the current mode are shown in the prototype implementation. The provenance graph visualization, for instance, is shown in a simplified version when in exploration mode, in full detail when in authoring mode, and not at all when in presentation mode. A different approach would be to show all views and elements at once and let the user decide which elements are useful for a specific task. Wohlfart and Hauser [WH07], for example, used such a unified interface. However, overwhelming the user with all possibilities can be distracting. In CLUE, we decided to reduce the elements in the interface by introducing three separate modes for exploration, presentation, and authoring, therefore making as much screen space as possible available for the data visualization.

One Tool for the Whole Process Lee et al. [LRIC15] raised the important question of whether one tool combining exploration, authoring, and presentation features is suitable and desirable. While Lee et al. state that it might be a promising endeavor, they also have concerns about it. The key question is whether a unified tool can cover all potential analysis needs. Creating a tool that allows all kinds of visual explorations is indeed challenging. We tackle this challenge by providing a library that existing visual exploration tools can use. However, we also consider adding options to import images, videos and websites into Vistories, so that the presentation can be complemented by the output of incompatible tools. For these parts of a Vistory, however, we will not be able to provide a back-link to the analysis.

Collaboration Collaborative visual data exploration is a relevant and promising research direction. CLUE captures all actions performed during a visual exploration on a semantic level. Hence, extending this approach such that multiple users can perform an analysis based on the same provenance graph is a logical next step. In our current implementation, we support sequential collaboration, that is one user at a time can perform the analysis, but the user can change over time. However, our ultimate goal is to enable synchronous collaboration. This, will introduce several additional challenges, such as synchronization issues, or the visualization of such multi-user provenance graphs.

When introducing user management, we will also be able to restrict the operations allowed on a Vistory. For example, one could prohibit modification of a published Vistory and only allow a fork to be modified.

Full Provenance The current implementation of the CLUE model captures the steps carried out by the analyst during visual exploration. However, the datasets used during exploration, tools employed outside of our application, and which version of an application is used is not tracked. This information would be needed to truly reproduce every state of the exploration. The versioning of datasets, tools, and applications, however, is a subject of active research in other fields. For example, source code management systems such as Git and Subversion work well for all kinds of text files. In the biomedical domain, platforms such as Refinery [GPS⁺14] capture the provenance for the execution of workflows with all input and output data. Approaches based on Virtual Machines and Docker² can be used to capture application versions. In the future, we plan to integrate all versioning approaches mentioned into a comprehensive solution that then allows full provenance tracking for data-driven visual exploration.

Meta-Provenance CLUE currently captures only the visual exploration itself. All actions performed by the user in the authoring and presentation mode are not tracked. As users can jump to different branches of a provenance graph during the analysis, the sequence of actions performed by the user can only be reliably reconstructed via the timestamps of actions. An interesting future research direction is therefore to track the provenance of how the provenance graph was created. Capturing this meta-provenance graph would allow us to analyze the process of visual exploration and also the evolution and use of the CLUE model, including how stories evolve and how users collaborate.

²<http://www.docker.com>

Scalability Since provenance graphs grow very quickly, scalability is an inherent problem. To mitigate the scalability issue, the provenance graph visualization represents only states as nodes and actions as edges between them while hiding object nodes. We found this to be intuitive, since users tend to think in states. We found our DoI approach to be useful for managing medium-sized provenance graphs, but we realize that for large provenance graphs of complex analysis sessions, additional methods will need to be developed. One aspect we plan to investigate are user-specified states of interest that can influence both the visualization of provenance graphs and the selection of states for stories.

State Selection An important question when creating a story is which intermediate states to select given a target state. A simple approach is to use the path from the start of the exploration to the desired target state. This, however, may contain superfluous states and leads to long stories. Automatically identifying key states is challenging. While there are certain measures one could take, such as removing intermediate selections that were not pursued further, these assumptions are invalid in the general case. Manual annotations or bookmarks of states during an analysis are strong indicators for the relevance of states. These can be leveraged by suggesting key states for authoring and for emphasizing them in the provenance graph. We plan to investigate methods for encouraging users to externalize their assumptions and reasoning, which is also important with respect to reproducibility.

Animated Transitions Animated transitions are effective for communicating changes between different states. However, the CLUE library is independent of the visual exploration tool used. Therefore, a story can only suggest the duration of an animation between states, but the actual application must decide how the timing options are interpreted. Moreover, moving from one state to another can involve a series of actions. Currently, they are executed sequentially, but some of them could be executed in parallel in order to speed up the transition. Detecting independent actions, how they can be executed in parallel, and whether this can improve user understanding remain open research questions.

8 | Phovea

An Integrated Visual Analysis Platform for Biomedical Data

Contents

8.1. Key Aspects	107
8.2. Related Platforms	108
8.3. Ecosystem	109
8.4. Additional Libraries	112
8.5. Discussion	113

In many scientific domains data analysis has replaced data acquisition, generation, and storage as the main challenge. This challenge stems not only from the volume but also from the complexity and heterogeneity of the data. Molecular biology is a prime example for this trend, where large initiatives like *The Cancer Genome Atlas* project and emerging technologies such as single cell gene sequencing produce vast amounts of heterogeneous data. However, with datasets from different sources, with different meanings, on distinct levels of scale, and of various types (tables, text, graphs, etc.), there is the need for new visual analysis platforms that tackle these new challenges. We identified six key aspects that a visual analysis platform for biological data should support. We address these aspects with the development of *Phovea*. This chapter describes its ecosystem and architecture. Phovea is open source under the BSD license and hosted on <https://github.com/phovea>. A collection of applications based on Phovea is available at <http://caleydoapp.org>. In addition, some of them are briefly described in Appendix A.

8.1. Key Aspects

From our experience with domain collaborators and designing visualization systems in the past [LSKS10, LSS⁺12] we identified six key aspects that a visual analysis platform for biological data needs to support:

A I: Data Scale and Heterogeneity Not only is the size of datasets increasing, there is also a growing number of publicly available datasets that researches want to integrate. Taken together, we observe that the size, complexity, and heterogeneity increases beyond current analysis and visualization capabilities. The data spectrum ranges from clinical and expression data, over epigenetic data, to full genome sequence information. Challenges include accessing, processing, and interactively visualizing the data.

A II: Identifier Management An important aspect when integrating datasets from various sources is the mapping of identifiers between different annotation systems (e.g., Entrez, DAVID). Mappings, however, can be 1:1, 1: n , n : m , or even more complex if they are based on partially overlapping gene locations. Also, entities of different types (e.g., gene, protein, samples) that can be defined on different levels of granularity (e.g., chromosome, gene, base pair) lead to additional challenges.

A III: Multiple Coordinated Views (MCV) The integrated analysis of multiple interconnected datasets can lead to new insights, yet it is often sensible to show different datasets as independent views, as the visualization can then be chosen to best represent the data. The coordination of these views provides the links between the datasets. The MCV system needs to visually link the entities across the annotation systems and granularity levels involved.

A IV: Provenance and Collaboration A recent review showed that it was not possible to reproduce the findings from almost 90% of over 50 cancer genomics studies [BE12]. This highlights the need for all stages of the analysis to be reproducible, interpretable, and communicable, including the visual analysis. Integrated support for provenance tracking, sharing of results, communication, and collaboration are essential.

A V: Integrated Data Analysis The integration of algorithms, statistics, and machine learning approaches like clustering or dimensionality reduction are crucial for most applications of visual analysis platforms to biomedical data. The back and forth between analysts and algorithms should be as tight and swift as possible. For instance, when a data query cannot provide immediate feedback due to the complexity of the query or the size of the data, the system should report intermediate results which

the analyst can use to judge the correctness and suitability of the parametrization and adjust them if necessary [MPG⁺14]. Data mining algorithms can also be used for guiding analysts to interesting patterns proactively [SLG⁺14].

A VI: Adaptability The last key aspect deals with the adaptability to changing environments. A visualization framework needs to be flexible enough to allow for, e.g., the addition of new data types, storage backends, visualization techniques, or processing algorithms. The platform should also support the creation of customized setups that are tailored to a specific application use case.

8.2. Related Platforms

BioJS [GGs⁺13]¹ is a library for representing biological data. Its core is a small event-driven architecture that can be extended via plugins that are collected in a public registry. Interfaces are not defined by the library but described within a plugin’s documentation only. This allows easy setup and creation of plugins for a range of different data types (A VI and A I). However, developers aiming at using multiple plugins in a setup with multiple coordinated views have to handle the synchronization and data mapping between individual plugins manually—hampering A II and A III. Moreover, the library focuses on the visualization of data only, not how it is accessed or processed (A V). Dealing with large datasets in web-based frameworks is particularly challenging, since transferring the whole dataset to the client is not an option.

Caleydo [LSKS10]² is a standalone visualization framework for biological data and the predecessor of Phovea. Caleydo is implemented in Java as an Eclipse RCP ³ application and uses OpenGL/JOGL for rendering. Caleydo supports A II and A III, however, it lacks support for large datasets (A I), since it is a client-only application in which all datasets are loaded into main memory. Moreover, it has only rudimentary support for provenance (A IV) via a simple undo mechanism and the integrated data processing (A V) is limited to a fixed set of hard coded algorithms, such as clustering algorithms.

¹<http://biojs.net>

²<http://github.com/Caleydo>

³http://wiki.eclipse.org/Rich_Client_Platform

8.3. Ecosystem

The Phovea platform has a client-server architecture and can be divided into a client and server framework and two support components for developing new Visual Analytics (VA) applications. A VA application in this context is a web application that is using a subset of client and server plugins. While individual components can be used in isolation, they work best in combination. Client and server are coupled loosely via REST and WebSocket interfaces. Figure 8.1 illustrates the four components and their relation to each other. In combination all key-aspects introduced in Section 8.1 are addressed.

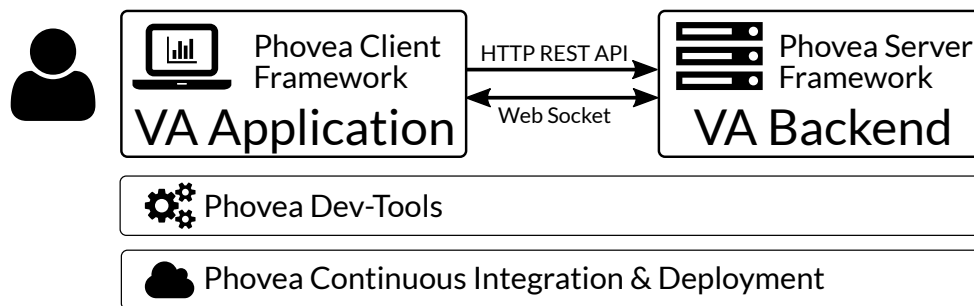


Figure 8.1.: The ecosystem of Phovea provides four components: client framework, server framework, dev-tools, and deployment. In combination they can be used to develop new Visual Analytics (VA) applications with server backends.

8.3.1. Phovea Client Framework

The Phovea client framework is an extensible JavaScript library written in Typescript. It consists of multiple plugins that can be linked together during build time. Moreover, plugins are lazy-loaded, reducing the initial start-up time. The following list contains a selection of standard plugins which are available at <http://github.com/phovea>:

phovea_core core library of the client framework, including commonly used data structures like vector, matrix, table, graph, and stratification (addressing A I), server-side communication, plugin management, and multiple coordinated view management (A III).

phovea_ui this library provides common styles and helper functions including dialogs based on Bootstrap⁴.

⁴<http://getbootstrap.com/>

phovea_d3 provides utility functions for using the popular D3 library [BOH11], including the creation of bands among visualizations for visually linking same items in different representations.

phovea_vis includes a set of standard visualization techniques including a box plot, histogram, scatterplot, pie chart, and heatmap implementation. The heatmap implementation supports the visualization of large datasets through generating the heatmap image on the server side.

phovea_importer is a general upload and import wizard for tabular data.

phovea_clue implements the CLUE model and corresponding provenance graph visualization techniques, addressing A IV. An introduction into CLUE is given in Chapter 7.

8.3.2. Phovea Server Framework

The Phovea server framework is a python based extensible web server library. It uses a similar extension mechanism as the client library and provides a REST and Websocket interface to clients. RESTful APIs are specified using the Swagger API framework ⁵. The following list contains a selection of server plugins available at <http://gitub.com/phovea>:

phovea_server is the core library of the server framework, which includes a unified data management and access, CSV data provider, RESTful APIs assembling, identifier (ID) management, and ID mapping (addressing A II).

phovea_processing_queue is an asynchronous processing engine based on Celery ⁶, addressing A V. In addition, it provides a small client library that simplifies the creation and management of tasks.

phovea_data_redis implements a data connector to a Redis database ⁷. Redis is a fast in-memory key value store that is used for ID management and mapping.

phovea_data_mongo implements a data connector to a Mongo database ⁸. User generated provenance graphs are stored within this database.

phovea_data_hdf is a data connector for reading files stored in the HDF5 file format ⁹.

⁵<http://swagger.io/>

⁶<http://celeryproject.org>

⁷<http://redis.io>

⁸<http://www.mongodb.com/>

⁹<http://hdfgroup.org/>

8.3.3. Phovea Dev-Tools

Besides the two core libraries, the Phovea platform consists of a set of development tools (dev-tools). These tools simplify the creation and management of Phovea plugins and applications based on Phovea.

Yeoman ¹⁰ is a scaffolding tool for creating new web applications. Phovea comes with an own generator available at <http://github.com/phovea/generator-phovea> including a set of useful commands, ranging from the creation of new plugins/applications to management of workspaces. A workspace is a set of locally checked out plugins that are linked together. In addition, for each workspace a PyCharm ¹¹ IDE project is created.

In Phovea client plugins and applications are developed using a set of tools and languages: TypeScript and SASS ¹² are programming languages in which the plugin is written. We use *Webpack* ¹³ to bundle these files together and *Karma Test runner* ¹⁴ to execute Unit-Tests. In addition, we assure code quality through *tslint* ¹⁵. Finally, API documentation is generated using *typedoc* ¹⁶. Server plugins are developed in Python and tested using *pytest* ¹⁷. Code quality is ensured by *flake8* ¹⁸. API documentation is generated using *Sphinx* ¹⁹.

8.3.4. Phovea Continuous Integration and Deployment

Besides the presented tools to support developers during the development of new applications and plugins, Phovea includes a set of tools dealing with continuous integration and deployment of created applications. We employ Travis ²⁰ for continuously executing units tests and checking code quality upon a push of a plugin or application.

Deploying an application includes defining a product definition. A product definition specifies which client and server plugins should be included in the deployed version of an

¹⁰<http://yeoman.io>

¹¹<http://www.jetbrains.com/pycharm/>

¹²<http://sass-lang.com>

¹³<http://webpack.github.io/>

¹⁴<http://karma-runner.github.io/>

¹⁵<http://palantir.github.io/tslint/>

¹⁶<http://typedoc.io/>

¹⁷<http://pytest.org>

¹⁸<http://flake8.pycqa.org/en/latest/>

¹⁹<http://http://www.sphinx-doc.org>

²⁰<http://travis-ci.org/>

application. This can defer from the local version, since additional data provider can be included, for example. Building a product definition result in a set of Docker ²¹ images. These images can be deployed in any docker container service including Amazon AWS or Google Cloud. Applications crated within the Caleydo ecosystem are hosted and accessible at <http://caleydoapp.org>.

8.4. Additional Libraries

8.4.1. LineUp.js

The LineUp visualization technique introduced in Chapter 4 was originally implemented in Caleydo. However, it has been ported to a standalone JavaScript library, available at <http://github.com/Caleydo/lineupjs>. It is actively being developed and recent extensions include supporting advanced column types, like box plots and sparklines.

8.4.2. CLUE Library

The CLUE library is a plugin of Phovea, available at http://github.com/phovea/phovea_clue. It is a prototypical implementation of the CLUE model introduced in Chapter 7. Individual provenance graph visualizations are implemented in D3 [BOH11]. We use the headless scriptable web browser PhantomJS ²² to generate screenshots on the server by replaying actions of the provenance graph. This is a generic approach for replaying the Phovea application enhanced with CLUE without the need for any customizations.

In order to use the library in a visual exploration application, is must use the *command design pattern* [GHJV95] for all recordable actions. These actions will then be captured in the provenance graph. In Gapminder, one of the usage scenarios described in Section 7.4.1, recordable actions include choosing attributes for individual axes, switching scales of axes, and selecting years and countries. More sophisticated applications, such as StratomeX, support a larger set of actions.

²¹<http://docker.com/>

²²<http://phantomjs.org>

8.5. Discussion

Using Phovea from the beginning when creating a new Visual Analysis application is the best case scenario. By providing useful tools creating a new application becomes easy. However, in real world scenarios platforms like the Refinery Platform ²³ are developed over the multiple years and may want to use Phovea components. Similarly, existing client applications may want to integrate the proposed CLUE model and face similar challenges when integrating it.

Integration in existing platforms Phovea is composed of multiple separate components that work best in combination. However, individual components like the client or server framework can be used in isolation as described in Section 8.1. Both frameworks are published on two package management systems, namely NPM ²⁴ and PyPi ²⁵, for the client and server, respectively. The main challenge to integrate Phovea components with other platforms are the interfaces that are used and provided by the frameworks. On the client side TypeScript interfaces are used. The communication between the client and server framework is described using the Swagger Open API standard. The web service provided by the server framework is based on the Web Service Gateway Interface (WSGI) ²⁶ specification. In addition, since Phovea provides a plugin mechanism, existing platforms can inject their platform specific adapters easily.

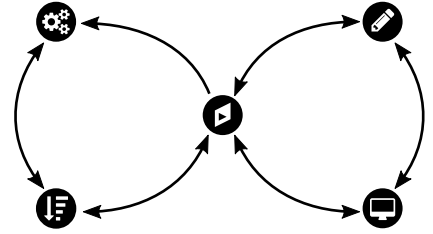
Integration of CLUE into existing code base Another common integration task it to extend an existing web application with CLUE functionality. As described in Section 8.4.2 a requirement for tracking, authoring, and presenting a visual analysis is using the command design pattern. Through separating the triggering and execution of an action, it is possible to store the trigger and related parameters into a provenance graph that can be replayed in a later stage. Moreover, when executing an action, the action has to provide the configuration for a trigger that wants to invert this action again. This ensures that analysts can flexible navigate within the provenance graph through reverting and applying different action triggers.

²³<http://www.refinery-platform.org/>

²⁴<http://npmjs.org>

²⁵<http://pypi.python.org/pypi>

²⁶<http://wsgi.org>



9 | Conclusion

Contents

9.1. Discussion and Future Work	114
9.2. Conclusion	117

9.1. Discussion and Future Work

This thesis has described a wide spectrum of methods and techniques for visually guiding users in selection, exploration, and presentation tasks. This results in a variety of possible future work directions.

Ranking Optimizations LineUp, as introduced in Chapter 4, provides an efficient and effective visualization for the exploration of multi-attribute rankings. Analysts can interactively manipulate how different attributes contribute to the final score in order to explore the ranking space, i.e., the space of all possible rankings with the given set of attributes. The recently published *WeightLifter* paper [PSTW⁺17] introduced a technique for exploring this ranking space. In the future, we plan to extend LineUp by providing means for optimizing rankings, for instance, by calculating and communicating the extent to which one or multiple attributes need to be changed to achieve a given rank (see R IX of the LineUp requirement analysis). This can be seen as an optimization problem within the ranking space.

Proactive Guidance Chapter 5 introduced a guidance process for selecting a data subset from a data collection. The approach is based on a query wizard in which the analyst formulates a query that is used to configure an algorithm that computes a score for each data subset. The resulting scores are then used to rank and prioritize the data subsets.

However, in the current approach the analyst must actively trigger and request guidance. A possible future research direction is to proactively guide analysts in their exploration. In the scenario described in Chapter 5 based on StratomeX, guidance could suggest potentially interesting data subset combinations that have a large overlap. The challenges include finding the balance between automated discovery of patterns and manual exploration in which the user can find patterns in an exploratory way.

Advanced Guidance in Domino Domino is a flexible generic visual exploration technique for multi-dimensional data that allows users to quickly reassemble existing visualization techniques. In our implementation, placeholders support analysts in creating combined blocks by indicating possible positions for placing a subset in the current Domino setup. In the future, we plan to enrich the placeholder interaction concept with further guidance capabilities that suggest potentially interesting relationships. Depending on the task at hand, possible drop positions could be ranked according to similarity measures, such as the correlation between groups of two partitioned blocks or advanced statistical measures such as scagnostics for orthogonal and Pargnostics [DK10] for parallel item relationships of two numerical blocks. For example Wilkinson et al. [WAG06] used scagnostics such as density, shape, and outliers of 2D scatterplots to classify and guide users to potentially interesting pairwise combinations in high dimensional datasets.

Multi-level Exploration Domino is a general technique for exploring data subsets and their relations to each other. However, it is difficult to drill down to the individual item level. LineUp, in contrast, is a general technique for ranking item collections that works at the item level only. In the future, we want to combine both concepts into one, such that analysts can start with exploring the dataset at a higher abstraction and aggregation level using a similar concept as in Domino, but can also drill down on demand. This is in accordance with Shneiderman’s information seeking mantra: “Overview first, zoom and filter, then details-on-demand.” [Shn96].

Provenance Graph Analysis In the CLUE model presented in Chapter 7, all actions performed by analysts during the visual exploration are captured in a provenance graph. The captured actions can be used in a later stage as a basis for communicating the discovery by authoring a *Vistory*. In the future, we plan to analyze the collection of captured provenance graphs. This can be done for a single graph, (e.g., by detecting cycles and similar states) or for a collection of graphs (e.g., by detecting shared analysis patterns and action sequences). Both can be used to support analysts by pointing to possible next actions or by indicating loops in the analysis. Provenance graph analysis can not only

support an analyst in the visual exploration process, it also provides insights into how analysts explore datasets in general. A major challenge related to the meta-analysis of provenance graphs is the definition of a similarity measure for states. The objects a state consists of and their semantics heavily depend on the application in which CLUE is used. Further, the analyst might be interested in just a subset of what a state consists of. For example, the analyst might be interested in finding only similar states that have the same selected items as the current state. Another challenge is how to effectively and efficiently communicate the similarity of states. On the one hand, the visual solution needs to be scalable such that the analyst can get an overview of all similarities. On the other hand, the visualization needs to encode the similarities and dissimilarities between two or more states in detail.

Using Guidance to Capture Users' Motivations Aside from capturing the “raw” provenance information, such as the data subsets manipulated or interaction performed, user motivation is of interest to fully understand and reproduce an exploration. We propose that integrated guidance mechanisms can be used to capture a user’s rationale during an exploration. Users have to formally express their intentions in order to be given support by the guidance system, for instance, by interacting with a wizard. These expressions can be tracked and used to infer user intentions. Moreover, since the guidance system has only a limited set of options for interaction, users are forced to formally express their intention within this set of options to receive guidance. This results in a win-win situation for both users and the reproducible system. Users receive guidance that speeds up their process, and the system can collect more provenance information about user intentions.

In addition, by providing a prioritization for the suggestions produced by the guidance system, the choices and modifications for the ranking can provide additional insights in the decision process of the analyst. This is especially useful when multiple options are examined. In a later stage, this information can be applied to understand why a specific option was chosen.

Gamification Performing an effective visual analysis is challenging due to the complexity of the domain problem. For example, in biomedicine effectively exploring data collections and generating or verifying new hypotheses require extensive experience and time-consuming studies. While domain knowledge will always remain a requirement, the system can better support the analyst in such exploration tasks. Better guidance that not only helps during but also before the exploration by introducing the visual analytics application and possible tasks enables novices to perform analyses. Gamification, which integrates aspects of games and game theory in non-game contexts, could potentially play a role in

improving guidance systems. This includes the possibility to share and comment on the one's work and on work done by others. A good example of using gamification aspects in the biomedical field is the protein folding game ¹. In this game, anyone can try to find a good protein fold by playing a game. Once a player has found a structure that fulfills particular minimal requirements, a scientist with domain knowledge takes a look at it. A possible direction for future work is to create and apply a similar idea to other visual analytics applications and domains. Challenges include how to introduce the task, problem domain, and tools such that novices can use them.

Open Visual Analysis Platform Chapter 7 introduces *Vistories*. We plan to launch a platform for sharing, viewing, and exploring Vistories along with the provenance graphs, data, and applications. Our vision is that an increasing number of visual exploration systems will capture the provenance graphs of their analysis sessions and that these provenance and Vistory packages could be submitted along with papers as supplementary material. In addition, authors could provide a link below a static screenshot figure in their paper that points to a Vistory reproducing the figure. This has the potential to simplify the job of reviewers, ensure reproducibility of the findings, improve communication of the findings, and ultimately speed up scientific progress.

9.2. Conclusion

In the course of this work several improvements and contributions for guiding and supporting users in their visual exploration of heterogeneous data were designed, implemented, and published. However, the results are not the end of the journey, but just a milestone on the path to the greater goal of better guiding analysts in order to discover more valuable findings in a shorter period of time. The CLUE model and the Phovea visual analysis platform both form a promising basis for further research directions.

FIN

¹<http://fold.it>

Bibliography

- [AG16] E. Alexander and M. Gleicher. Task-Driven Comparison of Topic Models. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):320–329, January 2016.
- [AMA⁺14] Bilal Alsallakh, Luana Micallef, Wolfgang Aigner, Helwig Hauser, Silvia Miksch, and Peter Rodgers. Visualizing Sets and Set-typed Data: State-of-the-Art and Future Challenges. In Rita Borgo, Ross Maciejewski, and Ivan Viola, editors, *Eurographics conference on Visualization (EuroVis)– State of The Art Reports*, pages 1–21. Eurographics, June 2014.
- [AWD12] A. Anand, L. Wilkinson, and T. N. Dang. Visual pattern discovery using random projections. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST '12)*, pages 43–52, October 2012.
- [BBH⁺17] M. Behrisch, B. Bach, M. Hund, M. Delz, L. Von Rüden, J. D. Fekete, and T. Schreck. Magnostics: Image-Based Search of Interesting Matrix Views for Guided Network Exploration. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):31–40, January 2017.
- [BCS⁺05] Louis Bavoil, Steven P. Callahan, Carlos Scheidegger, Huy T. Vo, Patricia Crossno, Claudio T. Silva, and Juliana Freire. VisTrails: Enabling Interactive Multiple-View Visualizations. In *Proceedings of the IEEE Conference on Visualization (VIS '05)*, pages 135–142. IEEE, 2005.
- [BDS⁺13] Michael Behrisch, James Davey, Svenja Simon, Tobias Schreck, Daniel Keim, and Jörn Kohlhammer. Visual Comparison of Orderings and Rankings. In *Proceedings of the EuroVis Workshop on Visual Analytics (EuroVA '13)*, 2013.
- [BE12] C. Glenn Begley and Lee M. Ellis. Drug development: Raise standards for preclinical cancer research. *Nature*, 483(7391):531–533, 2012.

- [Ber10] Jacques Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. ESRI Press, Redlands, CA, USA, 2010. First published in French in 1967.
- [BOH11] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3: Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.
- [Bri39] Willard Cope Brinton. *Graphic presentation*. Brinton associates, 1939.
- [Bro13a] Broad Institute TCGA Genome Data Analysis Center. Clustering of Methylation: consensus NMF. 2013.
- [Bro13b] Broad Institute TCGA Genome Data Analysis Center. Clustering of miRseq mature expression: consensus hierarchical. 2013.
- [Bro13c] Broad Institute TCGA Genome Data Analysis Center. Clustering of miRseq mature expression: consensus NMF. 2013.
- [Bro13d] Broad Institute TCGA Genome Data Analysis Center. Clustering of mRNAseq gene expression: consensus hierarchical. 2013.
- [Bro13e] Broad Institute TCGA Genome Data Analysis Center. Clustering of mRNAseq gene expression: consensus NMF. 2013.
- [Bro13f] Broad Institute TCGA Genome Data Analysis Center. Clustering of RPPA data: consensus hierarchical. 2013.
- [Bro13g] Broad Institute TCGA Genome Data Analysis Center. Clustering of RPPA data: consensus NMF. 2013.
- [BTel3] Cameron W Brennan, TCGA Research Network, and et al. The somatic genomic landscape of glioblastoma. *Cell*, 155(2):462–477, 2013.
- [BW08] L. Byron and M. Wattenberg. Stacked Graphs - Geometry & Aesthetics. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '08)*, 14(6):1245–1252, 2008.
- [CBS⁺15] Michael Correll, Adam L. Bailey, Alper Sarikaya, David H. O'Connor, and Michael Gleicher. LayerCake: a tool for the visual comparison of viral deep sequencing data. *Bioinformatics*, 31(21):3522–3528, November 2015.
- [CC00] Yizong Cheng and George M Church. Biclustering of Expression Data. In *Proceedings of the Conference on Intelligent Systems for Molecular Biology*

(ISMB '00), pages 93–103, Palo Alto, CA, USA, 2000. AAAI Press.

- [CC07] Christopher Collins and Sheelagh Carpendale. VisLink: Revealing Relationships Amongst Visualizations. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '07)*, 13(6):1192–1199, 2007.
- [CGM⁺17] Davide Ceneda, Theresia Gschwandtner, Thorsten May, Silvia Miksch, Hans-Jörg Schulz, Marc Streit, and Christian Tominski. Characterizing Guidance in Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics (VAST'16)*, 23(1):111–120, 2017.
- [CLN86] Daniel B Carr, Richard J Littlefield, and Wesley L Nicholson. Scatterplot matrix techniques for large N. In *Proceedings of the Symposium on the Interface of Computer Sciences and Statistics*, pages 297–306. Elsevier North-Holland, 1986.
- [CM84] William S. Cleveland and Robert McGill. Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. *Journal of the American Statistical Association*, 79(387):531–554, 1984.
- [CSC⁺12] Christina Curtis, Sohrab P. Shah, Suet-Feung Chin, Gulisa Turashvili, Oscar M. Rueda, Mark J. Dunning, Doug Speed, Andy G. Lynch, Shamith Samarajiwa, Yinyin Yuan, Stefan Gräf, Gavin Ha, Gholamreza Haffari, Ali Bashashati, Roslin Russell, Steven McKinney, Metabrix Group, Anita Langerød, Andrew Green, Elena Provenzano, Gordon Wishart, Sarah Pinder, Peter Watson, Florian Markowetz, Leigh Murphy, Ian Ellis, Arnie Purushotham, Anne-Lise Børresen-Dale, James D. Brenton, Simon Tavaré, Carlos Caldas, and Samuel Aparicio. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346–352, June 2012.
- [CvW11] Jerry H.T Claessen and Jarke J. van Wijk. Flexible Linked Axes for Multivariate Data Visualization. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '11)*, 17(12):2310–2316, 2011.
- [DHRL⁺12] Cody Dunne, Nathalie Henry Riche, Bongshin Lee, Ronald Metoyer, and George Robertson. GraphTrail: Analyzing Large Multivariate, Heterogeneous Networks While Supporting Exploration History. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, pages 1663–1672. ACM, 2012.
- [DK10] Aritra Dasgupta and Robert Kosara. Pargnostics: Screen-Space Metrics

- for Parallel Coordinates. *IEEE Transactions on Visualization & Computer Graphics*, 16(6):1017–1026, 2010.
- [ESBB98] Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences USA*, 95(25):14863–14868, 1998.
- [Few12] Stephen Few. *Show Me the Numbers: Designing Tables and Graphs to Enlighten*. Analytics Press, 2nd edition, 2012.
- [Fig14a] A. Figueiras. How to Tell Stories Using Visualization. In *Proceedings of the Conference on Information Visualisation (IV '14)*, pages 18–18, 2014.
- [Fig14b] A. Figueiras. Narrative Visualization: A Case Study of How to Incorporate Narrative Elements in Existing Visualizations. In *Proceedings of the Conference on Information Visualisation (IV '14)*, pages 46–52, 2014.
- [FSJ13] Sara Johansson Fernstad, Jane Shaw, and Jimmy Johansson. Quality-based guidance for exploratory dimensionality reduction. *Information Visualization*, 12(1):44–64, January 2013.
- [Fur86] George W. Furnas. Generalized fisheye views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '86)*, pages 16–23. ACM, 1986.
- [GAW⁺11] Michael Gleicher, Danielle Albers, Rick Walker, Ilir Jusufi, Charles D. Hansen, and Jonathan C. Roberts. Visual Comparison for Information Visualization. *Information Visualization*, 10(4):289–309, October 2011.
- [GGL⁺14] Samuel Gratzl, Nils Gehlenborg, Alexander Lex, Hanspeter Pfister, and Marc Streit. Domino: Extracting, Comparing, and Manipulating Subsets across Multiple Tabular Datasets. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '14)*, 20(12):2023–2032, 2014.
- [GGL⁺15] Samuel Gratzl, Nils Gehlenborg, Alexander Lex, Hendrik Strobelt, Christian Partl, and Marc Streit. Caleydo Web: An Integrated Visual Analysis Platform for Biomedical Data. In *Poster Compendium of the IEEE Conference on Information Visualization (InfoVis '15)*, Chicago, IL, USA, 2015. IEEE.
- [GGS⁺13] John Gómez, Leyla J. García, Gustavo A. Salazar, Jose Villaveces, Swanand Gore, Alexander García, Maria J. Martín, Guillaume Launay, Rafael Alcántara, Noemi del Toro, Marine Dumousseau, Sandra Orchard, Sameer Ve-

- lankar, Henning Hermjakob, Chenggong Zong, Peipei Ping, Manuel Corpas, and Rafael C. Jiménez. BioJS: an open source JavaScript framework for biological data visualization. *Bioinformatics*, 29(8):1103–1104, April 2013.
- [GHJV95] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman, 1995.
- [Gil13] Michael Gillhofer. Bi-Cluster Visualization. *Bachelor Thesis*, 2013.
- [GLG⁺13] Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Hanspeter Pfister, and Marc Streit. LineUp: Visual Analysis of Multi-Attribute Rankings. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 19(12):2277–2286, 2013.
- [GLG⁺16] Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Nicola Cosgrove, and Marc Streit. From Visual Exploration to Storytelling and Back Again. *Computer Graphics Forum*, 35(3):491–500, June 2016.
- [GPS⁺14] Nils Gehlenborg, Richard Park, Ilya Sytchev, Psalm Haseley, Stefan Luger, Anton Xue, Marc Streit, Shannan Ho Sui, Winston Hide, and Peter J. Park. Refinery Platform: A Foundation for Integrative Data Visualization Tools. In *Poster Compendium of the Symposium on Biological Data Visualization (BioVis '14)*, 2014.
- [HA85] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, December 1985.
- [Har06] Sandra G. Hart. NASA-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(9):904–908, 2006.
- [HB03] Mark Harrower and Cynthia A. Brewer. ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. *The Cartographic Journal*, 40(1):27–37, 2003.
- [HBS⁺16] Michael Hund, Dominic Böhm, Werner Sturm, Michael Sedlmair, Tobias Schreck, Torsten Ullrich, Daniel A. Keim, Ljiljana Majnarić, and Andreas Holzinger. Visual analytics for concept exploration in subspaces of patient groups. *Brain Informatics*, 3(4):233–247, December 2016.
- [Hea96] Christopher G. Healey. Choosing effective colours for data visualization. In

Proceedings of the IEEE Conference on Visualization (Vis '96), pages 263–270. IEEE, 1996.

- [HK81] J. A. Hartigan and B. Kleiner. Mosaics for contingency tables. *Proceedings of the Symposium on the Interface*, pages 268–273, 1981.
- [HLS⁺12] Clemens Holzhüter, Alexander Lex, Dieter Schmalstieg, Hans-Jörg Schulz, Heidrun Schumann, and Marc Streit. Visualizing Uncertainty in Biological Expression Data. In *Proceedings of the SPIE Conference on Visualization and Data Analysis (VDA '12)*, volume 8294, page 82940O. IS&T/SPIE, 2012.
- [HMSA08] Jeffrey Heer, Jock Mackinlay, Chris Stolte, and Maneesh Agrawala. Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '08)*, 14(6):1189–1196, 2008.
- [HR07] J. Heer and G. G Robertson. Animated Transitions in Statistical Data Graphics. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '07)*, 13(6):1240–1247, 2007.
- [HSC⁺13] Matan Hofree, John P Shen, Hannah Carter, Andrew Gross, and Trey Ideker. Network-based stratification of tumor mutations. *Nature Methods*, 10:1108–1115, September 2013.
- [Hun09] Lawrence Hunter. *The processes of life: an introduction to molecular biology*. MIT Press, March 2009.
- [HV81] Harold V. Henderson and Paul F. Velleman. Building Multiple Regression Models Interactively. *Biometrics*, 37(2):391–411, June 1981.
- [ID90] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the IEEE Conference on Visualization (Vis '90)*, pages 361–378. IEEE, 1990.
- [IML13] J.-F. Im, M.J. McGuffin, and R. Leung. GPLOM: The Generalized Plot Matrix for Visualizing Multidimensional Multivariate Data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2606–2614, 2013.
- [Ins85] Alfred Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(4):69–91, 1985.
- [JE12] Waqas Javed and Niklas Elmqvist. Exploring the design space of composite visualization. In *Proceedings of the IEEE Pacific Visualization Symposium*

(*PacificVis '12*), pages 1–8. IEEE, March 2012.

- [JE13] W. Javed and N. Elmqvist. ExPlates: Spatializing Interactive Analysis to Scaffold Visual Exploration. *Computer Graphics Forum (EuroVis '13)*, 32(3pt4):441–450, 2013.
- [JTS08] Mathias John, Christian Tominski, and Heidrun Schumann. Visual and analytical extensions for the table lens. In *Proceedings of the SPIE Conference on Visualization and Data Analysis (VDA '08)*, 2008.
- [KBH06] Robert Kosara, Fabian Bendix, and Helwig Hauser. Parallel Sets: Interactive Exploration and Visual Analysis of Categorical Data. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):558–568, 2006.
- [KKEM10] Daniel A Keim, Joern Kohlhammer, Geoffrey Ellis, and Florian Mansmann, editors. *Mastering the Information Age - Solving Problems with Visual Analytics*. Eurographics, Goslar, Germany, 2010.
- [KLC08] Paul Kidwell, Guy Lebanon, and William S. Cleveland. Visualizing incomplete and partially ranked data. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1356–1363, 2008.
- [KM13] Robert Kosara and Jock Mackinlay. Storytelling: The next step for visualization. *Computer*, (5):44–50, 2013.
- [KNS04] M. Kreuseler, T. Nocke, and H. Schumann. A History Mechanism for Visual Data Mining. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '04)*, pages 49–56. IEEE, 2004.
- [KRL97] J. Kolojejchick, S. F Roth, and P. Lucas. Information appliances and tools in Visage. *IEEE Computer Graphics and Applications*, 17(4):32–41, 1997.
- [KSJ⁺14] Bum Chul Kwon, Florian Stoffel, Dominik Jäckle, Bongshin Lee, and Daniel Keim. VisJockey: Enriching Data Stories through Orchestrated Interactive Visualization. In *Poster Compendium of the Computation + Journalism Symposium*, 2014.
- [KV05] Seon-Young Kim and David J. Volsky. PAGE: Parametric Analysis of Gene Set Enrichment. *BMC Bioinformatics*, 6(1):144, 2005.
- [LHV12] Endre M. Lidal, Helwig Hauser, and Ivan Viola. Geological storytelling: graphically exploring and communicating geological sketches. In *Proceedings of the Symposium on Sketch-Based Interfaces and Modeling (SBIM '12)*,

pages 11–20. Eurographics Association, 2012.

- [LPK⁺13] Alexander Lex, Christian Partl, Denis Kalkofen, Marc Streit, Samuel Gratzl, Anne Mai Wassermann, Dieter Schmalstieg, and Hanspeter Pfister. Entourage: Visualizing Relationships between Biological Pathways using Contextual Subsets. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 19(12):2536–2545, 2013.
- [LRIC15] Bongshin Lee, N.H. Riche, P. Isenberg, and S. Carpendale. More Than Telling a Story: Transforming Data into Visually Shared Stories. *IEEE Computer Graphics and Applications*, 35(5):84–90, 2015.
- [LSKS10] Alexander Lex, Marc Streit, Ernst Kruijff, and Dieter Schmalstieg. Caleydo: Design and Evaluation of a Visual Analysis Framework for Gene Expression Data in its Biological Context. In *Proceedings of the IEEE Symposium on Pacific Visualization (PacificVis '10)*, pages 57–64. IEEE, 2010.
- [LSP⁺10] Alexander Lex, Marc Streit, Christian Partl, Karl Kashofer, and Dieter Schmalstieg. Comparative Analysis of Multidimensional, Quantitative Data. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '10)*, 16(6):1027–1035, 2010.
- [LSS⁺11] Alexander Lex, Hans-Jörg Schulz, Marc Streit, Christian Partl, and Dieter Schmalstieg. VisBricks: Multiform Visualization of Large, Inhomogeneous Data. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '11)*, 17(12):2291–2300, 2011.
- [LSS⁺12] Alexander Lex, Marc Streit, Hans-Jörg Schulz, Christian Partl, Dieter Schmalstieg, Peter J. Park, and Nils Gehlenborg. StratomeX: Visual Analysis of Large-Scale Heterogeneous Genomics Data for Cancer Subtype Characterization. *Computer Graphics Forum (EuroVis '12)*, 31(3):1175–1184, 2012.
- [Mac86] Jock Mackinlay. Automating the Design of Graphical Presentations of Relational Information. *ACM Transactions on Graphics*, 5(2):110–141, 1986.
- [MGT⁺03] Tamara Munzner, François Guimbretière, Serdar Tasiran, Li Zhang, and Yunhong 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)*, pages 453–462. ACM, 2003.
- [MPG⁺14] Thomas Mühlbacher, Harald Piringer, Samuel Gratzl, Michael Sedlmair, and

- Marc Streit. Opening the Black Box: Strategies for Increased User Involvement in Existing Algorithm Implementations. *IEEE Transactions on Visualization and Computer Graphics (VAST '14)*, 20(12):1643–1652, 2014.
- [Mun09] Tamara Munzner. A Nested Process Model for Visualization Design and Validation. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '09)*, 15(6):921–928, 2009.
- [Mun14] Tamara Munzner. *Visualization Analysis and Design*. CRC Press, Taylor & Francis Group, Boca Raton, 2014.
- [Net08] The Cancer Genome Atlas Research Network. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455(7216):1061–1068, 2008.
- [Nut13] Nutrient Data Laboratory. USDA National Nutrient Database for Standard Reference, Release 25, 2013.
- [Par11a] Charlie Park. Edward Tufte’s “Slopegraphs”, 2011.
- [Par11b] Charlie Park. A Slopegraph Update, 2011.
- [PDL⁺11] Suporn Pongnumkul, Mira Dontcheva, Wilmot Li, Jue Wang, Lubomir Bourdev, Shai Avidan, and Michael F. Cohen. Pause-and-play: Automatically Linking Screencast Video Tutorials with Applications. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*, pages 135–144. ACM, 2011.
- [PGS⁺16] Christian Partl, Samuel Gratzl, Marc Streit, Anne Mai Wassermann, Hanspeter Pfister, Dieter Schmalstieg, and Alexander Lex. Pathfinder: Visual Analysis of Paths in Graphs. *Computer Graphics Forum (EuroVis '16)*, 35(3):71–80, June 2016.
- [PGU12] A. Pilhofer, A. Gribov, and A. Unwin. Comparing Clusterings Using Bertin’s Idea. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2506–2515, December 2012.
- [PSTW⁺17] Stephan Pajer, Marc Streit, Thomas Torsney-Weir, Florian Spechtenhauser, Torsten Möller, and Harald Piringer. WeightLifter: Visual Weight Space Exploration for Multi-Criteria Decision Making. *IEEE Transactions on Visualization and Computer Graphics (InfoVis'16)*, 23(1):611–620, 2017.
- [PWR04] Wei Peng, Matthew O. Ward, and Elke A. Rundensteiner. Clutter Reduc-

tion in Multi-Dimensional Data Visualization Using Dimension Reordering. In *INFOVIS '04: Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04)*, pages 89–96, Washington, DC, USA, 2004. IEEE Computer Society.

- [Qua13] Quacquarelli Symonds. QS World University Ranking, 2013.
- [RC94] Ramana Rao and Stuart K. Card. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '94)*, pages 318–322. ACM, 1994.
- [RESC16] E.D. Ragan, A. Endert, J. Sanyal, and J. Chen. Characterizing Provenance in Visualization and Data Analysis: An Organizational Framework of Provenance Types and Purposes. *IEEE Transactions on Visualization and Computer Graphics (VAST '15)*, 22(1):31–40, 2016.
- [RH11] N. B. Robbins and R. M. Heiberger. Plotting Likert and Other Rating Scales. In *Proceedings of the 2011 Joint Statistical Meeting*, 2011.
- [RHF05] Patrick Riehmann, Manfred Hanfler, and Bernd Froehlich. Interactive Sankey diagrams. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '05)*, pages 233–240. IEEE, 2005.
- [RNP⁺10] Jason T Rich, J Gail Neely, Randal C Paniello, Courtney C J Voelker, Brian Nussenbaum, and Eric W Wang. A practical guide to understanding Kaplan-Meier curves. *Otolaryngology–Head and Neck Surgery*, 143(3):331–336, 2010.
- [Rob07] Jonathan 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)*, pages 61–71. IEEE, 2007.
- [SA06] Ben Shneiderman and Aleks Aris. Network Visualization by Semantic Substrates. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '06)*, 12(5):733–740, 2006.
- [Sch02] Judi Scheffer. Dealing with missing data. *Research letters in the information and mathematical sciences*, 3(1):153–160, 2002.
- [SCL⁺12] Conglei Shi, Weiwei Cui, Shixia Liu, Panpan Xu, Wei Chen, and Huamin Qu. RankExplorer: Visualization of Ranking Changes in Large Time Series Data.

- IEEE Transactions on Visualization and Computer Graphics*, 18(12):2669–2678, 2012.
- [SGAS16] Holger Stitz, Samuel Gratzl, Wolfgang Aigner, and Marc Streit. ThermalPlot: Visualizing Multi-Attribute Time-Series Data Using a Thermal Metaphor. *IEEE Transactions on Visualization and Computer Graphics*, 22(12):2594–2607, 2016.
- [SGG⁺14] Marc Streit, Samuel Gratzl, Michael Gillhofer, Andreas Mayr, Andreas Mitterecker, and Sepp Hochreiter. Furby: Fuzzy Force-Directed Bicluster Visualization. *BMC Bioinformatics*, 15(Suppl 6):S4, 2014.
- [SGKS15] Holger Stitz, Samuel Gratzl, Michael Krieger, and Marc Streit. CloudGazer: A Divide-and-Conquer Approach for Monitoring and Optimizing Cloud-Based Networks. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis '15)*, pages 175–182. IEEE, 2015.
- [SH08] Amit P. Sawant and Christopher G. Healey. Visualizing multidimensional query results using animation. In *Electronic Imaging 2008*, page 680904, 2008.
- [SH14] Arvind Satyanarayan and Jeffrey Heer. Authoring Narrative Visualizations with Ellipsis. *Computer Graphics Forum*, 33(3):361–370, 2014.
- [Shn96] Ben 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)*, pages 336–343, 1996.
- [SKKS08] Marc Streit, Michael Kalkusch, Karl Kashofer, and Dieter Schmalstieg. Navigation and Exploration of Interconnected Pathways. *Computer Graphics Forum (EuroVis '08)*, 27(3):951–958, 2008.
- [SLG⁺14] Marc Streit, Alexander Lex, Samuel Gratzl, Christian Partl, Dieter Schmalstieg, Hanspeter Pfister, Peter J. Park, and Nils Gehlenborg. Guided visual exploration of genomic stratifications in cancer. *Nature Methods*, 11(9):884–885, 2014.
- [SS05] Jinwook Seo and Ben Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, 4(2):96–113, 2005.
- [SSL⁺12] Marc Streit, Hans-Jörg Schulz, Alexander Lex, Dieter Schmalstieg, and Hei-

drun Schumann. Model-Driven Design for the Visual Analysis of Heterogeneous Data. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):998–1010, 2012.

- [SSTR93] Manojit Sarkar, Scott S. Snibbe, Oren J. Tversky, and Steven 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)*, pages 81–91. ACM, 1993.
- [STM⁺05] Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, Scott L. Pomeroy, Todd R. Golub, Eric S. Lander, and Jill P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43):15545–15550, October 2005.
- [SW08] Yedendra Babu Shrinivasan and Jarke J. van Wijk. Supporting the analytical reasoning process in information visualization. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*, pages 1237–1246. ACM, 2008.
- [SWL⁺11] Markus Steinberger, Manuela Waldner, Alexander Lex, Marc Streit, and Dieter Schmalstieg. Context-Preserving Visual Links. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '11)*, 17(12):2249–2258, 2011.
- [The13] The Cancer Genome Atlas Research Network. Comprehensive molecular characterization of clear cell renal cell carcinoma. *Nature*, 499(7456):43–49, July 2013.
- [Tim12] Times Higher Education. Times Higher Education 100 Under 50, 2012.
- [TMF⁺12] A. Tatu, F. Maaß, I. Färber, E. Bertini, T. Schreck, T. Seidl, and D. Keim. Subspace search and visualization to make sense of alternative clusterings in high-dimensional data. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 63–72, October 2012.
- [TMH⁺13] Tuan Zea Tan, Qing Hao Miow, Ruby Yun-Ju Huang, Meng Kang Wong, Jieru Ye, Jieying Amelia Lau, Meng Chu Wu, Luqman Hakim Bin Abdul Hadi, Richie Soong, Mahesh Choolani, Ben Davidson, Jahn M Nesland, Ling-Zhi Wang, Noriomi Matsumura, Masaki Mandai, Ikuo Konishi, Boon-Cher Goh, Jeffrey T Chang, Jean Paul Thiery, and Seiichi Mori. Functional genomics identifies five distinct molecular subtypes with clinical relevance and

pathways for growth control in epithelial ovarian cancer. *EMBO Molecular Medicine*, 5(7):983–998, July 2013.

- [Tuf83] Edward Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 2nd edition edition, 1983.
- [Tuf95] Edward Tufte. *Envisioning information*. Graphics Press, Cheshire Conn., 5th edition, 1995.
- [vdEvW13] Stef van den Elzen and Jarke J. van Wijk. Small Multiples, Large Singles: A New Approach for Visual Data Exploration. *Computer Graphics Forum (EuroVis '13)*, 32(3pt2):191–200, 2013.
- [Vea10] Roel G.W. Verhaak and et al. Integrated Genomic Analysis Identifies Clinically Relevant Subtypes of Glioblastoma Characterized by Abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer Cell*, 17(1):98–110, 2010.
- [VM12] Christophe Viau and Michael J. McGuffin. ConnectedCharts: Explicit Visualization of Relationships between Data Graphics. *Computer Graphics Forum (EuroVis '12)*, 31(3pt4):1285–1294, 2012.
- [WAG06] L. Wilkinson, A. Anand, and R. Grossman. High-Dimensional Visual Analytics: Interactive Exploration Guided by Pairwise Views of Point Distributions. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1363–1372, November 2006.
- [WGK10] Matthew Ward, Georges Grinstein, and Daniel A. Keim. *Interactive Data Visualization: Foundations, Techniques, and Application*. A.K. Peters, Natick, MA, USA, 2010.
- [WH07] Michael Wohlfart and Helwig Hauser. Story Telling for Presentation in Volume Visualization. In *Proceedings of the Symposium on Visualization (EuroVis '07)*, pages 91–98. Eurographics Association, 2007.
- [Wil05] Leland Wilkinson. *The grammar of graphics*. Springer, 2nd edition, 2005.
- [YGX⁺09] Xiaoru Yuan, Peihong Guo, He Xiao, Hong Zhou, and Huamin Qu. Scattering Points in Parallel Coordinates. *IEEE Transactions on Visualization and Computer Graphics (Infovis '09)*, 15(6):1001–1008, 2009.

A | Phovea Applications

A couple of applications have already been developed with Phovea. The following selection covers applications that are directly related to this thesis or in which the author has contributed. A full list of public applications can be found at <http://caleydoapp.org>.

A.1. Caleydo LineUp and LineUp.js

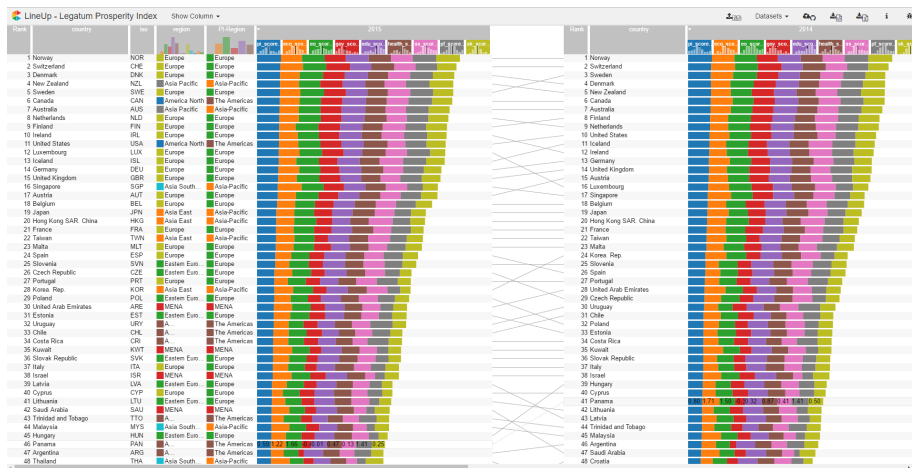


Figure A.1.: Screenshot of Caleydo LineUp. Caleydo LineUp is a Phovea based application using the underlying independent LineUp.js library for multi-attribute rankings. In this example a comparison over the years of the Legatum Prosperity Index¹ is shown.

Website	http://lineup.caleydo.org
Source Code	http://github.com/Caleydo/lineup
	http://github.com/Caleydo/lineupjs
Public Version	http://lineup.caleydoapp.org

Caleydo LineUp is a web application for exploring multi-attribute rankings shown in Figure A.1. It is using LineUp.js – a JavaScript implementation of the visualization technique introduced in Chapter 4. Users can explore a set of preloaded datasets and upload their own ones. The resulting visual analysis can be easily shared by downloading the result or by uploading the data to Github Gist ². The latter one produces a shareable URL.

A.2. Caleydo Gapminder with CLUE

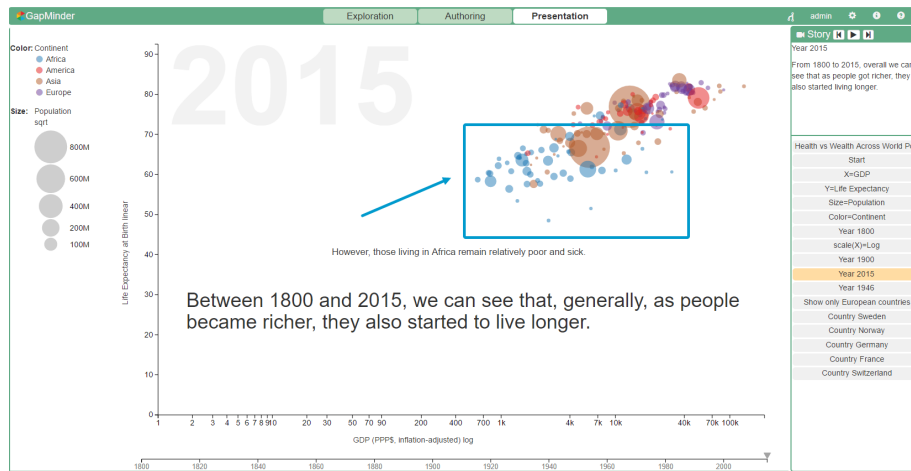


Figure A.2.: Screenshot of Caleydo Gapminder

Website <http://clue.caleydo.org>
Source Code <http://github.com/Caleydo/gapminder>
http://github.com/phovea/phovea_clue
Public Version <http://gapminder.caleydoapp.org>

Caleydo Gapminder is a visual analytics tool inspired by the original Gapminder ³. This application is used as running example in Chapter 7. In addition to the original Gapminder functionality, the presented CLUE model is integrated. A detailed usage scenario can be found in Section 7.4.1.

²<http://gist.github.com>

³<http://gapminder.org>

A.3. StratomeX.js with CLUE

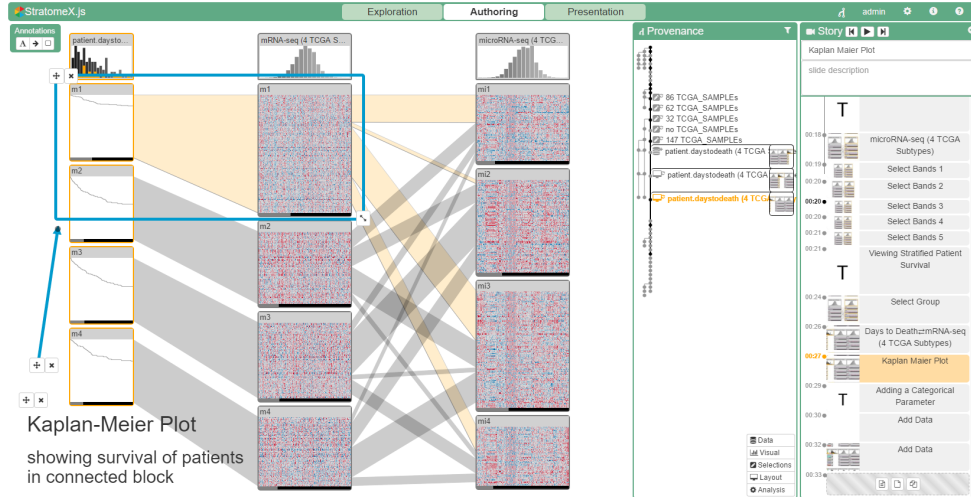


Figure A.3.: Screenshot of Caleydo StratomeX

Website <http://stratomex.caleydo.org>
Source Code http://github.com/Caleydo/stratomex_js
http://github.com/phovea/phovea_clue
Public Version <http://stratomex.caleydoapp.org>

StratomeX.js is an implementation of the *StratomeX* [LSS⁺12] visualization technique with extensions for the *Guided StratomeX* approach introduced in Chapter 5 and CLUE functionality. An introduction into the visualization technique can be found in Section 3.1.

A.4. Caleydo Pathfinder

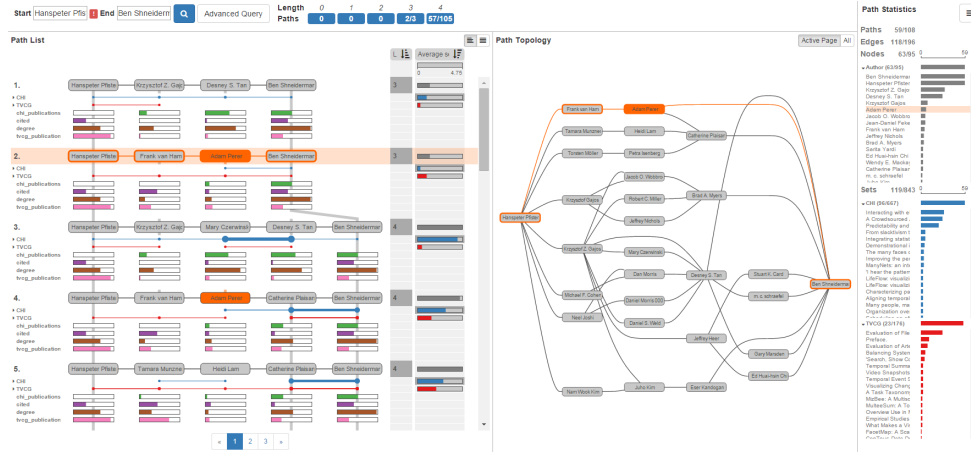


Figure A.4.: Screenshot of Caleydo Pathfinder [PGS⁺16]

Website <http://pathfinder.caleydo.org>
Source Code <http://github.com/Caleydo/pathfinder>
http://github.com/Caleydo/pathfinder_ccle
http://github.com/Caleydo/pathfinder_graph
Public Version <http://pathfinder.caleydoapp.org>

Pathfinder [PGS⁺16] is a visualization technique for path based tasks in large heterogeneous networks. It uses a Neo4j⁴ database for storing the graph. While this is not a core technique of this thesis, the author contributes to the implementation of the prototype.

⁴<http://neo4j.com/>

Curriculum Vitae

Personal Data

Name Dipl.-Ing. **Samuel Gratzl**, Bsc.
samuel.gratzl@gmx.at

Born February 5, 1986 in Linz, Austria

Nationality Austria



Research Interests

- Information Visualization
- Visualization on the Web
- Biological Data Visualization
- Visual Analytics and Knowledge Discovery
- Software Design

Professional

since 11/2016 Co-founder and CTO of datavisyn GmbH

11/2016 – 01/2017 Lecturer at Hagenberg University of Applied Sciences, Austria

11/2016 – 01/2017 Lecturer at Salzburg University of Applied Sciences, Austria

since 06/2016 Project Assistant at the Institute for Computer Graphics, JKU, Linz

02/2016 – 03/2016 Teaching Assistant for the Visualization Class at the Imperial College London, London, UK

01/2015 – 04/2015 Research Fellow at the School of Engineering & Applied Sciences, Harvard University, Boston, MA, USA

02/2015 – 05/2015 Teaching Assistant for the Visualization Class (CS171) at the School of Engineering & Applied Sciences, Harvard University, Boston, MA, USA

11/2012 – 06/2016 Predoctoral Associate at the Institute for Computer Graphics, JKU, Linz

03/2011 – 07/2011 Teaching Assistant at the Institute for Computer Graphics for Computer Graphics and Mixed Reality, JKU, Linz

03/2010 – 07/2010 Teaching Assistant at the Institute for Computer Graphics for Computer Graphics and Mixed Reality, JKU, Linz

07/2007 – 12/2012 MathConsult GmbH (Linz, Austria); Part Time; Software Engineer with focus on Web development and build management, Linz

09/2006 – 12/2012 Lernfamilie (Linz, Austria); Part Time; Webmaster, Linz

Education

since 10/2012 Doctoral program in Computer Science at Johannes Kepler University Linz.
Supervision: Ass.-Prof. Dipl.-Ing. Dr.techn. Marc Streit

10/2012 Master's degree (Dipl.-Ing.) from Johannes Kepler University Linz *with highest distinction*
Thesis: *MAPS - a mobile automatic puzzle solver*
Supervision: Univ.-Prof. Dr. Gerhard Widmer and a.Univ.-Prof. Dr. Josef Scharinger

04/2010 – 10/2012 Master's studies in Pervasive Computing at Johannes Kepler University Linz

04/2010 Bachelor's degree from Johannes Kepler University Linz *with highest distinction*
Thesis: *Conception and Realization of a Deployment Process for the UnRiskFactory project*
Supervision: Univ.-Prof. Dr. Dr. h.c. Hanspeter Mössenböck
in cooperation with MathConsult GmbH

10/2006 – 04/2010 Bachelor's studies in Computer Science at Johannes Kepler University Linz

Awards And Scholarships

2016 *Honorable Mention Best Paper Award*, EG/VGTC Conference on Visualization (EuroVis' 16)

2015 *Honorable Mention Best Poster Award*, IEEE Information Visualization (InfoVis' 15)

2015 *Human Technology Interface Award of the State of Styria*

2015 *Dissertation Scholarship of the State of Upper Austria*

2014	<i>Marshallplan Scholarship for Research at Harvard University</i>
2014	<i>Honorable Mention Best Paper Award</i> , IEEE Information Visualization (InfoVis'14)
2013	<i>Best Paper Award</i> , IEEE Information Visualization (InfoVis'13)
2007 – 2010, 2012	<i>Award for excellent performance as a student</i> (Leistungsstipendium) granted by the Faculty of Engineering and Natural Sciences, Johannes Kepler University Linz

Grants

2015 – 2018	<i>TourGuide - Navigation System for Capturing and Analyzing Complex Clinical Data</i> Funded by State of Upper Austria, Innovatives OÖ2020, Grant no. 851460, 400k EUR.
2015 – 2018	<i>VisOnFire - Visual Analysis of Large and Heterogeneous Scientific Workflows for Analytical Provenance.</i> Funded by Austrian Science Fund (FWF), Grant no. P27975-NBL, 348k EUR.
2015 – 2018	<i>Development of Multi-Dimensional Genomic Data Visualization and Data Mining Techniques.</i> Funded by Boehringer Ingelheim Regional Center Vienna, 210k EUR
2013 – 2016	<i>PIPES-VS-DAMS - Privacy Preserving Visual Dynamic Network Analysis for Advanced System Monitoring on Multiple Scales.</i> Funded by the Austrian Research Promotion Agency (FFG), IKT der Zukunft program, Grant no. 840232, 410k EUR.

Top International Collaboration Partners

- Prof. Hanspeter Pfister, Ph.D, School of Eng. & Applied Sciences, Harvard Univ., Boston, MA, USA
- Nils Gehlenborg, Ph.D, Broad Institute of MIT and Harvard, Boston, MA, USA
- Alexander Lex, Ph.D, University of Utah, Salt Lake City, UT, USA

Publications

Peer-reviewed Journal Publications

- [1] Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Nicola Cosgrove, and Marc Streit. From Visual Exploration to Storytelling and Back Again. *Computer Graphics Forum (EuroVis '16)*, 2016. to appear.
- [2] Christian Partl, Samuel Gratzl, Marc Streit, Anne Mai Wassermann, Hanspeter Pfister, Dieter Schmalstieg, and Alexander Lex. Pathfinder: Visual Analysis of Paths in Graphs. *Computer Graphics Forum (EuroVis '16)*, 2016. to appear.
- [3] Holger Stitz, Samuel Gratzl, Wolfgang Aigner, and Marc Streit. ThermalPlot: Visualizing Multi-Attribute Time-Series Data Using a Thermal Metaphor. *IEEE Transactions on Visualization and Computer Graphics*, 2016. Accepted with minor revision.
- [4] Marc Streit, Alexander Lex, Samuel Gratzl, Christian Partl, Dieter Schmalstieg, Hanspeter Pfister, Peter J. Park, and Nils Gehlenborg. Guided visual exploration of genomic stratifications in cancer. *Nature Methods*, 11(9):884–885, 2014. PMC4196637.
- [5] Samuel Gratzl, Nils Gehlenborg, Alexander Lex, Hanspeter Pfister, and Marc Streit. Domino: Extracting, comparing, and manipulating subsets across multiple tabular datasets. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '14)*, 20(12):2023–2032, 2014.
- [6] Thomas Muhlbacher, Harald Piringer, Samuel Gratzl, Michael Sedlmair, and Marc Streit. Opening the black box: Strategies for increased user involvement in existing algorithm implementations. *IEEE Transactions on Visualization and Computer Graphics (VAST '14)*, 20(12):1643–1652, 2014.
- [7] Marc Streit, Samuel Gratzl, Michael Gillhofer, Andreas Mayr, Andreas Mitterecker, and Sepp Hochreiter. Furby: Fuzzy force-directed bicluster visualization. *BMC Bioinformatics*, 15(Suppl 6):S4, 2014.
- [8] Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Hanspeter Pfister, and Marc Streit. LineUp: Visual analysis of multi-attribute rankings. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 19(12):2277–2286, 2013.
- [9] Alexander Lex, Christian Partl, Denis Kalkofen, Marc Streit, Samuel Gratzl, Anne Mai Wasserman, Dieter Schmalstieg, and Hanspeter Pfister. Entourage: Visualizing relationships between biological pathways using contextual subsets. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 19(12):2536–2545, 2013.

Peer-reviewed Conference and Workshop Publications

- [1] Holger Stitz, Samuel Gratzl, Michael Krieger, and Marc Streit. CloudGazer: A Divide-and-Conquer Approach for Monitoring and Optimizing Cloud-Based Networks. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis '15)*, pages 175–182. IEEE, 2015.

Posters

- [1] Samuel Gratzl, Nils Gehlenborg, Alexander Lex, Hendrik Strobelt, Christian Partl, and Marc Streit. Caleydo Web: An Integrated Visual Analysis Platform for Biomedical Data. In *Poster Compendium of the IEEE Conference on Information Visualization (InfoVis '15)*. IEEE, 2015.
- [2] Stefan Luger, Holger Stitz, Samuel Gratzl, Nils Gehlenborg, and Marc Streit. Interactive Visualization of Provenance Graphs for Reproducible Biomedical Research. In *Poster Compendium of the IEEE Conference on Information Visualization (InfoVis '15)*. IEEE, 2015.
- [3] Christian Partl, Samuel Gratzl, Marc Streit, Hanspeter Pfister, Dieter Schmalstieg, and Alexander Lex. Pathfinder: Visual Analysis of Paths in Heterogeneous Graphs. In *Poster Compendium of the IEEE Symposium on Visualization in Data Science (VDS '15)*. IEEE, 2015.
- [4] Holger Stitz, Samuel Gratzl, Wolfgang Aigner, and Marc Streit. ThermalPlot: Visualizing Multi-Attribute Time-Series Data Using a Thermal Metaphor. In *Poster Compendium of the IEEE Conference on Information Visualization (InfoVis '15)*. IEEE, 2015.
- [5] Holger Stitz, Samuel Gratzl, Stefan Luger, Nils Gehlenborg, and Marc Streit. Transparent layering for visualizing dynamic graphs using the flip book metaphor. In *Poster Compendium of the IEEE VIS Conference*. IEEE Computer Society Press, 2014.

Linz, March, 2017

Sworn declaration

I certify that this research thesis is the result of my own work, except as acknowledged, and has not been submitted, either in part or whole, for a degree at this or any other University.

The present thesis is identical to the document that has been submitted electronically.

Individual chapters of this cumulative thesis have been published as international conference articles and journal papers (be referred to [GLG⁺13, GGL⁺14, GGL⁺15, GLG⁺16, SLG⁺14] for further details).

Linz, March 2017



Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die vorliegende Dissertation ist mit dem elektronisch übermittelten Textdokument identisch.

Einzelne Kapitel dieser kumulativen Dissertation wurden als Arbeiten auf internationalen Konferenzen und in Journalen publiziert. Die entsprechenden Artikel sind unter [GLG⁺13, GGL⁺14, GGL⁺15, GLG⁺16, SLG⁺14] im Literaturverzeichnis gelistet.

Linz, März 2017

