FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Data-Driven User Guidance in Multi-Attribute Data Exploration

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Medizinische Informatik

eingereicht von

Klaus Eckelt, BSc
Matrikelnummer 1127705

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Univ.-Doz. Dipl.-Ing. Dr.techn. Eduard Gröller
Mitwirkung: Univ.-Prof. Dipl.-Ing. Dr.techn. Marc Streit

Wien, 18. August 2018

Klaus Eckelt	Eduard Gröller

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Data-Driven User Guidance in Multi-Attribute Data Exploration

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Medical Informatics

by

Klaus Eckelt, BSc
Registration Number 1127705

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Univ.-Doz. Dipl.-Ing. Dr.techn. Eduard Gröller
Assistance: Univ.-Prof. Dipl.-Ing. Dr.techn. Marc Streit

Vienna, 18th August, 2018

Klaus Eckelt	Eduard Gröller

Erklärung zur Verfassung der Arbeit

Klaus Eckelt, BSc
Rainerstraße 21/4, 4020 Linz

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 18. August 2018

Klaus Eckelt

Acknowledgements

I would like to thank everyone who supported me throughout the thesis in one way or another. Many thanks to Eduard Gröller and Marc Streit, whose cooperation allowed me to do this thesis. I also want to thank the members of the Institute of Computer Graphics of the Johannes Kepler University Linz for welcoming me in their team and supporting me throughout the thesis. Thanks to the project partners at the RISC Software GmbH for their collaboration and the medical experts from the Kepler University Hospital for their feedback. Finally I want to thank my family, friends and colleagues for their support, but also providing the necessary distraction.

Development and results reported here are in part based upon datasets generated by the The Cancer Genome Atlas (TCGA) Research Network. I want to thank the specimen donors and research groups involved in their generation.

The TourGuide project (FFG 851460) is funded by the State of Upper Austria via the Innovatives Oberösterreich 2020 program.

Kurzfassung

Das Finden von Beziehungen in großen Datensammlungen ist eine immer wiederkehrende Aufgabe, welche jedoch durch ständig wachsende Datenmengen und deren Heterogenität komplexer wird. Zur Visualisierung multivariater Daten gibt es mehrere Ansätze: (i) Projektionstechniken nehmen eine Dimensionsreduktion vor, bevor die Datensätze dargestellt werden, wodurch sich Gruppierungen, basierend auf der Ähnlichkeit der Datensätze, bilden; (ii) Überblickstechnicken, die ausgewählte Attribute darstellen, um Muster und Verknüpfungen zu finden; (iii) tabellarische Visualisierungen, welche auch die einzelnen Werte der Datensätze anzeigen und so ihre Analyse und detaillierte Exploration erlauben.

Doch während die Analyse einzelner Datensätze in Tabellen einfach ist, wird das Finden von Ähnlichkeiten in der restlichen Datensammlung mühevoll. Auch bei Überblickstechnicken stößt man beim Vergleich großer Datenmengen schnell an Grenzen.

In der vorliegenden Arbeit wird ein Prozess zur Führung von Nutzerinnen und Nutzern präsentiert, um sie bei der Datenexploration zu unterstützen. Ausgehend von einer selektierten Teilmenge an Daten, werden Attribute vorgeschlagen, welche Ähnlichkeiten zu jenen aufweisen.

Attribute oder Datensätze können ausgewählt und anhand mehrerer Ähnlichkeitsmaße mit der gesamten Datensammlung verglichen werden . Ein selektiertes Attribut wird mit allen weiteren Attributen verglichen. Datensätze werden allerdings mit sämtlichen Gruppierungen aller Attribute verglichen. Numerische Attribute werden so diskretisiert, dass eine der resultierenden Gruppen möglichst hohe Ähnlichkeit aufweist. In hierarchischen Attributen wird nach dem ähnlichsten Teilbaum gesucht.

Die vorgestellte Benutzerführung ist nicht nur unabhängig vom Datentyp, sondern auch von der Domäne und der Visualisierung der Daten. Demonstriert wird dies durch die Verwendung verschiedener Datensammlungen und der Integration des Prozesses in zwei Visual Analytics (VA) Systeme. Medizinerinnen und Mediziner des Kepler Universitätsklinikums Linz verwenden diese VA Systeme zur Analyse vergangener Krebstherapien. Sie wurden regelmäßig in die Entwicklung eingebunden um die Benutzerführung und Darstellung der Ähnlichkeitsmaße zu verbessern.

Abstract

Seeking relationships in multi-dimensional datasets is a common task, but can quickly become tedious due to the heterogeneity and increasing size of the data. Its visualization can be approached in a variety of ways: (i) projection techniques decrease the number of dimensions to a fraction before visualizing items, creating clusters where similarities in the high-level space may be derived; (ii) overview visualization techniques display selected attributes and all of their items' values to discover patterns and find relationships; (iii) tabular techniques give an insight into the individual items and thus favor their detailed analysis and exploration.

However, while the interactive selection of a data subset during exploration is most easily done with tabular visualizations, finding relationships and patterns is not. Also, with overview techniques the number of attribute combinations quickly outgrows reasonable dimensions.

In this thesis, a data-driven touring process for Visual Analytics (VA) tools is presented that guides users in discovering relationships for a data subset of their interest. Based on the user's selection, attributes that show some kind of similarity are presented.

The selection can be done on attribute and item level. While a selected attribute is compared to all other attributes in the dataset, item sets are compared to the individual categories of attributes. This comparison can be based on a number of similarity measures. To cope with heterogeneity of data types, numerical attributes are discretized to achieve maximum similarity. In hierarchical attributes, the most similar subtree is sought.

The touring process is also independent of the data domain and its visualization. This independence is demonstrated by the use of three different datasets and the integration of the touring process into two VA systems.

These extended systems were shown to medical experts of the Kepler University Hospital, who will use them in the near future. Their feedback was incorporated to improve the guidance process.

Contents

Kurzfassung **ix**

Abstract **xi**

Contents **xiii**

1 Introduction **1**
1.1 Terms . . . 1
1.2 Problem Statement . . . 2
1.3 Aim of the Work . . . 3
1.4 Methodological Approach . . . 3
1.5 Structure of the Work . . . 5

2 Related Work **7**
2.1 Multi-Attribute Data Visualization . . . 7
2.1.1 Point-based Data Visualization . . . 7
2.1.2 Line-based Data Visualization . . . 9
2.1.3 Region-based Data Visualization . . . 9
2.1.4 Overview Visualization Techniques . . . 11
2.1.5 Projection Visualization Techniques . . . 13
2.1.6 Tabular Visualization Techniques . . . 16
2.1.7 Multidimensional Genomic Data Visualization . . . 19
2.1.8 Clinical Data Exploration Tools . . . 26
2.2 User Guidance in Visual Analytics . . . 28
2.3 Similarity Measures . . . 30
2.3.1 Numerical Similarity Measures . . . 30
2.3.2 Categorical Similarity Measures . . . 32
2.3.3 Hierarchical Similarity Measures . . . 34
2.4 Discretization of Numerical Attributes . . . 37

3 Touring Process Concept **39**
3.1 Touring Approach . . . 41
3.2 Guidance Model . . . 41
3.3 Similarity Measures . . . 43

3.3.1 Attribute Similarity Measures . . . 43
3.3.2 Item Set Similarity Measures . . . 44
3.3.3 Discretization of Numerical Attributes . . . 47
3.3.4 Similarity with Hierarchical Attributes . . . 49
3.4 Visual Integration . . . 49

4 Implementation of the Touring Process **53**
4.1 Caleydo Phovea Platform . . . 53
4.2 Server-Side Similarity Scoring . . . 54
4.2.1 API . . . 56
4.2.2 Similarity Measures . . . 57

5 Integration of the Touring Process **61**
5.1 StratomeX . . . 62
5.2 Ordino . . . 64

6 Results and Feedback **69**
6.1 Results . . . 69
6.1.1 Server-Side Performance . . . 70
6.1.2 StratomeX . . . 70
6.1.3 Ordino . . . 74
6.2 Feedback . . . 76
6.2.1 Feedback to StratomeX Integration . . . 77
6.2.2 Feedback to Ordino Integration . . . 77

7 Conclusion and Discussion **79**
7.1 Discussion . . . 80
7.2 Future Work . . . 81

A Appendix **83**

List of Figures **87**

List of Tables **89**

List of Listings **91**

List of Equations **93**

Acronyms **95**

Bibliography **97**

CHAPTER 1

Introduction

Across a broad range of research areas, analysis and exploration of data is an everyday task to discover connections, patterns, and gain insights. Healthcare, for example, has always been an area where large amounts of data are recorded and used for treatment, even more so through medical and technical advancements and the transition to Electronic Health Records (EHRs). In cancer genomics, advances in high-throughput sequencing continuously increase the amount of produced data [27; 96]. Genomic data takes a highly relevant role in cancer. The Cancer Genome Atlas (TCGA) collected data for 33 cancerous disease types with over 11.000 cases in total [70]. For each case, clinical data is stored together with biomolecular data, such as gene expression, methylation, or copy number data, recorded using high-throughput sequencing [54].

This ever increasing amount of gathered data provides many opportunities such as drug target identification and clinical decision support, improving patient care and reducing costs [89]. However, the amount of data recorded outgrows the ability to process it and time required for analysis increases.

The next section shortly describes the terms used for the different data elements. Subsequently the problem to be solved and the aim of this thesis are discussed. The methodological approach follows before the chapter is concluded by outlining the remainder of this thesis.

1.1 Terms

In this thesis the following terms are used to refer to certain elements of data:

- A *value* is the fundamental form of data, representing a single number or string.
- An *attribute* is a list of *values* for a particular observation. All values of an attribute share the same type and can be numerical, describing the observation

with continuous numbers, categorical, or hierarchical, like the diagnosis codes discussed in the next section.

- An *item* is also a list of *values*, but across multiple observations. The *values* for the different *attributes* thus describe an item.

- A *dataset* collects all values of all items for every attribute.

A dataset of music artists, for example, would contain attributes such as genre, number of released records, and year of first live performance. In the dataset, artists are represented by items and described by their individual values for the attributes.

1.2 Problem Statement

With the diversity of data types found in large datasets, specifically in healthcare where data is coming from doctors, nurses, medical devices, external parties, and so forth, date, text, boolean, and numerical values are recorded. Furthermore, the data can be structured in categories and hierarchies. To insure interoperability between healthcare providers, standardized formats for medical data have been established in the past.

Examples for this standards are Fast Healthcare Interoperability Resources (FHIR) [39] for medical data exchange, Digital Imaging and Communications in Medicine (DICOM) [71] specifically for imaging data, Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT) [101] as a common clinical terminology, Logical Observation Identifiers Names and Codes (LOINC) [90] for laboratory observations, or International Classification of Diseases (ICD)[1] [129] for disease classification and its enhancement, ICD for Oncology (ICD-O) [128], particularly for topology and morphology classification of tumors that both have a hierarchical structure. Despite standardization, the heterogeneity remains and multi-attribute datasets with numerical, categorical, and hierarchical attributes need to be analyzed.

To cope with data's vast volume or inherent complexity, the analysis requires both, computational power and human reasoning [11]. Visual Analytics (VA) tools provide interfaces that combine these two to allow users to explore and reason from the data.

If data is explored with VA, users may already have hypotheses to test, about correlations or patterns in the data, or want to see effects that different characteristics cause. There might be additional attributes, which are not in the users' mind, but show similarities to the attributes they are using. However, comparing attributes to discover new links and correlations quickly becomes tedious, as the possible combinations of two attributes grow quadratically with their number. Equation 1.1 shows the formula of the binomial coefficient for combinations of two out of `n` elements. As a consequence, only attributes

[1] Full official name is *International Statistical Classification of Diseases and Related Health Problems.*

considered as promising by the expert are compared, while others are ignored.

$$\binom{n}{2} = \frac{n!}{2!(n-2)!} = \frac{n(n-1)}{2} = \frac{n^2-n}{2} \tag{1.1}$$

If the data increases in size and complexity, analysis tools do as well as more attributes and options become necessary [11; 44]. Inevitably, the analysis takes longer and training may be needed.

1.3 Aim of the Work

VA tools can assist their users in analyzing complex data by offering some sort of guidance. Be it the selection of appropriate VA methods or even infrastructure, structuring the analysis into tasks, retrieval of attributes and subsets that are interesting, or suggestion of an expert for a task [11].

This thesis focuses on data-driven guidance, although we will see that guidance in the remaining domains may be desirable in the future. With vast numbers of attributes to choose from, VA tools can direct users to the most promising ones and support them in gaining further insights.

Following the model for guidance in VA proposed by Ceneda et al. [11], users often know their target—to find some kind of relationship—but need some *directing* towards *paths* that lead them to it. With inputs like the currently used data subset, similarity measures can be calculated to suggest attributes, ranked by the resulting scores, to the users.

In this thesis we propose a data touring system that is independent of the data domain and the VA tool applied. Data of different types, be it numerical, categorical, or hierarchical data, are compared to find similar items inside the dataset and assist users in their exploration and direct them towards potentially interesting relationships. In this way, users are assisted in proceeding their analysis from the current state with the offered directions.

1.4 Methodological Approach

Three heterogeneous multi-attribute datasets, featuring clinical, genetic, and pathological data have been used to demonstrate data domain and data type generalization. The first is a data collection about *glioblastoma multiforme* [110]—a brain tumor—and the second about *clear cell renal cell carcinoma* [111], both from TCGA [72] and publicly available. The third dataset is a confidential excerpt of the Kepler University Hospital's tumor database, which also features diagnosis and tumor descriptions in the hierarchical ICD-10 and ICD-O standards mentioned above.

The touring system has been integrated into two VA tools, *StratomeX* [106] and *Ordino* [107] to demonstrate independence of the applied visualization-tool. StratomeX [106]

displays attributes as columns of stacked blocks. Each block represents an item set and is sized based on the number of items inside it. Blocks of adjacent columns that share items are connect by ribbons. The width of each ribbon is proportional to the number of items shared by the connected blocks. The resulting visualization thus results in parallel sets [4; 52]. Blocks and ribbons can be selected to specify the input for the touring system. Ordino [107] displays the data tabularly with an option to switch into overview mode where patterns and distributions can be seen more clearly. The tabular approach enables users to select individual items, giving them more freedom in specifying the input data for the touring system.

The touring system augments these tools, enabling users to query for similar data. Whether two data subsets are similar or not is assessed by various similarity measures for different data types. If the items of numerical attributes behave similar can be seen by their correlation, but they can also be considered similar if their mean, variance, or distribution match. For sets of items, their number of shared items divided by the sets' total number of unique items is a common similarity measure known as Jaccard index.

The touring process has first been integrated into StratomeX, using the Jaccard index as similarity measure to compare a selected item set with the categories of categorical attributes. The Jaccard index calculation was then extended to include hierarchical attributes. Plain percentages of overlap with the given and compared data sets respectively were added after gathering first feedback from users.

We also added the Pearson correlation coefficient (PCC) for comparison of numerical attributes. To compare numerical data with item sets, and thus categories, a discretization algorithm was integrated that bins a numerical attribute such that the resulting groups' similarity is maximized. The implemented measures were selected as they are among the most commonly used ones [13; 28; 66].

Integration in both VA tools involves solely selection of input, starting similarity tasks with calls to a Representational State Transfer (REST) Application Programming Interface (API) and displaying the resulting scores to the users. Hence, the proposed touring process can be easily integrated into any desired data exploration tool.

This thesis work is part of the TourGuide project [105], a research platform to record and analyze clinical data in order to improve patient care, covering data analysis itself, but also the acquisition, handling, and data modeling. While the touring process introduced in this thesis is independent of the data domain and type, qualitative feedback by medical professionals from the Kepler University Hospital, the future users and partners in the TourGuide project, was gathered. As they are first of all treating patients and using the tool to improve and target care, simplicity was more important than feature richness in the design considerations.

1.5 Structure of the Work

The next chapter presents work related to this thesis. First, appropriate visualization methods are discussed, followed by ways to guide users in VA tools. Section 2.3 presents similarity measures for data comparison. The chapter is concluded with discretization techniques for numerical attributes in Section 2.4.

The concept of the touring process is described in Chapter 3. We describe the overall touring process, the used similarity measures, how numerical attributes will be binned, how hierarchical items are compared, and concepts for the visual integration.

Chapter 4 explains the implementation of the touring process. It covers how the data is compared and scored and which additional software packages are used to do so.

The integration of the touring process is described in Chapter 5. We describe StratomeX and Ordino, how the user interface has been adapted, and how the touring process can be used in the two tools.

The resulting touring process and its visual output are presented in Chapter 6, together with some performance measures. Responses from medical professionals and the implications from their feedback are covered as well.

Finally, the thesis ends in Chapter 7 with a conclusion of our work and a discussion. The chapter also provides thoughts on the possible future work.

CHAPTER 2

Related Work

The guidance method introduced in this thesis is demonstrated on clinical and oncogenomic data exploration. In Section 2.1 we discuss applications and methods to visualize such multi-dimensional data. Related work regarding user guidance in VA is discussed in Section 2.2. Section 2.3 presents similarity measures for numerical, categorical, and hierarchical data. The chapter is concluded by methods to discretize numerical data (Section 2.4).

2.1 Multi-Attribute Data Visualization

For the visualization of multi-attribute data a myriad of approaches exists, developed and refined to support the exploration in different use-cases or with different data.

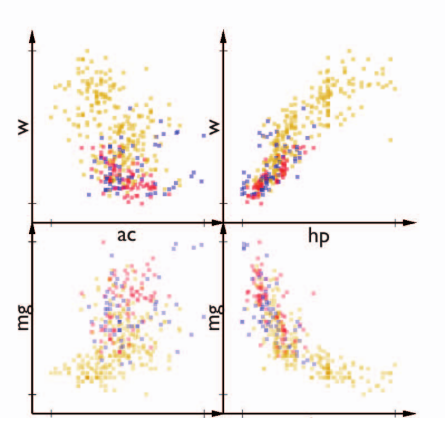
First, a couple of visualization techniques are discussed. Following Gratzl et al. [35] and Ward et al. [122], we divide them into three groups: point-, line-, and region-based visualizations (Sections 2.1.1 to 2.1.3).

These visualizations are adopted by several tools for multi-dimensional data exploration. These tools are discussed in Sections 2.1.4 to 2.1.6, divided into overview techniques, projection techniques, and tabular techniques [25].

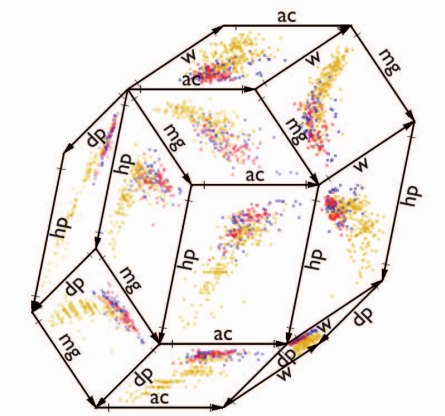
Related work regarding the visualization of our used data is discussed as well. Section 2.1.7 presents genomic data visualizations and Section 2.1.8 tools to visualize and explore clinical data.

2.1.1 Point-based Data Visualization

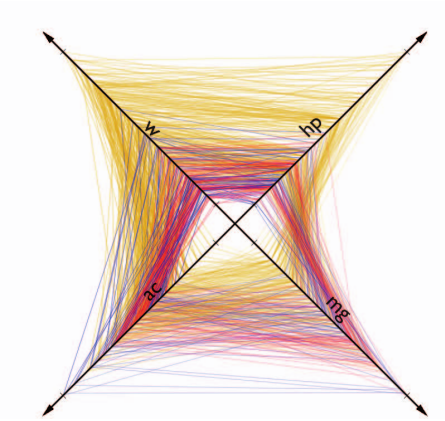
A widely used method for visualizing multi-attribute data and probably the best known point-based visualization to compare two attributes is a scatter plot [14]. In a Cartesian



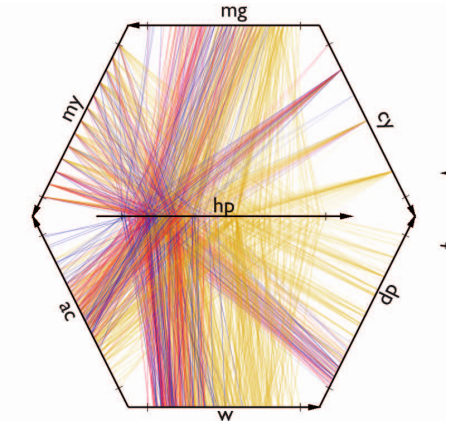
(a) Scatter plot matrix for pairwise comparison of four attributes.



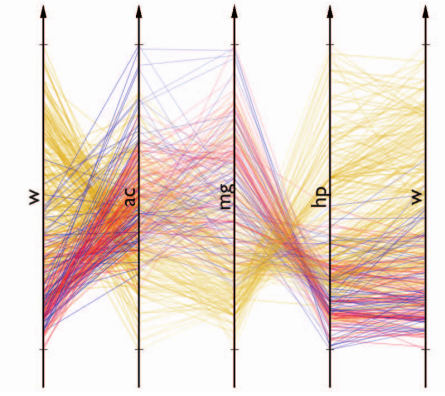
(b) Hyperbox with pairwise comparison of five attributes.



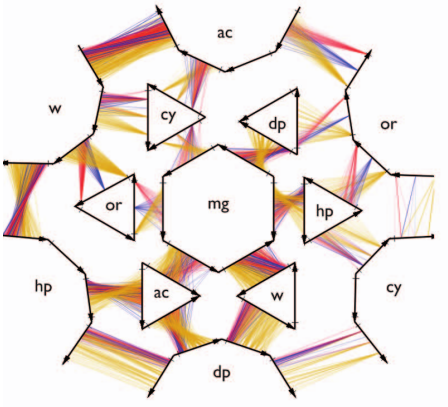
(c) Radar chart with four attributes.



(d) Time Wheel comparing multiple attributes to horse power (hp).



(e) Parallel coordinates plot (PCP) with four attribute comparisons.



(f) A many-to-many parallel coordinates plot (PCP) with seven attributes.

Figure 2.1: Line- and point-based visualizations from Flexible Linked Axes (FLINA) [14]. They show attributes of a car dataset with weight (w), acceleration (ac), horse power (hp), miles per gallon (mg), displacement (dp), cylinders (cy), model year (my), and origin (or).

coordinate system each axis represents an attribute. The values of the item for each attribute determine its position in the plot.

Scatter plots suffer from the limitation that only two attributes can be compared, which is circumvented by variations like hyperboxes [3] and scatter plot matrices [14]. Both compare multiple pairs of attributes in scatter plots that are arranged next to each other. Their composition can be seen in Figure 2.1. While a scatter plot matrix arranges the individual plots equally in a grid (see Figure 2.1a), a Hyperbox "is a two-dimensional projection of an N-dimensional box" [14]. As a result of this projection, each scatter plot varies in orientation and skewness, leading to focused and peripheral plots (see Figure 2.1b).

2.1.2 Line-based Data Visualization

Examples of line-based visualizations are *radar charts*, *time wheels*, and *parallel coordinates plots (PCPs)*. They have in common that the attributes are represented as axis on which the items are mapped according to their actual numerical value for that attribute. Categorical attributes have their categories evenly spread along the axis. Lines connect the values of an item between axes. How the attributes' axes are aligned and which ones are connected differentiate the three methods.

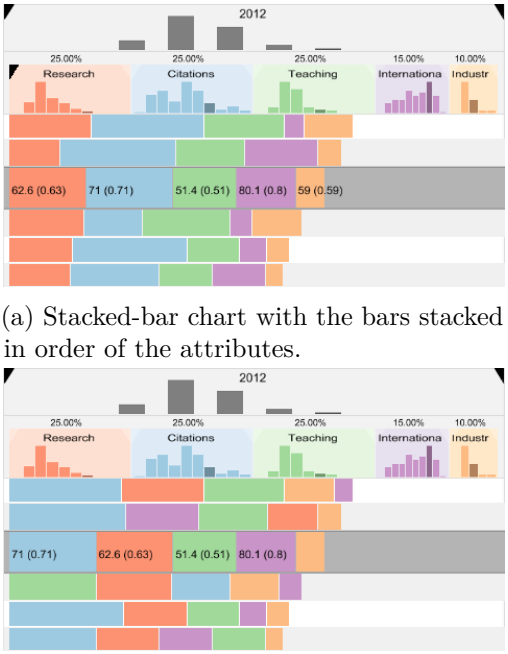
In a radar chart, or a star plot, the attributes' axes are aligned in a star pattern, and the items' values are connected between adjacent axes (compare Figure 2.1c). The time wheel has an axis of reference in the center and further axes radially aligned around it [116]. Lines connect the item values on the radial axes with the horizontal axis (see Figure 2.1d).

In a PCP, the axes are aligned parallely and item values on adjacent axes are linked similar to radar charts (see Figure 2.1e) [45]. A variant of PCPs is the many-to-many relational PCP described by Lind et al. [58]. They propose axes configurations so that the many-to-many PCP can show relationships between each of four or seven attributes at once [58]. Arrangements for three, five, six, and eight attributes are shown by Claessen and van Wijk [14]. An example with seven attributes is given in Figure 2.1f.

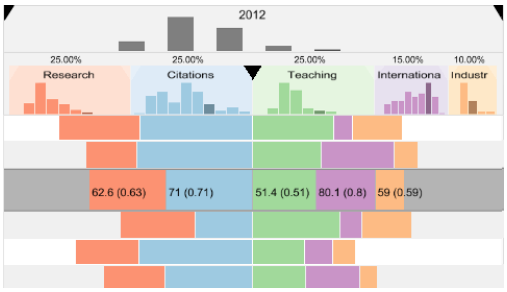
2.1.3 Region-based Data Visualization

Region-based approaches can represent values by bar lengths, size of pie slices, or area variation. Pie charts can be used to inspect the distribution of an attribute's items into its categories. If the data is quantitative or an ordering can be inferred, an item's values for multiple attributes can be compared with various bar charts. Multi-bar charts align all bars on a common baseline so that the users can clearly see differences between individual attributes. Stacked-bar charts can be used to compare totals from multiple attributes by aligning their bars, creating a single one (see Figure 2.2a) [104].

By varying the order or baseline of stacked-bar charts, comparison of individual attribute values can be simplified. Ordering the attributes in a stacked-bar chart by their

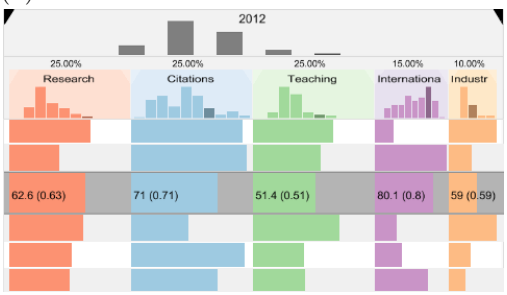


(a) Stacked-bar chart with the bars stacked in order of the attributes.



(b) Stacked-bar chart with a shifted baseline.

(c) Stacked-bar chart with the bars ordered according to their length.



(d) Stacked-bar chart with a baseline for every attribute.

Figure 2.2: Stacked-bar chart variations from LineUp [35] (also see Section 2.1.6).

individual length, contributions to the total length can easily be seen from the order of attributes (see Figure 2.2c). Changing the baseline makes comparison of the total sum more difficult, but helps users to compare the values of the attributes next to the baseline [104]. Additionally, the baseline can be placed such that attributes with a negative and positive association are separated and their stacked bars grow in opposite directions (as in Figure 2.2b) [35]. Instead of a single baseline, each attribute can be aligned on its own baseline (see Figure 2.2d). With all bars horizontally aligned, single attribute values are compared easily.

Area charts, where the area between line and axis is colored, can be used in similar ways to bar charts to display data over time. By overlaying multiple areas, they have a common baseline and can be compared. By stacking areas on top of each other, one has the same possibilities as with stacked-bar charts (compare Figure 2.2).

Stream graphs [7] are a variation of stacked area graphs that have a varying central baseline (see Figure 2.3). Stream graphs work well if the displayed categories start and end at different times and are useful to discover patterns and trends over time as in Figure 2.3 [49; 92]. In contrast to area charts, stream graphs do not support negative numbers and focus on the overall representation, rather than values on individual points [40].

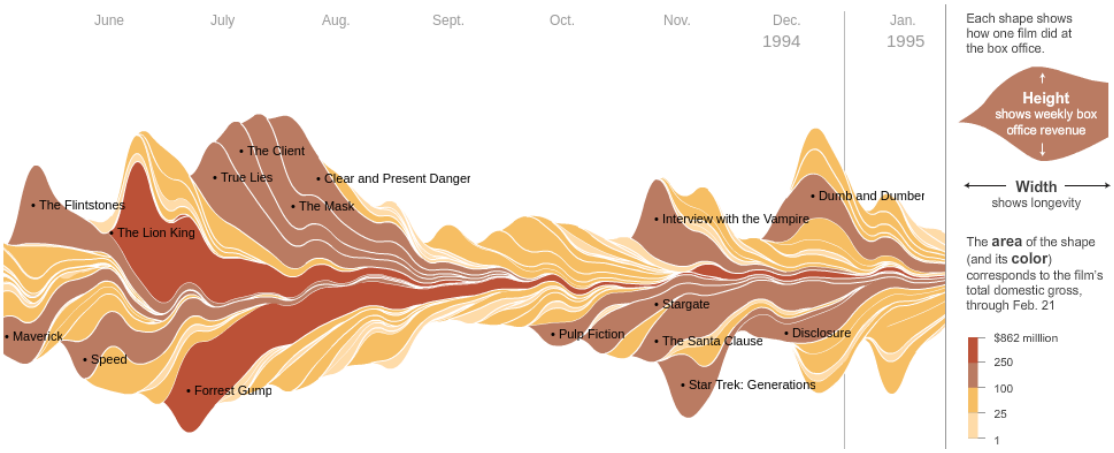


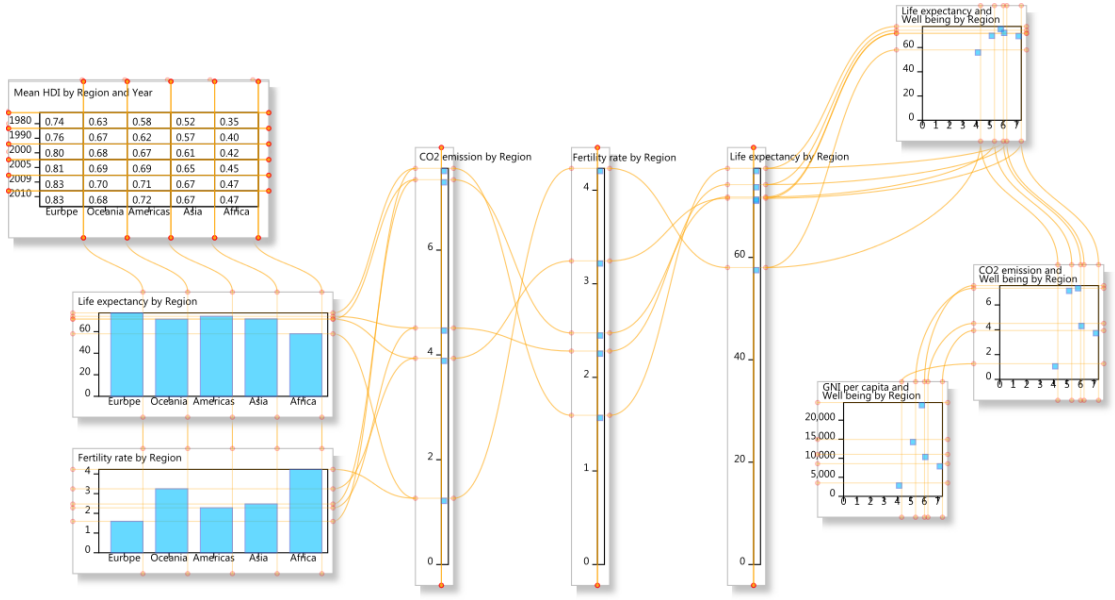
Figure 2.3: A stream graph showing inflation adjusted movie revenues between May 1994 an January 1995 [112]. The color indicates the overall success of a movie, while the shape shows if it was an instant hit on the opening weekend and how long it could attract viewers. *The Lion King* for example, was released in June 1994 and was an instant hit, but also regained popularity towards the end of 1994.

2.1.4 Overview Visualization Techniques

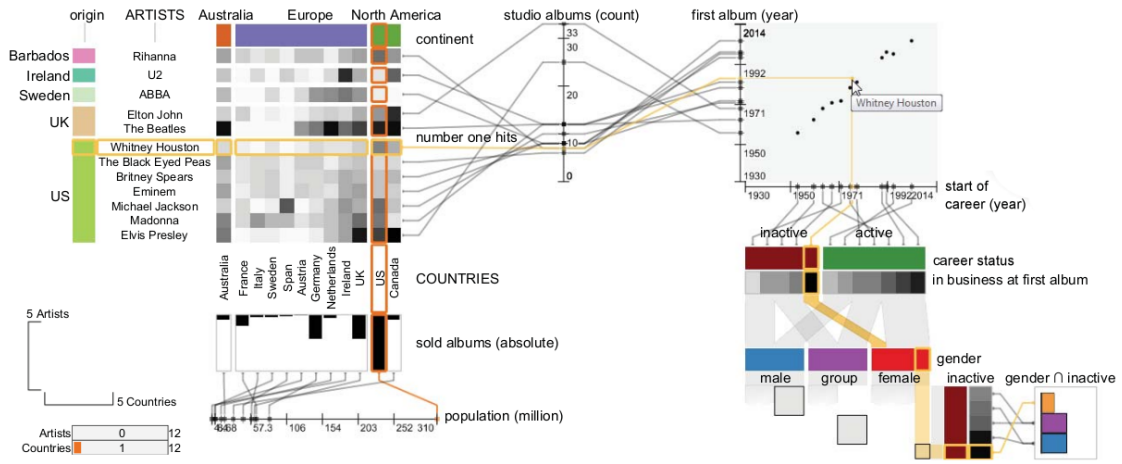
Overview techniques incorporate the attributes in visualizations to see relationships and distributions on a large scale of items (see Figure 2.4). *Flexible Linked Axes (FLINA)* [14] is a generalization of axis-based techniques, as described in Sections 2.1.1 and 2.1.2, and allows the users to create visualizations by drawing and linking axes on a canvas (see Figure 2.1). All of the discussed point-based and line-based visualizations are supported by FLINA. Multi-dimensional data and visualizations that are not axis-based, like pie charts or stream graphs of Section 2.1.3, are not supported

ConnectedCharts [118] (see Figure 2.4a) enhances the idea of Claessen and van Wijk [14] and removes the data and visualization restrictions discussed above. Thus, non-axis-based visualizations like bar charts and additional two-dimensional charts are available. ConnectedCharts also handles tabular datasets and supports data aggregation.

To link visualizations not only on an item level, but also between subsets or whole datasets, Gratzl et al. further refined the approach and developed *Domino* [36]. Domino excels in data exploration by the ability to define, visualize, and relate subsets of the data (see Figure 2.4b) and also integrates FLINA [14] and ConnectedCharts [118] as data visualization techniques.



(a) ConnectedCharts [118] with tabular data connected to bar charts, parallel coordinates plots (PCPs), and scatter plots.



(b) Domino [36] showing relations inside a musical dataset with a heatmap, a scatter plot, parallel sets and parallel coordinate plots, and a bar chart.

Figure 2.4: Examples of overview visualization techniques.

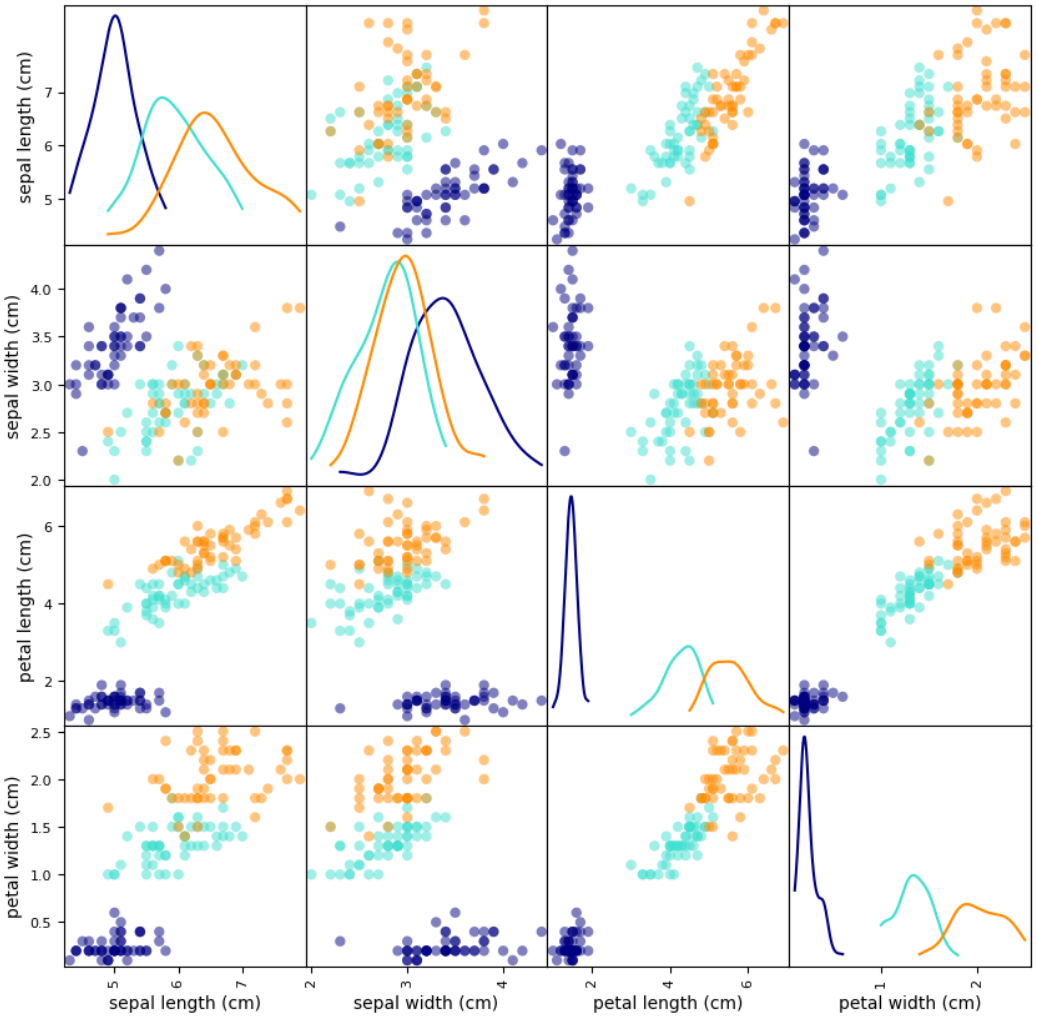
2.1.5 Projection Visualization Techniques

With projection techniques the dataset's dimensionality can be reduced before the items are visualized or by the visualization itself, e.g., with principal component analysis (PCA) [1], Multidimensional Scaling (MDS) [59], or t-distributed stochastic neighbor embedding (t-SNE) [61]. Figure 2.5 demonstrates these projection techniques with Fisher's Iris flower dataset [20]. The dataset was transformed using Scikit-learn [78], a Python [88] library that offers the three projection techniques, and visualized with Matplotlib [43], a Python plotting library.

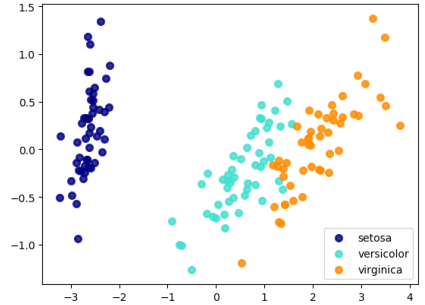
High-dimensional data is translated to the projection space with typically two dimensions, showing the data's information in a planar layout. In this projection, the positioning of individual items conveys information about their similarity to other items; with similar items being closer and unrelated items being placed farther apart. Groupings of items and patterns can easily be seen in these projections and studies have shown that they outperform scatter plot matrices and three-dimensional visualizations [102]. A drawback, due to the dimensionality reduction, is that the resulting plots have no meaningful axes, which makes it difficult to translate similarities or differences in the projection space to the actual attributes [102].

The technique developed by Stahnke et al. [102] use MDS for dimensionality reduction and show the result in a two-dimensional scatter plot. By showing the error, introduced by the projection, as halo around items, reliability can be determined during data exploration. Items can be selected and grouped to be analyzed and compared to the dataset. Analyzing items in the projection, the distribution of the original attributes' values and their distribution in the projections are of interest. This can be seen with density plots on the side and heatmaps as an overlay on the projection (see Figure 2.6). This helps users to counter the above mentioned drawback of projection techniques. By plotting attributes over the projection, it can be seen as a hybrid solution that also gives an overview of the dataset.

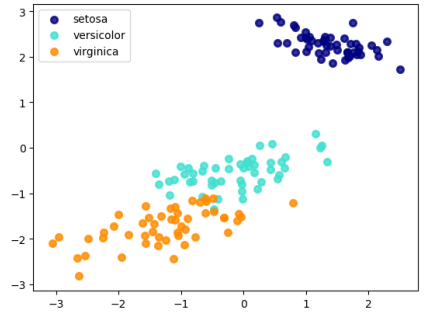
ProxiLens [41] also uses a projection of high-dimensional data to a scatter plot for data exploration and features a semantic lens, with which an area in the plot can be focused. The item in the center of the lens serves as reference to calculate the distance in the high-dimensional space to items adjacent in the projection. The lens moves so called *false neighbors*, items which are close in the projection but not in the high-dimensional space, to its border. The distance of *true neighbors* is visualized in the focused area by color, which is getting brighter with proximity (see Figure 2.7).



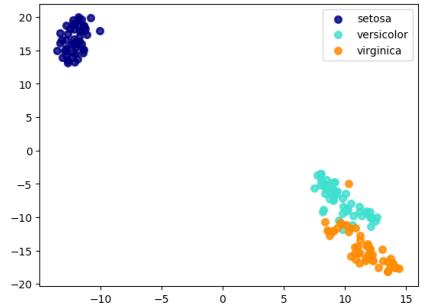
(a) Scatter plot matrix with the dataset's attributes. The diagonal shows a histogram of each species for every column. Setosa = navy, Versicolor = turquoise, Virginica = orange.



(b) Dataset projection with PCA.



(c) Dataset projection with MDS.



(d) Dataset projection with t-SNE.

Figure 2.5: Visualizations of Fisher's Iris flower data set [20]. It contains 50 measurements of length and width for petal and sepal leaves for each of the three species (Setosa, Versicolour, and Virginica).

Figure 2.6: Probing Projections [102] visualization with a heatmap overlay showing the value distribution of *Educational attainment* in the projection space.

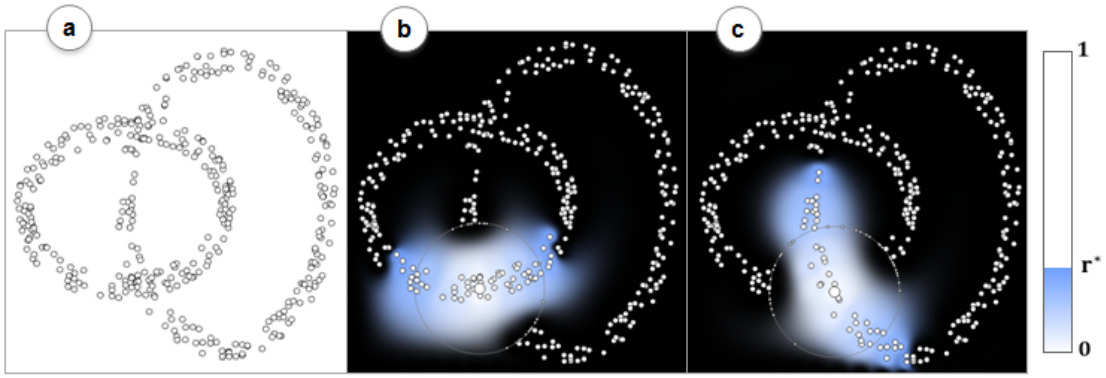


Figure 2.7: ProxiLens [41] applied to a two-dimensional projection (using principal component analysis (PCA)) of three-dimensional interlaced rings (Subfigure *a*). According to the high-dimensional distances between the points, false neighbors from the second ring are pushed towards the lens' border and have a black background, while the points of the true neighbors have a bright background (Subfigures *b* and *c*).

Figure 2.8: LineUp [35] with two rankings of universities that differ in attribute weightings. Changes in the ranking through the different weights are shown by slope graphs.

2.1.6 Tabular Visualization Techniques

If the items' actual values are of interest, a tabular arrangement is the visualization of choice. Widespread general applications like *LibreOffice Calc* [57] and *Microsoft Excel* [65] use spreadsheets to display multi-attribute data. They support sorting, grouping, and filtering and offer multiple chart options such as bar, pie, and bubble charts or scatter plots, but do not target users who want to interactively explore the data. Encoding values of items within the cells of a table is very limited in these tools and the direct comparison of data is tedious.

LineUp [35] (see Figure 2.8) visualizes multi-attribute data in a table as well and can encode numerical cell values with variations of bar charts, described above. Histograms are displayed atop each attribute to get a quick overview. Users can combine and weight attributes, merging their bars to a stacked-bar chart. This enables them to quickly create rankings of items based on multiple attributes or understand a ranking's composition. To compare rankings, a separator is inserted between them. Slope graphs within the separator connect identical items from both rankings.

Of course, tools may combine projection, overview, and tabular visualization techniques to a hybrid solution that gives an overview, but also provides insight into the individual item values.
Taggle [25] is a tabular technique that allows the analysis of each individual item's values and has the option to switch in an overview mode, where patterns and distributions can be seen more clearly. Large multi-attribute datasets are displayed in a scalable table that can be filtered and sorted. In the detail mode, numerical item values are represented by bar charts (see Figure 2.9).

Figure 2.9: Taggle [25] with football player data in detail mode. While players who shot with either the left or both feet are aggregated, data for those who shoot with the right foot is displayed in detail. A box plot visualizes the aggregated items for the attribute *age*, and a histogram summarizes the aggregated items of attribute *height*.

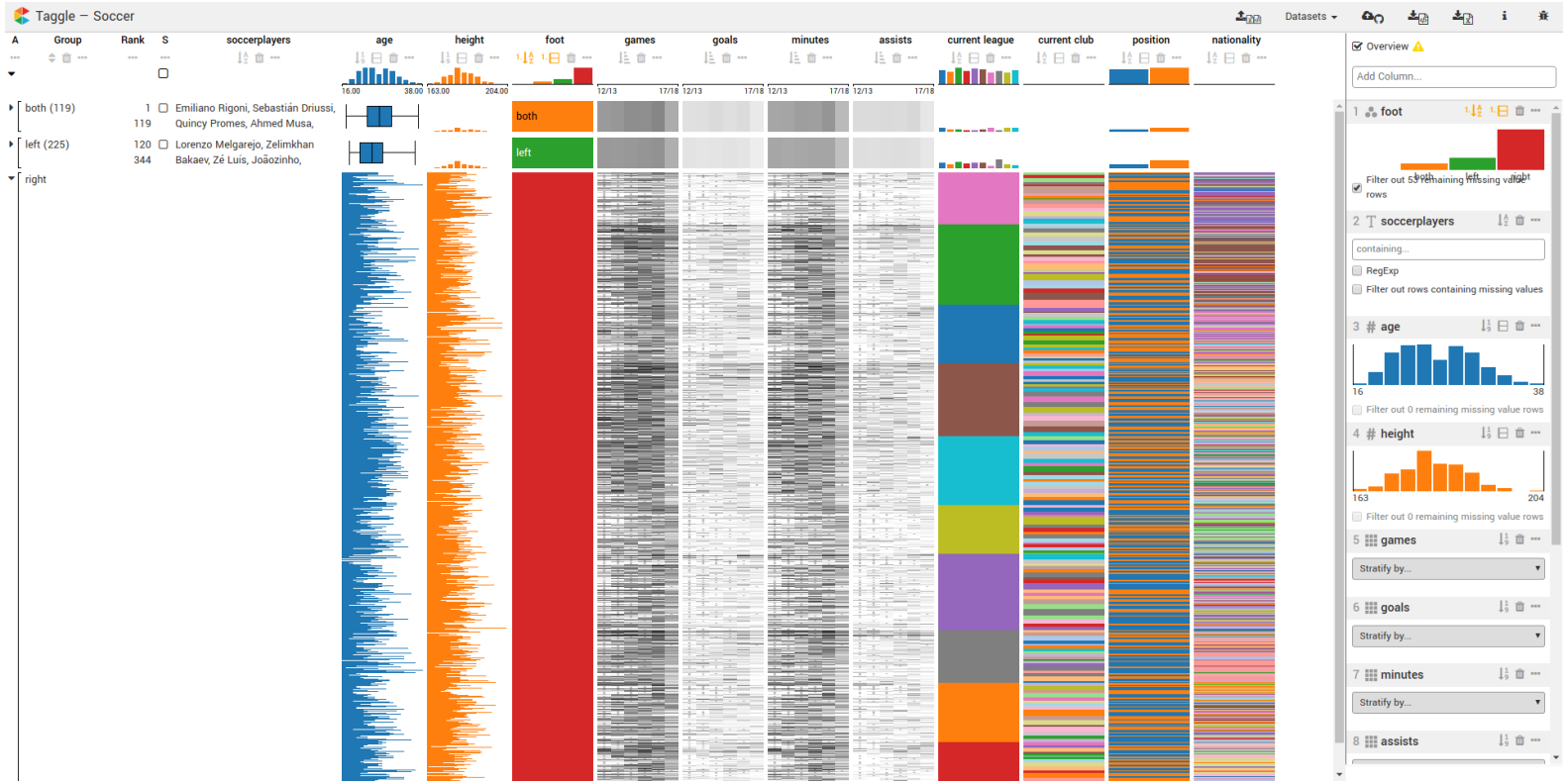


Figure 2.10: Taggle [25] with football player data in overview mode. Items of players shooting with the right foot are shrunk to a height of one pixel to fit into the available space.

Just like LineUp, Taggle can combine and weight multiple columns to create a total score, displayed as a stacked-bar chart. Categories of stratified columns are represented by colors. Groups can be created by splitting numerical attributes by a threshold or via categorical attributes. These groups can be aggregated to make space for the remaining items, summarizing the values of the aggregated items in choosable visualizations, such as bar charts, boxplots, or histograms (see Figures 2.9 and 2.10).
The overview mode tries to display as many items as possible by decreasing the items' height down to one pixel until all fit into the visible area (see Figure 2.10). Already aggregated groups of items keep their size, and selected items are displayed in full size as well so users can still inspect the values of interesting items. If a height of one pixel per item is not small enough to fit everything into the available space, scrollbars are introduced, until either the item quantity is reduced by filtering, or a group of items is aggregated. The data selection panel on the right hand side of Figures 2.9 and 2.10 is used to add and remove attributes, sort, stratify, and filter them.

2.1.7 Multidimensional Genomic Data Visualization

High-throughput genome analysis technologies can generate large sets of genomic data. Projects such as TCGA and the International Cancer Genome Consortium (ICGC) catalog this data for a variety of tumor types [96; 110]. Researchers may use these catalogs to gain further insights and to improve patient outcomes [110]. Visualizations specifically for genomic data are discussed separately in this section.

Schroeder et al. [96] reviewed visualization methods and tools developed for multidimensional genomic data and grouped them into in three different visualization approaches. The first approach is to visualize genomic coordinates, like popular genome browsers such as the *Integrative Genomics Viewer (IGV)* [113] or the University of California, Santa Cruz (UCSC) *Cancer Genomics Browser* do [96]. Both visualize a tumor sample's data along genome coordinates, which is the most common approach (see Figure 2.11) [96].
Circos [53] is a tool whose approach is different as it displays data circularly (see Figure 2.12) [53; 96]. If used for genomic data, chromosomes are represented by circular segments that form a ring [53]. In the segments genomic features of the chromosomes, such as mutations or copy number variation, are shown in layers [53]. The individual genes in these segments can be linked to highlight disease related genes or similar genome subsets [53].

Another way to visualize genomic data are networks. Networks can represent relationships and interactions among biologic structures. Genes and their encoded proteins build nodes that can be shaped or colored to reflect genetic features. *Cytoscape* [99] is a tool frequently used in bioinformatics, to analyze these interaction networks and identify important features such as cancer drivers and drug targets (see Figure 2.13) [96; 99]. In addition to the Java application, there is also a web version [60], a JavaScript library [24], and the *cBio Cancer Genomics Portal* [12], which implements a Cytoscape adaptation optimized for analysis of TCGA data [96].

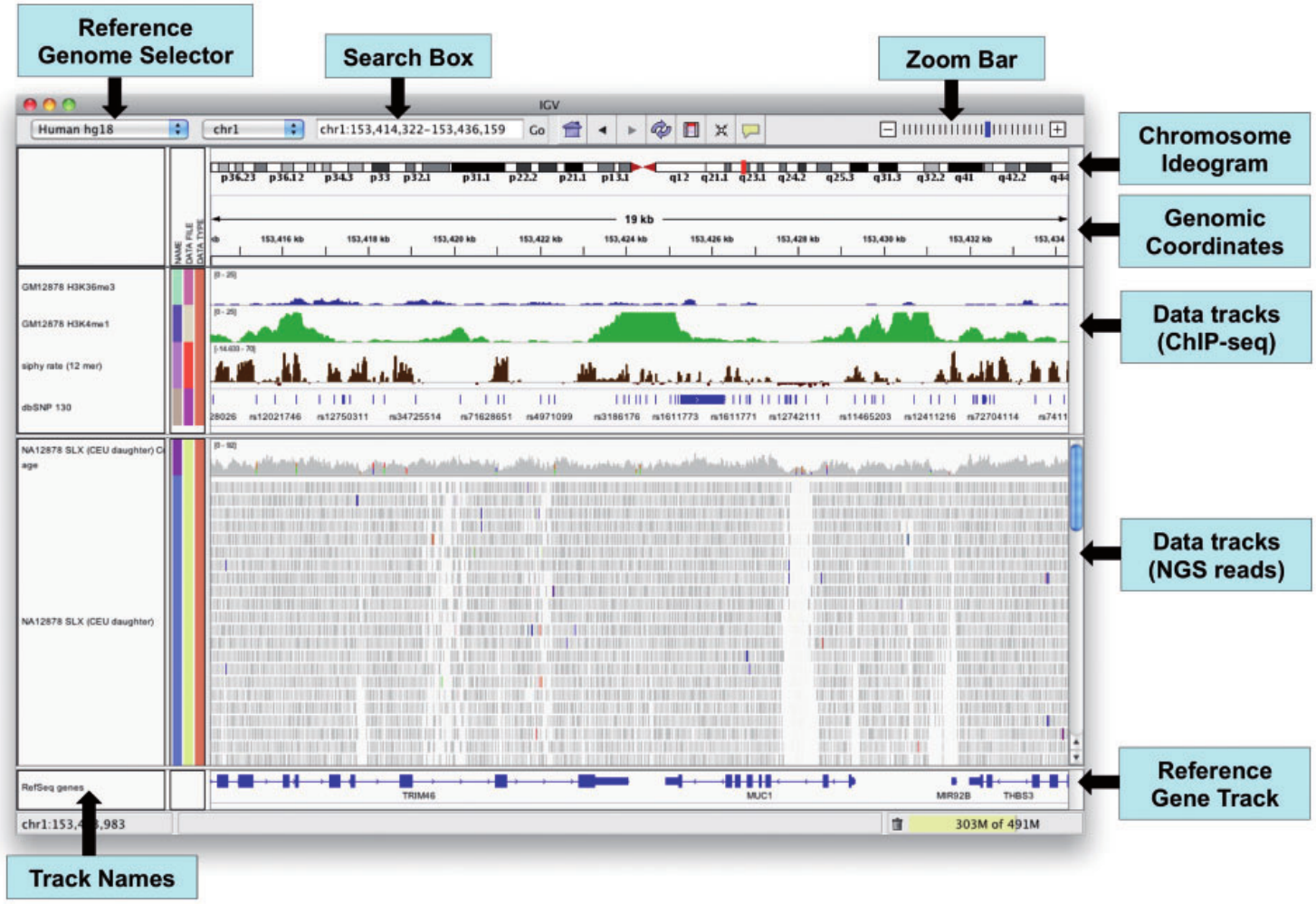


Figure 2.11: Integrative Genomics Viewer (IGV) application window. At the top, the chromosome ideogram and genomic coordinates display the currently analyzed section. Data of Chromatin immunoprecipitation DNA-sequencing (ChIP-seq) and Next-generation sequencing (NGS), two methods to acquire genomic data, is displayed in the center. The reference gene track at the bottom shows the encoded genes.

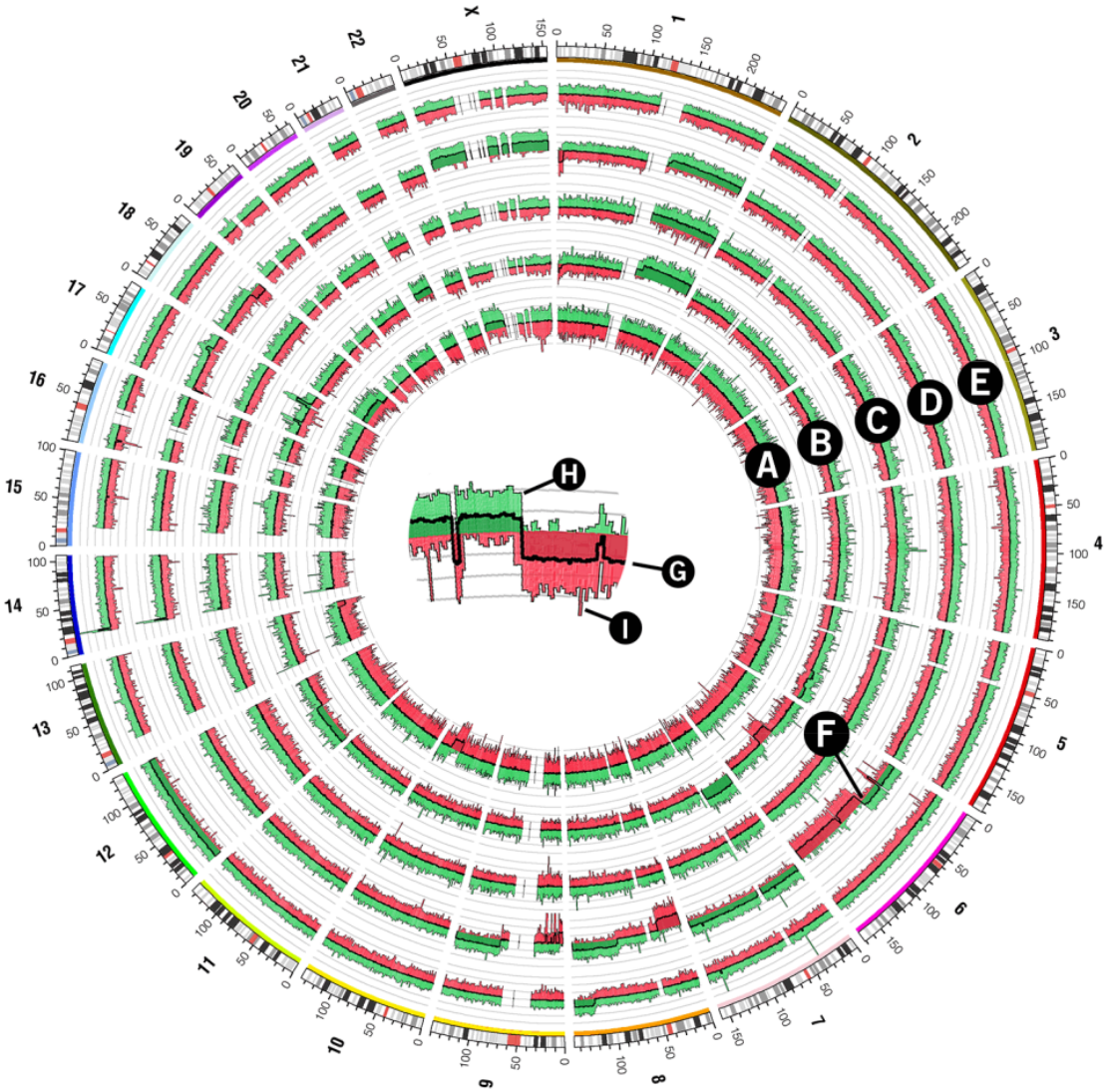


Figure 2.12: Circos [53] showing whole-genome copy number profiles of five tumor samples (histograms A-E) for every chromosome (segments 1-22 and X). The outer ring represents each chromosome with an ideogram. A segment's arc length is determined by the chromosome's number of Base pairs (bp), ranging from about 46 million base pairs (bp) (chromosome 21) to almost 250 million bp (chromosome 1). The copy number histogram F of chromosome 6 is displayed in the center of the visualization. The line G displays the average value of 250 probes. The color of the area between G and zero shows whether the copy number is increased or decreased on average. H and I show the maximum and minimum three-probe average, respectively.

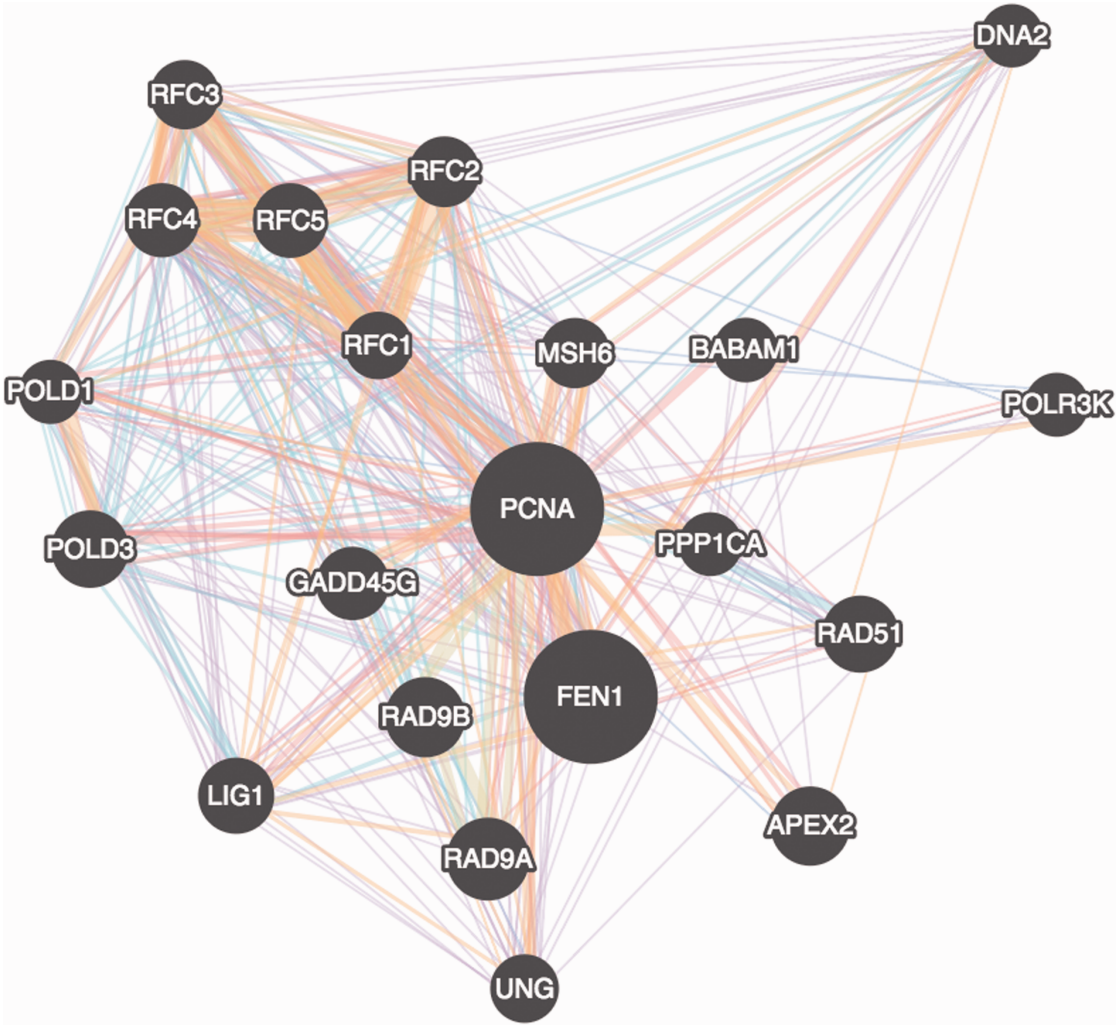


Figure 2.13: A gene-gene interaction network visualized with Cytoscape.js [24]. Interacting genes are connected by edges. The number of edges corresponds to the interaction strength. The edge's color shows the interaction type. The node's size is determined by its protein score.

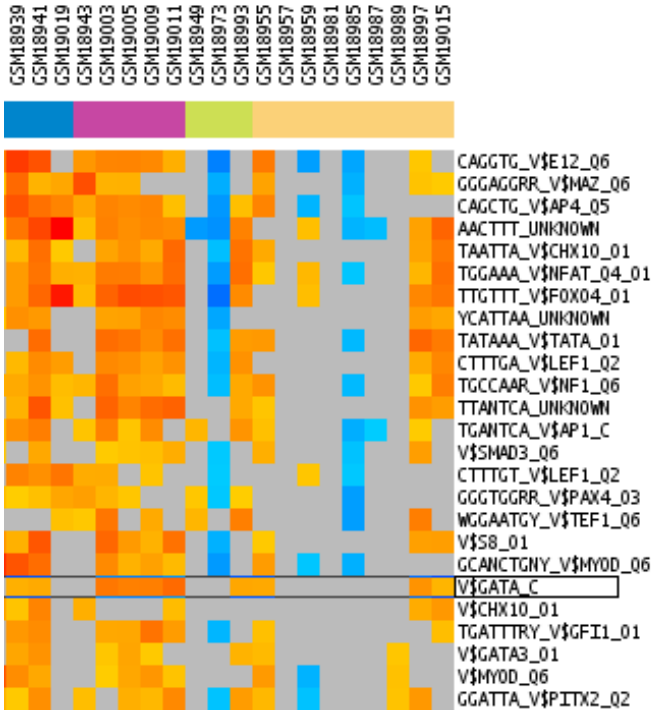


Figure 2.14: An example of a heatmap in Gitools showing the deviation of genes (rows) for various tissue samples (columns) from their respective mean expression level [29]. While values inside the 95 % confidence interval are gray, cells are colored from light to dark orange and blue if the z-score goes towards plus and minus 10, respectively.

The cBio Cancer Genomics Portal also offers a viewer called OncoPrint to depict genomic data with heatmaps [12]. A heatmap visualizes the data of a matrix by representing values with colors in a grid and has turned out as an intuitive and effective way to visualize biological data [79]. In oncogenomics this usually means that each row represents a genomic entity, such as a gene, and each column represents a sample or vice versa (see Figure 2.14) [56; 96]. Gitools and StratomeX are two more representants of this visualization technique [96]. The second tool is also used for an exemplary implementation of the thesis' guidance method, which is why Gitools and StratomeX are now discussed more thoroughly.

Gitools [79] has interactive heatmaps (compare Figure 2.14) in which users analyze and browse the loaded data. It offers enrichment analysis, a so called Oncodrive method, correlation calculation, analysis of overlap, and result combination [79]. With enrichment analysis, over-represented (or under-represented) biological characteristics of the gene subsets from the heatmap can be found [67; 108; 114]. Oncodrive finds rows in the analyzed data matrix that are significantly different from others with the idea to identify genes that are altered in tumorous data samples [79]. The correlation calculation can

be used to compare a column or row with the rest of the matrix [79]. Overlap analysis shows matching *true* elements in binary rows and columns [79]. To combine the resulting significance values of several tests on different datasets, a combined test of significance is created with the weighted Z-method [79]. Browsing the heatmap, users are also able to search, filter, and cluster it, to move rows and columns and customize the heatmaps appearance (by changing colors, sizes, and labels, for example) [79].

In contrast to Gitools, where users operate inside a heatmap, *StratomeX*'s [56] approach focuses on the comparison of multiple attributes and finding relationships between them [56; 96]. Genomic data is visualized with heatmaps. Users can add multiple heatmaps with genomic data, as well as categorical and numerical attributes, which can be used to include clinical data in the analysis [56]. All displayed attributes are connected by ribbons that reflect the number of shared items in their width (see Figure 2.15) [56]. Former a Java application, as reviewed by Schroeder et al. [96], StratomeX transitioned to a web application [37]. The web version is also used to demonstrate the integration of the proposed touring process (compare Chapter 3 and 5).

Ordino [107] is a web-based tool to display data in a table, with the features already discussed for Taggle in Section 2.1, and does therefore not fit into the categorization by Schroeder et al. [96] used above. Ordino is the second application in which we integrate our touring process (see Figure 5.2). Users can rank, filter, and explore attributes in the table and select an item subset for further analysis in detail views. Two-dimensional data can be visualized with heatmaps inside a column. The item's values from the matrix are shown as a bar of varying color. In detail views, additional information on the selected item subset is displayed. This can be done with an additional table, external resources (e.g., Ensembl [131]) or specialized visualizations, e.g., OncoPrint, already mentioned above for the cBio Cancer Genomics Portal. Additionally, users can save their current analysis state for later continuation, as basis for future analysis or for sharing with colleagues.

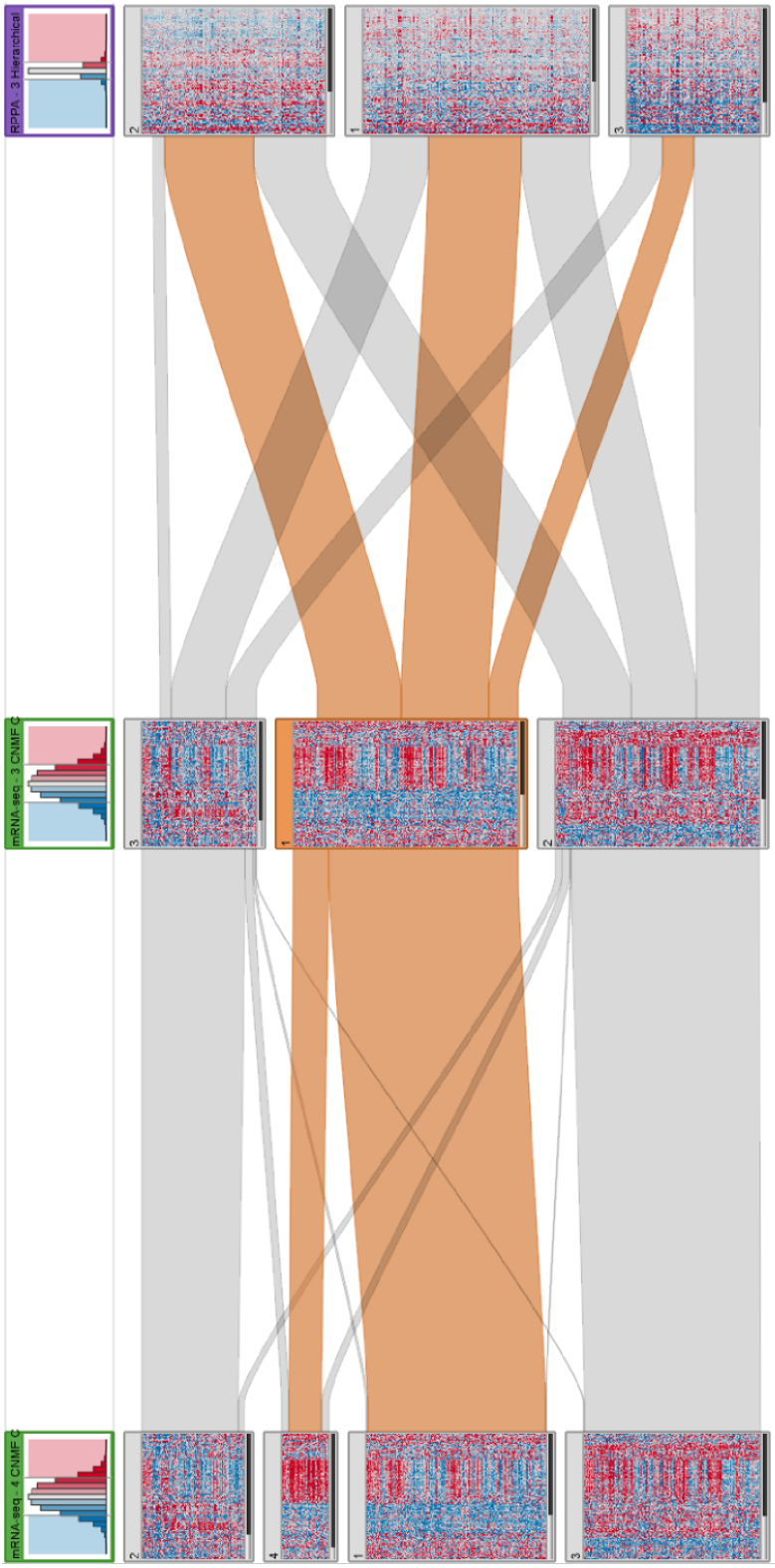


Figure 2.15: StratomeX [55; 106] with three heatmap columns. The ribbons' width represent the number of shared items in the categories. While the items are almost identically stratified in the first two columns, the third one shows little similarity. The subsets 1 and 4 of *Stratification 1* merge together into the subset 1 of *Stratification 2*.

2.1.8 Clinical Data Exploration Tools

Data with medical information is different to datasets with other content by legal, ethical, and social aspects, by the various existing standards, its myriad of sources (transcripts, laboratory values, imaging data, etc.) and the resulting heterogeneity [51]. An Electronic Health Record (EHR) collects this data, from different sources in different standards, over time and can be shared among care providers [123].

Visualizations for clinical data and EHRs have to represent more categorical data as the tools for genomic data discussed above. Additionally, clinical data and EHRs data is typically recorded over a period of time.

From an EHR, the history of a patient with information on symptoms, diagnosis, and treatments is at hand and needs to be presented efficiently [8]. Health-data visualization-tools can be used to get an overview of a single patient's history, for clinical research, or national studies on population health [100]. In either way, the goal is to gain insights from the clinical data. The visualizations can support decision-making and can reveal patterns in the history of one or multiple patients [8; 89].

To overview a patient's recent past, vital sign records or flow sheets are the traditional paper-based visualization. It is usually a spreadsheet, where the attributes are listed as rows, and its values are recorded over time along the columns. Vital sign records only contain the key attributes of a patient, such as temperature, blood pressure, and pulse, and depict trends and anomalies. Systolic and diastolic blood pressure is often visualized by a graph [93]. Based on the development of the physiological data, intervention needs or treatment effects can be identified [123].

Ordóñez et al. [77] use the *Multivariate Time Series Amalgam (MTSA)* visualization for clinical and physiological data, which is used in Intensive Care Units (ICUs), alternatively to the just described vital sign records. The visualization uses a radar chart (as discussed in Section 2.1) to display multiple attributes over time. The age of a data point is visualized by color, getting darker the more recent an observation is [77].

The *TimeLine* by CompuGroup Medical (CGM) is another electronic version of such a vital sign record. It shows vital parameters, appointments, medication, and further information and users can add and edit data directly from the visualization [132].

For exploration of a whole EHR, *LifeLines* [86; 87] is one of the most mature visualizations and was extended to the applications LifeLines2 [121] and LifeFlow [127], which will be discussed below. To provide an overview of patient records, LifeLines lists events vertically along the horizontal time axis. The events are represented by line segments, normal and abnormal data is color coded. With the compact layout of line segments, many attributes can be visualized simultaneously [93; 123]. Through the alignment on the time axis, one can infer temporal relationships of events [51].

To compare patients and reveal inter-patient patterns, the analysis of sets of patients must be possible. *LifeLines2* [121], which evolved from the single-patient visualization LifeLines [87], can visualize multiple patient records. It also uses a horizontal time axis

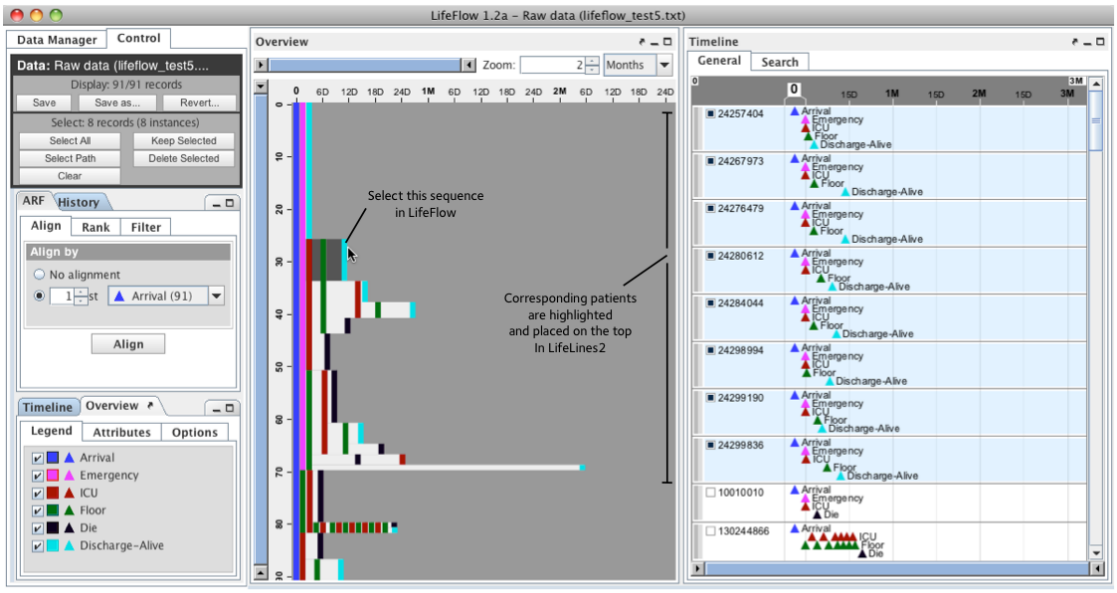


Figure 2.16: LifeFlow combined with LifeLines2 [127]. The center view with LifeFlow shows events of each patient by aligning colored blocks vertically. The color's meaning can be seen in the legend at the bottom left corner. The selected patients are displayed with LifeLines2, in the area in the right third of the image.

on which the events, now from multiple patients, are aligned vertically. LifeLines2 also offers an alignment feature, with which users can align patient records by a specific event to compare their precedent and subsequent events. In histograms, so called temporal summaries, distributions of certain event types, are shown. They can be used in a comparison mode to analyze multiple patient groups [93].

LifeFlow [127], also a development that originates from LifeLines [87], gives users a high-level overview of trends and patterns in millions of patient records [123]. LifeFlow can integrate LifeLines2 as shown Figure 2.16. This allows the exploration of a large number of EHRs as well as the analysis of EHRs from the selected subset [127].

Visualization of Time-Oriented Records (VISITORS) [50] has also evolved from a preceding patient record visualization, in this case from a work by Shahar et al. [97; 98]. VISITORS enhances the previous work to visualize data from multiple patient records [123]. From the raw data, higher level abstractions can be visualized to ease perception. It also adds an extensive query interface to search for specific attribute values, ranges, and combinations or to set temporal constraints [93].

Visualizations that plot patients as data points in a Cartesian coordinate system based on their similarity are *Dynamic Icons (DICON)* [42], *TimeRider* [91], and *Gravi++* [33], for example [93; 123]. By measuring the distances between patients in multiple attributes, similar patients are clustered in Gravi++. TimeRider uses two attributes as axes to create a scatter plot and can vary an item's icon by size, color, and shape depending

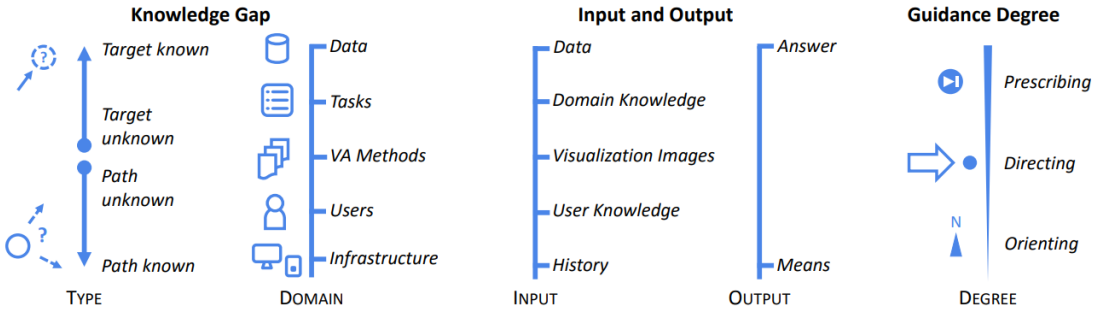


Figure 2.17: Characteristics of guidance in visual analytics by the knowledge gap, the input and output to bridge it, and the degree with which it is overcome [11].

on additional attributes [93]. DICON allows the users to search patients similar to a reference and forms clusters for further exploration [93; 123].

Zillner et al. [133] propose a semantic visualization of clinical data from multiple patients. With the help of semantics, classification of patients can be improved that in turn supports finding similar patients. With the *Semantic Facet Browser*, users can browse the data and find similar patients based on selected attributes.

2.2 User Guidance in Visual Analytics

The techniques discussed in the sections above are applied to analyze complex and large data structures. The sheer number of available attributes and the manifold options that are offered for data analysis can overwhelm users. As the purpose of Visual Analytics (VA) tools is to support the exploration with computing power, methods to guide the user are a helpful addition so that the analysis goal can be reached.

Ceneda et al. [11] define guidance "as a dynamic, iterative, and forward-oriented process that aims to help users in carrying out analytical work using VA methods" [11]. In their work, they extended Wijk's model of visualization [124] with components for the guidance process itself and the inputs and outputs of the guidance process (see Figure 3.2).

Following that model, the main aspects to characterize guidance are: (i) the knowledge gap that stops further exploration, (ii) the inputs and outputs, and (iii) the degree by which users are guided by the VA tool [11]. Figure 2.17 shows these aspects and their possible expressions. The knowledge gap can be divided by the type, i.e., whether users do not know the target of their current analysis, or they do not know the path to reach that target. Why guidance is needed, can also be split by the domain, i.e., whether a user does not know which infrastructure or VA methods she should use, which tasks are needed for successful analysis, which data or parts of data to use or whether someone else should perform the analysis or can support the user. An input from which the guidance process is started can of course come from the data, from the domain in which the analysis is done, or can be based on current visualization images. Users can also

provide inputs to the guidance process, e.g., by giving information about their knowledge, or by the actions the users have made previously. The input can also come from the user, or the users' knowledge derived from feedback by the user, or can be the historical actions of a user.

The output of the guidance process is the answer that narrows the knowledge gap and enables the user to continue the analysis task. This answer can be presented by different means, depending on the degree of guidance. Guidance can merely provide orientation while the users pursue their target. The users can also be directed towards promising paths to their target. With prescribing, the highest degree of guidance, the process takes control of the VA tool and actively continues the analysis automatically.

Shneiderman et al. [100] also highlight guidance in visualizations as a part of healthcare improvement. They suggest to query an EHR database for similar cases to retrieve information on the treatment and outcome and to identify the treating colleagues.

From the tools that were already discussed in this chapter, LineUp [35] helps users to find an unknown target from a dataset through ranking (compare Figure 2.8). While histograms in the column headers give orientation, the ranking of all items by the selected attribute—with the highest ranked items atop—directs users to the most interesting items in the dataset.

As the tabular structure and handling is similar to LineUp, both points also apply for Taggle [25] and Ordino [107]. In addition, to the histograms in the attributes' headers, they also show attribute distributions in the data selection panel. With the summaries of aggregated groups and by switching to the overview mode, further functionalities for orientation are available. LineUp, Taggle, and Ordino operate in the data domain, and use the currently analyzed data as input. Further guidance is added to Ordino with the touring process presented in this thesis.

Streit et al. [106] extended StratomeX [56] with a wizard to define queries to score each stratification. As the wizard has predefined steps and chooses the appropriate queries based on what the user wants to do, this is an example for prescribing actions. The resulting scores on the other hand direct users to attributes that are of interest. This may be the most similar guidance approach to the touring process proposed in this thesis. Already an important feature of the initial StratomeX implementation are the ribbons that reflect the number of shared items in their width and help users to keep the orientation. Besides the data itself, the guidance process utilizes domain knowledge by adjusting the queries based on what the users wants to do.

Domino [36], discussed in Section 2.1.4, is an example for guidance in the VA domain. If users want to add add visualization elements, Domino assists them by showing placeholders and live previews.

Stahnke et al. [102] use a technique similar to *Scented Widgets* [126] with small visualizations at controls to guide the user. While Scented Widgets use history as input and

show other users' activities next to the control to assist novice users [11], Stahnke et al. visualize a preview of a control's effect inside it.

2.3 Similarity Measures

Determining the similarity between two sets of data is a key part to suggest to users with which attributes a further analysis is most promising. In the following, various measures to calculate similarity or distance are presented, split by the type of attribute they are based on.

Given a measure A that outputs the data's relationship as distance $Dist_A$ in the range $[0, X]$, it can be converted to a similarity measure Sim_A by subtracting the calculated distance from the maximum distance as in Equation 2.1.

$$Sim_A = X - Dist_A \tag{2.1}$$

2.3.1 Numerical Similarity Measures

The following section discusses similarity measures for numerical attributes. We will use X and Y to represent numerical attributes, and x and y to represent individual values of these attributes.

If an attribute's values are a function of another attribute, they are usually not independent and correlation can serve as a measure of similarity. Correlation measures the strength and direction of the association of two attributes and can take values between -1 and $+1$, with ± 1 as perfect linear relationship. A correlation of zero indicates no linear relationship. The absolute value of a correlation function can therefore used as a measure of similarity: $Sim(X, Y) = |Corr(X, Y)|$. Table 2.1 shows general rules for interpreting correlation coefficients.

The Pearson correlation coefficient (PCC) can be used if the compared attributes are normally distributed and have a linear relationship. However, the PCC is highly affected by extreme values and inconclusive as soon as one attribute is not normally distributed [69]. Still, for normal distributed data, the PCC is the most frequently used coefficient [13].

$$Corr_{Pearson}(X, Y) = \frac{\mathrm{cov}(X, Y)}{\sigma_X \cdot \sigma_Y} \tag{2.2}$$

Equation 2.2 shows that the PCC is calculated by dividing the covariance of the attributes to be compared by the product of the respective standard deviations.

The Spearman's rank correlation coefficient (SRCC) of attributes X and Y is defined as the PCC of their ranked representations X_r and Y_r (see Equation 2.3) [13; 48]. This means that the SRCC does not use the individual values of measurement pairs, but

Correlation Coefficient	Interpretation
[0, 0.3)	Negligible correlation
[0.3, 0.5)	Minor correlation
[0.5, 0,7)	Medium correlation
[0.7, 0.9)	Major correlation
[0.9, 1.0]	Very high correlation

Table 2.1: Interpreting correlation coefficients [69].

assigns ranks to the values of each attribute and uses pairs of these ranks for comparison. By using the ranked representations, the SRCC can be used in cases where the PCC is not appropriate, e.g., if one of the attributes is skewed or ordinal, and the SRCC is also not affected by extreme values such as the PCC [69]. Thereby, the SRCC can also be used for non-normally distributed data and the relationship between attributes can be non-linear (but still monotonic).

$$Corr_{Spearman}(X_r, Y_r) = corr_{Pearson}(X_r, Y_r) = \frac{\text{cov}(X_r, Y_r)}{\sigma_{X_r} \cdot \sigma_{Y_r}} \tag{2.3}$$

The Kendall rank correlation coefficient (KRCC) is even less sensitive to outliers and interpretation is simpler, as the confidence intervals are more reliable and interpretable [13; 73]. Like the SRCC, the KRCC does not use the attributes' values, but the values' ranks. Equation 2.4 shows the KRCC's definition. Given a pair of value ranks, the following pairs can either be concordant (ordered in the same way) or discordant (ordered differently) [13; 30]. The total number of concordant and discordant pairs are subtracted in the numerator, i.e., a high value in the numerator means that most pairs are concordant and the two attribute rankings are consistent. To normalize the KRCC into a range between -1 and $+1$, the difference of concordant and discordant pairs is divided by the total number of possible rank pairings. For two attributes, X and Y, where each contains n values, the total number of possible combinations is $n(n-1)/2$.

$$Corr_{Kendall}(X, Y) = \frac{|concordant\_pairs| - |discordant\_pairs|}{n(n-1)/2} \tag{2.4}$$

According to Chok [13], the PCC is still superior to the SRCC and the KRCC if the distribution is moderately non-normal as its disadvantages stem mostly from the sensitivity to outliers. The SRCC and the KRCC may also be used for ordinal attributes [69]. The KRCC is also superior to the SRCC if many values are equal and thus share the same rank, as the KRCC uses the order of pairs rather than the ranks [13].

Alternatively to correlation, there are also similarity scores for numerical data, such as *Gower*' similarity coefficient. As it is also applicable for categorical data, Gower's

similarity score is discussed in the next section.
The *Minkowski distance* is a general method to calculate the distance between two points and among the most commonly used measures for numerical data [2; 5]. Equation 2.5 shows how to calculate the Minkowski distance of two Attributes X and Y.

$$Dist_{Minkowski}(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^{\lambda}\right)^{1/\lambda} \tag{2.5}$$

Variables x_i and y_i represent the values of the two attributes and are subtracted. Variable λ specifies the order of the Minkowski distance. For orders of one and two, the Minkowski distance corresponds to the Manhattan and Euclidean distance, respectively [5].

2.3.2 Categorical Similarity Measures

Similarity and dissimilarity measures to assess distances between items for clustering are discussed and compared frequently, as in the work of Vijaymeena and Kavitha [119], Boriah et al. [5], dos Santos and Zárate [18], Alamuri et al. [2], and Šulc and Řezanková [134]. They discuss measures where the categories of the items are compared.

As the touring process, proposed in this thesis, suggests attributes that show similarity to a given item subset, only a selection of methods is discussed in the following. To cluster categorical data based on similarities, probabilistic measures such as the Goodall, Smirnov, and Anderberg similarity coefficients, information theoretic measures like the Lin, Burnaby, and Gambryan similarity coefficients, or frequency based measures as the (Inverse) Occurrence Frequency, and the Eskin similarity coefficient, are some of the available methods [2; 5].

For the following measures, sets of items are represented by A and B and single items by a and b. The number of attributes is n. Items of a categorical attribute have the category, to which the item belongs, as value. The *val* function returns the value of an item in a given attribute, e.g., $val(k, a)$ returns the value of item a in the k-th attribute.

The Szymkiewicz-Simpson or Overlap coefficient is a highly used measure due to its simplicity [6; 119; 134]. As can be seen in Equation 2.6, it is defined as the size of the intersection divided by the size of the smaller set.

$$Sim_{Overlap}(A,B) = \frac{|A \cap B|}{min(|A|, |B|)} \tag{2.6}$$

Hence, if one of the sets is a subset of the other, the Overlap coefficient is one. With the *Overlap Metric*, Stanfill and Waltz [103] use a similar measure, but for dissimilarity.

The Jaccard index or similarity coefficient is one of the most common similarity measures and goes back to 1912 [46; 66]. It is related to the Overlap coefficient, but normalizes

the number of intersecting items by the size of the union.

$$Sim_{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.7}$$

As the Jaccard index neglects true negative (TN) matches—values not present in both sets—Equation 2.7 can also be written as in Equation 2.8, with TP for true positive, FP for false positive, and FN for false negative matches.

$$Sim_{Jaccard} = \frac{TP}{TP + FP + FN} \tag{2.8}$$

Another similarity measures is the Sørensen–Dice coefficient [125]. It is defined in Equation 2.9 and with respect to Jaccard's index, the enumerator is doubled, while the denominator is larger by $|A \cap B|$ [109]. The Sørensen–Dice coefficient together with Jaccard's index are the most common indices [66].

$$Sim_{Dice}(A, B) = \frac{2 \cdot |A \cap B|}{|A| + |B|} \text{ or via matches as: } \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{2.9}$$

A measure that includes the true negative matches is the Rand index (see Equation 2.10). Motivated by the comparison of classifications, similar classified item pairs (TP and TN matches) are counted [120]. If the assignment of an attribute X's items into several categories is treated as a clustering, the Rand index can be used to calculate the similarity between the categorical attribute X and another categorical attribute Y. The Rand index can also be used on sets, as item sets are just subsets of the data (with potentially fewer categories or clusters). Item pairs that are in the same category in X and Y respectively are true positive matches, items pairs that are in different categories in both attributes count as true negative match. Item pairs that are either in the same category in X, but in different categories in Y or vice versa are false negative matches or false positive matches, respectively. The number of true negative matches can be high for large datasets, so that the Rand index may cause problems of comprehension as the true negative matches boost the similarity if comparing attributes or large item sets.

$$Sim_{Rand}(X, Y) = Sim_{Rand}(A, B) = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.10}$$

Gower's similarity coefficient compares two items a and b as defined in Equation 2.11. The Gower similarity coefficient can be used to compare items by multiple attributes and is not only applicable on categorical data, but also for continuous numerical, or binary data [18]. The similarity of two items for the k-th attribute is calculated by $s_k(a, b)$ and w_k denotes the weight of this similarity. The function of $s_k(a, b)$ varies, depending on

the attribute's type (see Table 2.2 and Equations 2.12-2.13). The weight w_k is zero if the comparison is invalid, otherwise one. Comparisons are invalid if an item has no defined value for the attribute or, in the case of binary data, if both items are $false$.

$$Sim_{Gower}(a,b) = \frac{\sum_{k=1}^{n} w_k \cdot s_k(a,b)}{\sum_{k=1}^{n} w_k} \tag{2.11}$$

Table 2.2 shows the values of s_k and w_k for comparison of binary data. As true negative matches are not weighted, Gower's similarity score is identical to the Jaccard index for binary attributes.

$val(k,a)$	$true$	$true$	$false$	$false$
$val(k,b)$	$true$	$false$	$true$	$false$
s_k	1	0	0	0
w_k	1	1	1	0

Table 2.2: Gower's similarity coefficient only counts items as similar if both are $true$. Comparisons where both values are $false$ are ignored and do not contribute to the total weight of Equation 2.11.

Equation 2.12 defines the similarity score s_k between numerical attributes. The absolute difference between the values of a and b is divided by the range of values r_k for the k-th attribute .

$$s_k(a,b) = 1 - \frac{|val(k,a) - val(k,b)|}{r_k} \tag{2.12}$$

For categorical attributes, s_k is one or zero, depending on whether the items' categories match (see Equation 2.13).

$$s_k(a,b) = \begin{cases} 1 & \text{if } val(k,a) = val(k,b) \\ 0 & \text{otherwise} \end{cases} \tag{2.13}$$

2.3.3 Hierarchical Similarity Measures

While numerical and categorical attributes are the majority in the used datasets, the dataset of the Kepler University Hospital also contains hierarchical attributes in form of ICD-10 and ICD-O codes. Hierarchies can be treated as trees of categories and categorical attributes as a hierarchy with only one level. However, the similarity measures discussed for categorical attributes are not appropriate for hierarchies, as they do not consider the

distance between two categories in the tree. This is important for the ICD attributes, as only codes of leaf nodes or parents of leaf nodes are used in practice.

The following measures determine the similarity between two categories inside an hierarchical attribute. Single items are represented by a and b, while sets of items are represented by A and B. Items of a hierarchical attribute have a category as value, like items of categorical attributes. Variables X and Y represent two arbitrary categories of a hierarchical attribute and *root* its root node. The function $p(X, Y)$ returns the number of edges on the shortest path between X and Y. This minimum number of edges that separate two categories can be used as a simple distance metric: $d_{Edges} = p(X, Y)$ [28]. The function $anc(X, Y)$ returns the nearest common ancestor of X and Y. The *depth* function returns the depth of a node in the tree (compare Figure 2.18).

As categories close to the root usually represent higher differences than categories in deeper levels of the hierarchy tree, Wu and Palmer [130] defined a similarity score that takes the depth of the categories into account:

$$s_{Wu}(X, Y) = \frac{2 \cdot depth(anc(X, Y))}{p(X, anc(X, Y)) + p(Y, anc(X, Y)) + 2 \cdot depth(anc(X, Y))} \tag{2.14}$$

The number of edges between the categories' common ancestor and the root node is determined in the numerator by $depth(anc(X, Y))$. The denominator sums up the number of edges between the common ancestor and X, Y, and *root*, respectively. As a result, sibling nodes become more and more similar ,the deeper they are in the tree (compare Figure 2.18 and Table 2.3).

Girardi et al. [28] measure the distance in hierarchies by comparing the number of edges that separates categories, the depth of the categories in the hierarchy, and items with multiple categories in a hierarchy. The distance between two items X and Y in the hierarchical attribute is defined in Equation 2.15 (also compare Figure 2.18 and Table 2.3).

$$d(X, Y) = \frac{p(X, Y)}{depth(X) + depth(Y)} \tag{2.15}$$

The items' distances are calculated for each item in the respective set to every item in the other set, and are weighted by the size of the sets:

$$Dist_{Girardi}(A, B) = \frac{1}{|A \cup B|} \left(\sum_{x \in A \setminus B} \frac{1}{|B|} \sum_{y \in B} d(val(a), val(b)) + \sum_{b \in B \setminus A} \frac{1}{|A|} \sum_{x \in A} d(val(b), val(a)) \right) \tag{2.16}$$

Instead of the distance from Equation 2.15, the similarity measure by Wu and Palmer [130] $s_{Wu}(X, Y)$ can also be used in Equation 2.16. As Equation 2.17 shows, $d(X, Y)$ is just a

transformation of $s_{Wu}(X,Y)$ using the Equation 2.1 from the beginning of this section. The number of edges along the shortest path between two nodes can also be calculated with depth differences, e.g.: $p(X,Y) = depth(X) - depth(anc(X,Y)) + depth(Y) - depth(anc(X,Y))$. The relationship between the two measures can also be seen in Table 2.3.

$$
\begin{aligned}
d(X,Y) &= \frac{p(X,Y)}{depth(X) + depth(Y)} \\
d(X,Y) &= \frac{depth(X) + depth(Y) - 2 \cdot depth(anc(X,Y))}{depth(X) + depth(Y)} \\
d(X,Y) &= 1 - \frac{2 \cdot depth(anc(X,Y))}{depth(X) + depth(Y)} \\
1 - s_{Wu}(X,Y) &= 1 - \frac{2 \cdot depth(anc(X,Y))}{p(X, anc(X,Y)) + p(Y, anc(X,Y)) + 2 \cdot depth(anc(X,Y))} \\
1 - s_{Wu}(X,Y) &= 1 - \frac{2 \cdot depth(anc(X,Y))}{depth(X) + depth(Y) - 2 \cdot depth(anc(X,Y)) + 2 \cdot depth(anc(X,Y))} \\
1 - s_{Wu}(X,Y) &= 1 - \frac{2 \cdot depth(anc(X,Y))}{depth(X) + depth(Y)} = d(X,Y)
\end{aligned}
\tag{2.17}
$$

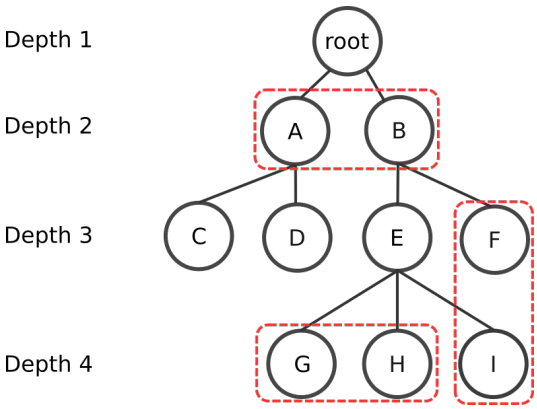


Figure 2.18: Example of a tree with 10 nodes. The nodes circled in red are compared in Table 2.3.

Measure	Score
$s_{Wu}(A,B)$	0.5
$d(A,B)$	0.5
$s_{Wu}(G,H)$	0.75
$d(G,H)$	0.25
$s_{Wu}(F,I)$	$\frac{4}{7}$
$d(F,I)$	$\frac{3}{7}$

Table 2.3: The nodes of the tree in Figure 2.18 are comapred by the similarity and distance measures of Wu and Palmer [130], and Girardi et al. [28], respectively.

2.4 Discretization of Numerical Attributes

In Section 2.3, we have presented various similarity measures. All of these similarity measures have in common, that they only compare attributes or items of the same type. Discretization methods split a numerical attribute into a number of bins, which transforms the numerical attribute into a categorical one. Through the categorization, numerical and categorical attributes can be compared with the similarity measures of Section 2.3.2. The discretization of numerical attributes is a common preprocessing task for machine learning and data mining and helps to homogenize a dataset of categorical and numerical attributes. Maslove et al. [62] evaluated unsupervised and supervised discretization strategies specifically for clinical datasets.

Unsupervised strategies split a numerical attribute into k bins, no matter how large the difference is in the other attributes [19]. The number of bins is chosen beforehand and is selected by the user or the program, e.g., three bins for low, medium, and high, or based on some function like:

$$k = max(1, 2 \cdot log(l)) \tag{2.18}$$

with l being the number of distinct values in the numerical attribute [19].

As the name suggests *equal (interval) width discretization* splits the numerical attribute into k equally sized bins. *Equal frequency discretization* similarly puts an equal number of values into each of the k bins. The third evaluated unsupervised method is *k-means clustering*, which tries to create bins of values by minimizing the distance between the values in these bins. As the output of k-means clustering is dependent on the chosen starting conditions it is a common approach to repeat the clustering multiple times and use the median bin thresholds [62].

Supervised strategies on the other hand use additional information to assign items into bins [19]. Minimum Description Length (MDL) [94] is an unsupervised technique that tries to extract the maximum amount of information while avoiding over-fitting and thus defines k by itself [94; 95]. The numerical attribute is split at points where the entropy gain is highest [64]. The ChiMerge algorithm starts with putting each item into an own bin [62]. Adjacent bins are then merged if their items' class labels are similar [62]. Supervised methods can also use predefined reference ranges to split the data into a given number of bins [62]. As the reference ranges need to be gathered beforehand, this method is not applicable to the general touring process proposed in this thesis.

The above discretization methods are static, meaning that they consider the discretization of one attribute to be independent from the discretization of the other attributes [26]. In contrast, dynamic discretization takes into account the dependencies to other numerical attributes [26]. Gama et al. [26] propose such a dynamic method that treats the possible discretizations of an attribute as a hierarchy. At the top of the hierarchy, all values are in one bin, and at the bottom of the hierarchy, every value is in an own bin. For

the discretization of multiple numerical attributes, all generated hierarchies are used to determine the number of bins for each attribute.

CHAPTER 3

Touring Process Concept

Finding relationships in large datasets can quickly become tedious, as the number of combinations of two attributes from a dataset grows rapidly (compare Equation 1.1). To find relationships in underrated attributes or causes for a present item subset, this chapter describes the design and concept of the touring process proposed by this thesis.

The data used by the developed touring process is composed of attributes. Attributes may be tabularly or hierarchically structured with numerical, textual, or categorical values. The touring process should determine the similarity between sets of items as well as attributes. Users can start the touring process for either, attributes or items, to receive guidance for further analysis. The second input, besides the scope, on which the similarity is calculated, is the similarity measure that will be used by the touring process.

The goal is to have a touring process that:

1. compares data independently of the data type, so that the similarity method does not limit the number of compared attributes.
2. is independent of the data domain.
3. is independent of any VA tool, but also easy to integrate in any desired VA tool.
4. has no impact on the analysis, i.e., users should neither have to stop their work until the touring process has finished, nor should the similarity score computation affect the user's system.
5. is simple to use, to keep the first hurdle of using the touring process low.

We focus on guiding users during the analysis of subsets of the data, i.e., item sets selected by the user. For attribute comparisons, all items must have similar characteristics in both attributes to score high. For item set comparisons however, only the compared data subsets need to be similar to result in a high similarity score.

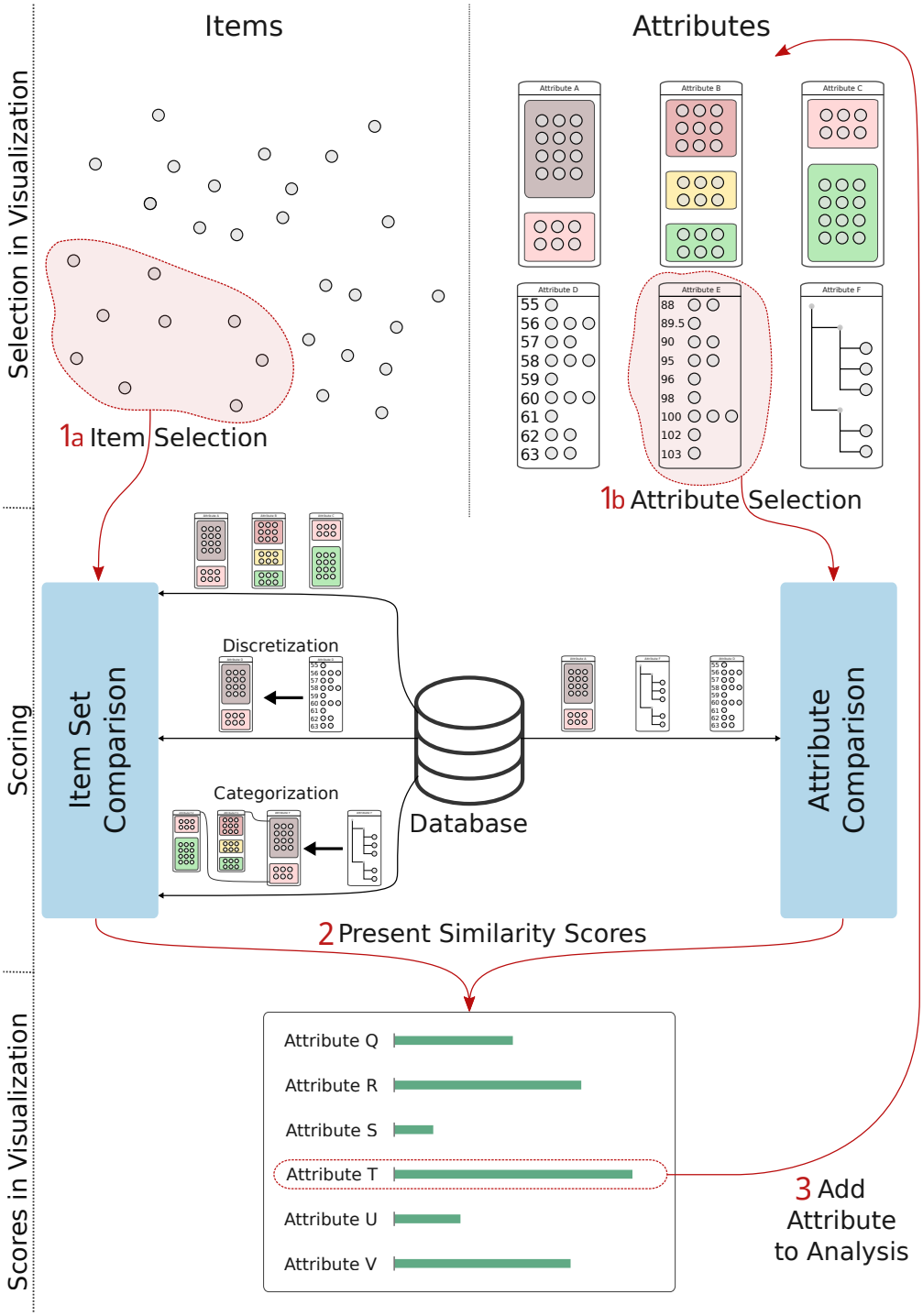


Figure 3.1: Schematic representation of the touring process. After selecting an item set or attribute, scores are calculated. For comparison of item sets, numerical attributes are discretized and hierarchical attributes are converted to categorical attributes. The resulting scores are presented to the user and can be used to continue the analysis.

3.1 Touring Approach

At the starting point of the touring process, users have already loaded one or more attributes of their choice for analysis (top right in Figure 3.1). This could be the metastasis and lymph node stage, two categorical attributes, the patients' age, a numerical attribute, and diagnosis, a hierarchical attribute as the ICD-10 disease classification system is structured hierarchically. From this data in view, a selection on item or attribute level is made (Steps 1a/1b in Figure 3.1) and a similarity measure gets chosen. The selected data subset and the measure are then used to start the comparison task.

For attribute-wise similarity calculation the PCC is implemented to compare numerical attributes (see Section 3.3.1). The server will iterate through all available attributes and calculate the correlation coefficient for every numerical one (middle section of Figure 3.1). For each attribute, for which a similarity score is calculated, the attribute's name together with the score is returned (bottom section of Figure 3.1).

For item-wise similarity, the Jaccard index or two variants of proportions can be calculated (see Section 3.3.2). The server will again iterate through all available attributes (center of Figure 3.1). From those attributes that are already categorical, the chosen similarity score is computed for each category. Hierarchical attributes are treated as a collection of categorical attributes. Numerical attributes are binned so that the resulting similarity score is maximized (see Section 3.3.3). For each attribute, the category with the highest similarity score and that category's name is returned. If the attribute was categorized to compute a similarity score, the thresholds of the bins are returned as well.

After the similarity scores have been returned by the task, the client may query the results and present them to the user (Step 2 in Figure 3.1). The user can use the scores to include attributes of high similarity in the visualizations and proceed with the the analysis (Step 3 in Figure 3.1) .

3.2 Guidance Model

Looking at the model of guided VA [11] in Figure 3.2, data (D) is transformed with visualizations and analytic means (V) into images (I), based on some specification (S). While the data component represents the complete dataset, the specification limits the attributes that are part of the image and defines their representation to the user. The images are perceived by the user, causing a knowledge change (dK/dt). With her knowledge the data is explored by varying the specifications. This can be the addition or removal of attributes, filtering of items or the change of visualizations.

The user's knowledge (K) and the data (D) serve as inputs for our touring process. From her current knowledge, the user has defined a data subset on which guidance is requested. The whole dataset is used for comparison with that subset.

The touring process (G) uses the inputs to provide promising attributes, which the user may or may not use to change the specification (dS/dt). Guidance may be extended by

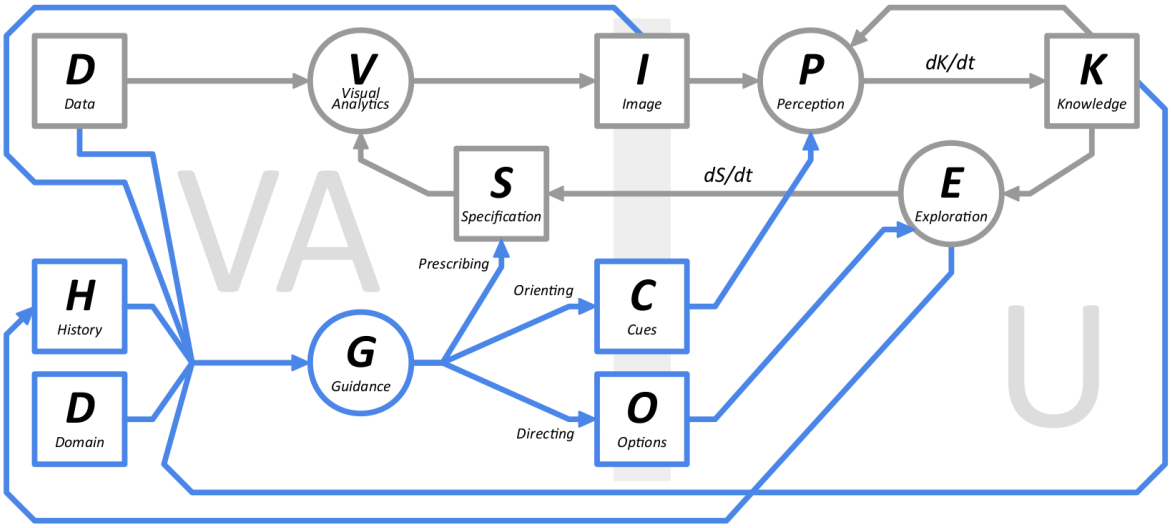


Figure 3.2: Van Wijk's model of visualization extended by guidance components [11]. Components of van Wijk's original model [124] are gray, while guidance components introduced by Ceneda et al. [11] are blue. Functions are represented by circles and transform artifacts, shown as squares.

Type	Domain	Input	Output	Degree
Target Known Path Unknown	Data	Data User knowledge	Similar data represented by scores	Directing

Table 3.1: Guidance characteristics of the proposed touring process.

selecting the appropriate similarity measure based on the data.

Taking the characteristics of guidance (see Figure 2.17) in VA defined by Ceneda et al. [11], the touring process is described in Table 3.1. The target is to identify a subset of items with similar characteristics or to find characteristics that explain the origin of a subset. The touring process presents to the user attributeswith similarity scores so that she can add them into the VA tool and continue the exploration. By changing the input data, the user can ask for further guidance and thus gradually narrow her knowledge gap.

To make the guidance unobtrusive, the integration only consists of buttons to start the touring process and the addition of similarity scores to their respective attributes (see Section 3.4).

3.3 Similarity Measures

The touring process recommends attributes based on the similarity to a user selected data subset. Section 2.3 already discussed various similarity measures from the literature. In this section, we present two additional measures for item sets and describe how the similarity measures are applied in the touring process. Altogether, four similarity measures have been implemented as part of this thesis. Three measures for comparison of item sets and one for the comparison of attributes. The measures described in Section 3.3.1, compare the given attribute to all other attributes applicable for that measure. The data-type-independent comparison was not used for the comparison of attributes. In contrast to items, attributes have clearly assigned data types. As a consequence, attributes would have to be converted, based on the type of the input attribute, which is not feasible (e.g., categorical to numerical).

By one of the measures described in Section 3.3.2, a given item set is compared to each category of categorical attributes. An item set can consist of just a single item or even all items in the dataset. Additionally, the item set similarity measure is used to split numerical attributes such that one of the resulting categories shows the maximum attainable score (see Section 3.3.3). In hierarchical attributes, the hierarchy is converted into categorical attributes (see Section 3.3.4), which are then compared individually.

3.3.1 Attribute Similarity Measures

The touring process uses attribute similarity measures to find attributes similar to a user-selected one. The measures compare the distribution of each attribute's items with the one given by the user. Between categorical attributes, similarity measures can check whether the items are distributed similarly into the categories of each attribute. Between numerical attributes, the correlation can be used to determine the similarity. We have implemented the Pearson correlation coefficient (PCC) for comparison of attributes as described below.

Pearson Correlation Coefficient

The Pearson correlation coefficient (PCC) [69] represents the linear relationship between two numerical attributes X and Y. The coefficient is denoted as ρ if computed for populations, as in the attribute case. The formula can be seen in Equation 3.1. The covariance of the attributes to be compared is divided by their respective standard deviations.

$$Corr_{Pearson}(X, Y) = \rho(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \cdot \sigma_Y} \tag{3.1}$$

The coefficient takes values between -1 and $+1$. A correlation coefficient of zero indicates that there is no linear relationship. Values of -1 and $+1$ indicate a perfect negative or

positive linear relationship, respectively. As the PCC only works on numerical attributes, attributes of other types are skipped for similarity computation.

Due to the sensitivity of the PCC to outliers, item pairs with missing values are omitted as the conversion to any number would have distorted the resulting score. Additionally, in the available dataset the usual case is that if one attribute is not recorded for an item, the others are missing as well. It follows that for an attribute whose items have no data, the PCC can not be determined and no similarity score will be reported.

Despite the disadvantages discussed in Section 2.3, PCC was chosen as it is most frequently used [13], also utilized in medical fields [79], and superior over SRCC and KRCC in moderately non-normal distributions [13].

The coefficient is denoted as r for a sample statistic, i.e., if the PCC is calculated for a subset of the items. This could be an additional method to compare a selection of items (see Chapter 7).

3.3.2 Item Set Similarity Measures

The similarity measures in this section take a set of items A as input and compare this set to the items of every category in every attribute. The items in these categories serve as second item set B for similarity calculation. Both item sets can be of any size.

Together with the item set the user can specify the similarity measure to be used. Though there is a similarity score computed for each of the attribute's categories, only the value of the best scoring category will be returned, together with its name.

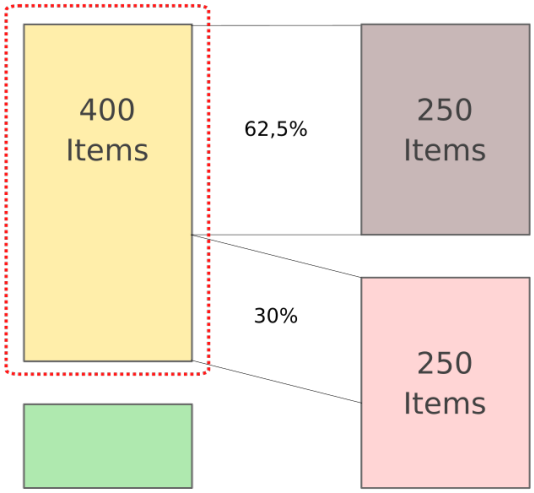
The comparison to all attributes supports users in finding categories where the selected item set is very prominent. For a group of patients that showed similarities in treatment, the comparison against all attributes can be used to find causes for their similar response characteristic.

As this approach requires categorized attributes, numerical ones are categorized on-the-fly so that the resulting categories' score is maximized (see Section 3.3.3). Hierarchical attributes are treated as a collection of categorical attributes (see Section 3.3.4).

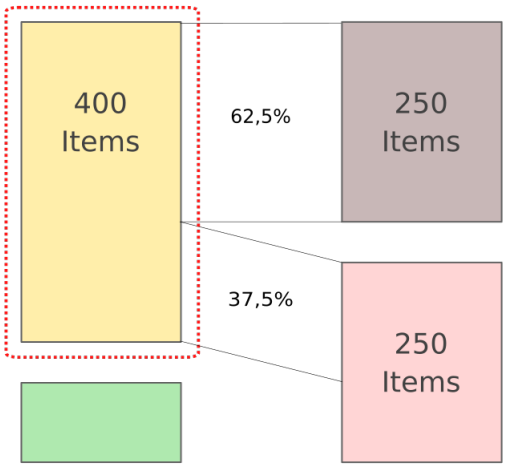
The Jaccard index was implemented as first similarity measure and is discussed in the next section. Additionally two versions of proportions can be calculated. All three ignore true negative (TN) matches in item sets, thus items that are part of the respective attributes but not of the compared item sets. This can clearly be seen in Figure 3.3.

Jaccard Index

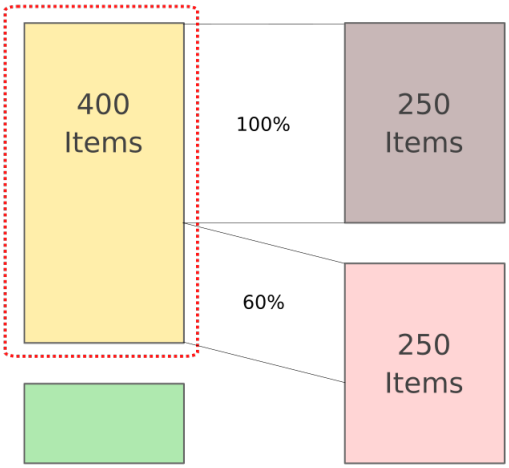
The Jaccard index [46] is a measure to calculate the similarity of two sets. It is defined as the size of the intersection, divided by the size of the union of the item sets (compare Equation 3.2 and Figure 3.3a). Consequently it has a range of 0 % (no common items in the sets) to 100 % (the sets share all items).



(a) Visual representation of Jaccard index calculation. The number of items present in both sets is divided by the number of total items, which is 400 for the upper category, but 500 for the lower category.



(b) Visual representation of the *Shared/Selected* proportion. The number of shared items is divided by the number of items in the user-selected set (400).



(c) Visual representation of the *Shared/Compared* proportion. The number of shared items is divided by the number of items in the compared set (250 for each category).

Figure 3.3: Visual representation of the implemented methods to calculate similarity between two sets of items. On the left side is a attribute with two categories, of which one has 400 items. This category is also the selected item set (red border). On the right is an attribute with two categories, each containing 250 items. Different colors are assigned to the categories. While the upper category of the right attribute shares all 250 items with the selected set, the lower category only shares 150 items. The similarity scores between the selected items and the two categories are shown for each method. True negative (TN) matches, i.e., items that are not part of the compared sets, e.g., the green category, are ignored by all three similarity measures.

$$Sim_{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{3.2}$$

The Jaccard index is a commonly used method to compare two sets as the number of identical items is normalized by the total number of items in both sets [28]. Thus, the selection of the user has to fit the comparison set very well to achieve a high similarity score.

Proportions

As the Jaccard index is based on the union of the item sets, but the user triggers the similarity calculation only with one of them, simple proportions have been added for easy understanding of relationships. The advantage of these simple methods is that a large item set, be it the selected or the one to be compared to, is not penalized like with the Jaccard index. Rather than the Jaccard index, these two methods are more related to the Overlap coefficient, discussed in Section 2.3, where the size of the sets' intersection is divided by size of the smaller set.

The first method is named **Shared/Selected** and simply shows how many of the items in the comparison set are also in the item set selected by the user (compare Equation 3.3 and Figure 3.3b).

$$Sim_{SS}(A, B) = \frac{|A \cap B|}{|A|} \tag{3.3}$$

If the user selected 400 items, of which 150 are in the comparison set together with 100 others, it will score $\frac{150}{400} = 37.5$ %. The advantage of this simple approach is that users can easily see if a large portion of the chosen items are part of another category. The disadvantage of this method is of course that small selections easily achieve high similarity scores. The fewer items are selected, the more likely it is that these items fall into the same categories in other attributes, most notably if those other attributes have few categories.

Shared/Compared is the second method and works vice versa by calculating the percentage of the user-selected items that are also part of the second item set (compare Equation 3.4 and Figure 3.3c).

$$Sim_{SC}(A, B) = \frac{|A \cap B|}{|B|} \tag{3.4}$$

If the user selects 80 items, of which 50 are in the comparison set together with 10 others, it will score $\frac{50}{60} = 83.33$ %. The advantage of this simple approach is that users can easily

see if subsets of the chosen items are a large portion of another category, even if the selected set is much larger then the compared set. The approach also causes that the items of small categories of other attributes may quickly be part of the selected set and score very high.

3.3.3 Discretization of Numerical Attributes

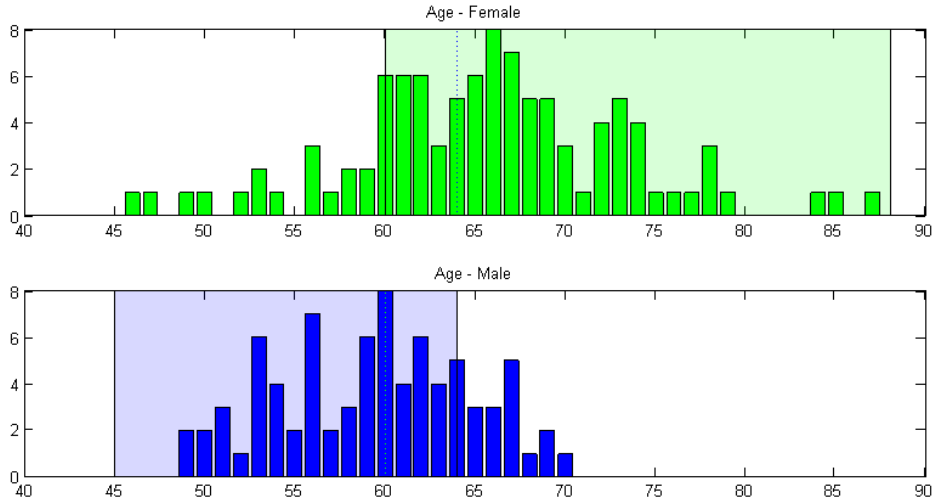
The similarity measures described above compare subsets of items. As numerical attributes have no categories—although there can be categorized attributes with numerical values—calculation of similarity scores would not be possible. Therefore the numerical data has to be split in bins, to use the resulting categories in the similarity measures already discussed in Section 3.3.2. In this manner, the numerical attributes integrate seamlessly into the touring process and can easily be compared to the categorical attributes.

If a numerical attribute is compared to an item set a, the touring process tries to categorize it by finding the best value to split the attribute into two bins. Items that have no value for that attribute are treated as a separate set and are excluded from the following operations. To find the categorization, the values of the attribute are first sorted in ascending order. Next, for each distinct value in the numerical attribute a set of items with values less or equal and a set of values greater or equal is formed. Each of these sets is compared to a. The highest scoring set determines the value by which the attribute is split into two disjoint sets. With the set of items with no value, the attribute consists of three sets after discretization which will serve as categories. As the discretization is dependent on the given item set, this is a supervised discretization method (compare Section 2.4).

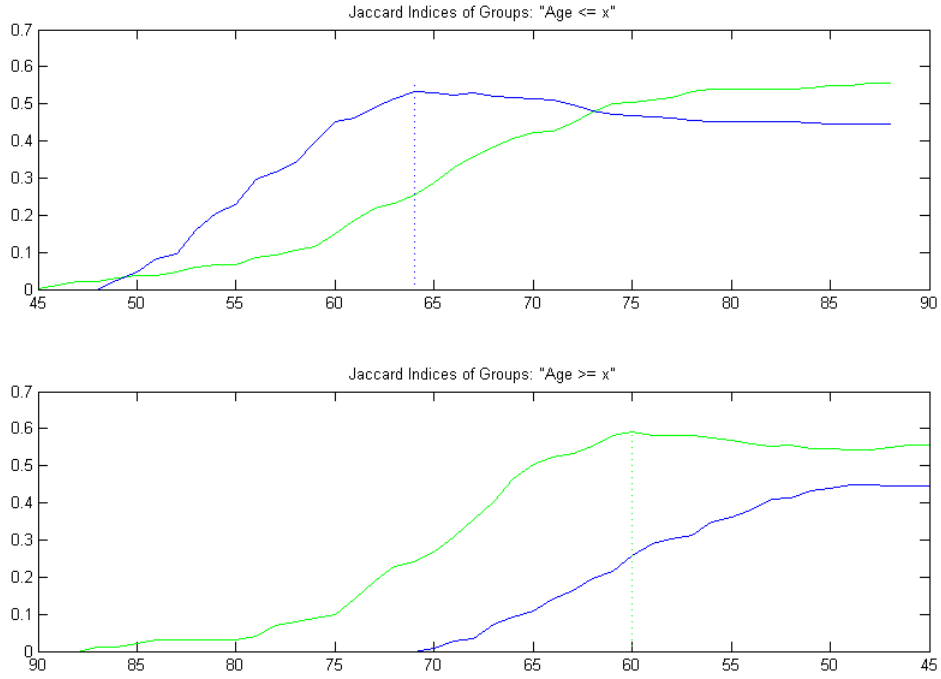
An example of the discretization of the numerical attribute *Age* is shown in Figure 3.4. Figure 3.4a shows the categorical attribute *Sex*, whose categories *female* and *male* have their items' *Age* values shown in histograms. Figure 3.4b shows the Jaccard indices for sets below and above a certain age. If compared with the *male* category, items of $age \leq 64$ form a set which is between the mean of the two *Sex* categories (compare Figure 3.4a). As the *female* category has a higher mean, the set with items over a specific value scores higher (compare Figure 3.4b). It can be seen that the curves that do not reach the maximum similarity score are simply rising towards the proportion of the sets total item number (here 100:80, compare Figure 3.4b). The MATLAB [63] script used for the binning in Figure 3.4 is provided in Listing A.1.

If there are significantly more items of the numerical attribute in the compared category than in other categories of the categorical attribute, the highest Jaccard index may be achieved by creating a set ranging from the minimum to the maximum value. Another discretization approach would be to maximize entropy, as discussed in Section 2.4, and calculate similarity afterwards.

The only requirement of this discretization method is that the items of the attribute have to be sortable. As the values of the numerical attributes are only used for sorting, this



(a) Histogram of the *Age* attribute. If compared to the category *Female*, the numerical attribute is split into a set of $age \leq 59$ and a set of $age > 59$, indicated by a light green background. If compared to the *male* category, sets of $age \leq 64$ (indicated by a light blue background.) and $age > 64$ are created.



(b) The curves shows the Jaccard indices of the sets $Age \leq x$ and $Age \geq x$ with the categories of the attribute *Sex*. The green line indicates comparison with the *female* category. It reaches the maximum of 0.592 at age 60. The blue line shows comparison with *male* items and reaches a maximum of 0.533 at age 64.

Figure 3.4: The numerical attribute *Age* is compared with the categorical attribute *Sex*. 100 of the items in *Sex* are normally distributed in the *female* category with $\mu = 66$ and $\sigma = 8$. In the *male* category 80 items are normally distributed with $\mu = 60$ and $\sigma = 5$.

approach is not limited to numerical attributes, but can also be applied to attributes of other types.

3.3.4 Similarity with Hierarchical Attributes

A hierarchical attribute is a special form of a categorical attribute as each category may contain further subcategories. Due to the nested categories, hierarchical attributes form a tree. Items are leaf nodes and define the finest level of categories implicitly by their value (compare Figure 3.5a). An example of such a hierarchy is the ICD-10 disease classification system.

For item set similarity scoring, the tree is converted to multiple categorical attributes. An additional category containing the items in the rest of the tree can be added as well. For each internal node that has items in its subtree, a categorical attribute is created that has one category for each child node (compare Figure 3.5). In this way, the similarity measures for categorical attributes can be used to compare a item set with subtrees of a hierarchical attribute.

The attribute's name serves as root node. Each node contains a list of child nodes as well as a flag to indicate whether item nodes of the tree that are not part of the subtree should form an additional category. A label for the resulting categorical attribute is also defined for each node. Details on how this is configured are discussed in Section 4.2.2.

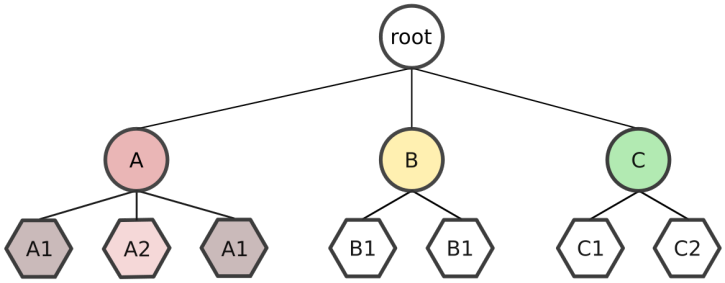
3.4 Visual Integration

Based on the guidance model in Section 3.2, the visual integration of the touring process requires the following three components:

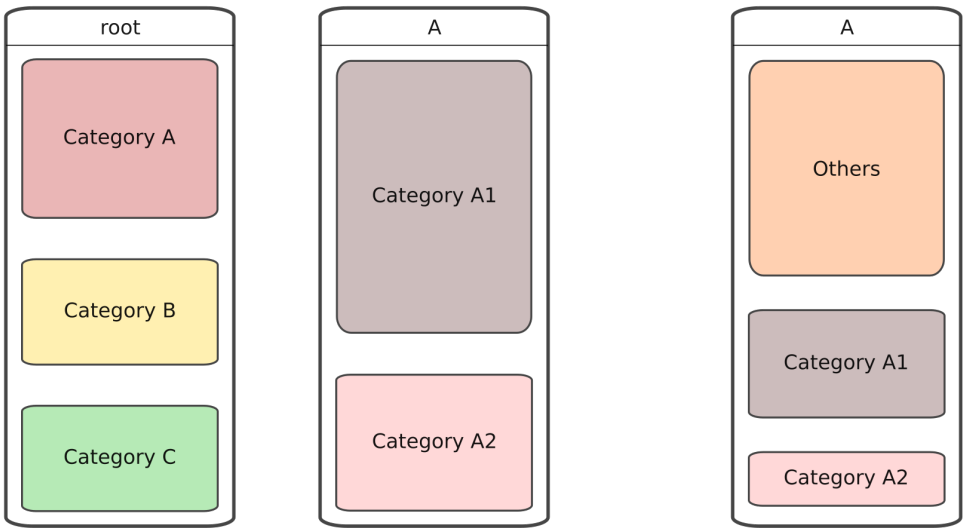
- A way to select a data subset and similarity measure as inputs.
- A control to start the touring process for this subset.
- Visualization of the resulting scores for each attribute.

The user guidance should extend the VA system in an unobtrusive and adaptive way, preferably using user interface elements and interactions that are already part of the system [11]. The introduction of additional views, methods of data selection or alike might otherwise have the opposite of the intended effect.

Users need to be able to select a data subset as input that will be compared to the whole dataset by the touring process. The data subset may either be an attribute or a set of items. Using a dataset that contains n items, the number of selected items can be in the range of $[1, n]$, though selections of items that are part of a category, or a combination of categories, will be the normal case.



(a) A hierarchy with seven items represented as tree. The root node and its three child nodes will each be turned into a categorical attribute (see below). The hexagonal nodes represent the individual items of the hierarchical attribute. The item's value, which is also its subcategory, is shown inside the nodes.



(b) Examples of categorical attributes formed from the hierarchy above. On the left side is the categorical attribute created from the root node, with a category for every child node. In the middle is the categorical attribute of node A. The items in this subtree fall into two categories (*A1* and *A2*). The categorical attribute on the right also contains the subtree of node A, but includes an additional category *Others* for items that are not part of the subtree. This category is significantly larger, as there are more items in the remainder of the tree than in the subtree of node A.

Figure 3.5: A hierarchical attribute represented by a tree is converted into multiple categorical attributes. Colors are assigned to the categories to highlight the transformation from the tree in a) to the attributes in b).

An appropriate similarity measure for the selected data is the second input needed by the touring process. The measure may be chosen by the VA tool, depending on the data, e.g., correlation for numerical attributes.
After selecting the data, for which guidance is needed, and a similarity measure the touring process gets started manually or automatically. Users are visually informed of the touring process' start and its current state.

Completion of the server-side data comparison is visualized by this information and the similarity scores that are presented for each attribute. In tools where attributes can dynamically be included in the analysis, the scores are best presented directly where attributes are added. Ranking the attributes by their similarity scores can be used to further guide users to those that are most relevant for the current analysis.

Representation of the similarity scores is suggested by bars (see Figure 3.1). The visualization by region is independent of the measure's value range. Users do not have to know the range of a score themselves and see differences at a glance. Knowledge of the exact scores can still be required and should be displayed on demand. For item sets, the attribute's category that achieved the displayed score shall be visible as well.

Users may start the touring process multiple times, differing in the selected data subset or similarity measure. The subset for which guidance was requested and the utilized similarity measure has to be shown if displaying multiple results.

A real world usage example for the touring process would be to find reasons why patients had no chemotherapy, even though their clinical parameters suggest the treatment. To do so, the user selects all items (patients) that either have lymph nodes metastasis or have a positive HER2 status[1], but did not receive a chemotherapy. In addition to the subset, the user has to choose a similarity measure. As she wants to find categories that differentiate the selected items, the presence of additional items in the compared categories should be penalized. Therefore, she will use the Jaccard index for the comparison of item sets in this example. With the items set and similarity measure specified, the touring process will start the comparison with all attributes. A common contraindication to a chemotherapy is the age of patient, so a we expect the touring process to discretize the numerical attribute *Age* such that the resulting categorical attribute has patients of high age in one category. Patients with multiple comorbidities are also often excluded from a chemotherapy treatment due to the side effects. The touring process should help the user to find additional contraindications that led to the treatment not being carried out. Similar examples, their results and visual representation of the similarity scores are shown in Chapter 6.

Users have to keep in mind though that repetitive queries for similar attributes may return significant results by chance, as the more comparisons are made, the more likely

[1]Human epidermal growth factor receptor 2 (HER2) is a protein, that controls breast cell growth. An amplified HER2 production thus leads to uncontrolled cell growth. The HER2 status indicates whether the process plays a role in the cancer.

it is that some difference is found. Due to this multiple testing problem, the relationship of attributes reported as similar has to be validated.

CHAPTER 4

Implementation of the Touring Process

The implementation of the touring process is split into two parts: it starts on the client-side, with a selection set by the user and their query for guidance. The main processing then happens on the server-side, where the available data is iterated, compared, and even categorized. The resulting similarity scores are reported back to the client-side which presents them to the user.

Both parts of the implementation are based on the Caleydo's Phovea Platform. The platform uses Python [88] for server-side implementations and Typescript [117] on the client-side, usually web applications running in the users' browsers.

4.1 Caleydo Phovea Platform

Caleydo is a data visualization project with focus on biomolecular data. The Caleydo platform is a framework on which the initial Java implementation of StratomeX, described in Chapter 2, was built. The Phovea platform is the web-based successor of the Caleydo platform. It is also intended for visual analysis, with a client-server architecture and focus on biomedical applications. For both client- and server-side, the platform provides support to quickly start interacting with the available data. As a web-based platform, Phovea was initially referred to as Caleydo Web [37; 38] before a recent name change [34].

The Phovea server [85] and server-side components are written in Python, while client-side components are written in Typescript, a typed superset of JavaScript that compiles to plain JavaScript understood by browsers.

The Phovea platform includes a plugin mechanism to enhance its functionality and defines classes for matrices, tabular and categorical data. Plugins may add support for datasets

that are stored in Hierarchical Data Format (HDF) [80], a Mongo[81] or Structured Query Language (SQL) [82] database, or add services like a task queue [83] (see Figure 4.1). Technically, web applications such as StratomeX and Ordino are plugins as well. The Caleydo website [9] lists further applications.

The HDF plugin is used to read TCGA KIRC [111] and GBM [110] data from an HDF5 file. Both contain clinical and genomic data with categorical and numerical attributes. As the data is directly read from the file, there is no separate database process as in Figure 4.1.
The Kepler University Hospital's tumor dataset is stored in a SQL database, but accessed via a REST API. The dataset contains clinical and genomic data, in categorical and numerical, but also hierarchical form.
Phovea's processing queue plugin is used for the server-side task to calculate the similarity, discussed in detail in the next section.

The Phovea platform also provides ID management and mapping. Each element in the data is assigned an additional unique integer, which will be used as ID within the Phovea platform. These IDs may be mapped to translate between annotation systems. As numbers they are also easily compared and sets of IDs can be expressed by ranges. These unique numerical IDs are also used for item selections.

4.2 Server-Side Similarity Scoring

The server-side is implemented in Python, a common programming language among data scientists. The Phovea plugin *phovea_processing_queue* utilizes Celery [10] to add background task processing and deferred execution functionality (see Figure 4.1). Celery can execute tasks immediately or on schedule and supports asynchronous and synchronous execution. One or more workers can be defined to handle the tasks. For the touring process, asynchronous tasks are started immediately on demand.

For this thesis, we created the new Phovea plugin *phovea_processing_similarity* (see Figure 4.1). It extends Phovea's task queue plugin with similarity measures for attributes and sets of items. The plugin is publicly available on GitHub [84].

Figure 4.1 gives an architectural overview of the touring process components. We have executed each process in an own Docker [16] container for better reproducibility. The Phovea server and the Celery container use the Docker Debian image defined in the Phovea server repository [85]. The Kepler University Hospital's data is stored in a SQL database, therefore we used the MariaDB [17] Docker image.

Figure 4.1: Architectural overview of the touring process. On the server-side, the Phovea server, the database, and Celery run in their own process. The picture uses a generic database and database plugin, as we have used SQL and HDF data in the touring process. The similarity plugin extends the processing queue plugin with similarity measures, calculated in separate tasks with Celery. Celery loads all attributes from the Phovea server for comparison. The client-side web application only calls the API of the Phovea Server.

Similarity Measure	Scope	Identifier
Jaccard index	Item sets	`jaccard`
Shared/Selected	Item sets	`shared-selected`
Shared/Compared	Item sets	`shared-compared`
Pearson correlation coefficient	Attributes	`pearson`

Table 4.1: Identifiers of the implemented similarity measures.

4.2.1 API

When the Phovea server adds the plugin, it registers a namespace *similarity* in which the plugin can define its own routes. These routes can be used to listen for HTTP requests on certain URLs. The Phovea server uses the Flask [21] framework to create this API. The plugin defines two routes to start the calculation with the implemented similarity measures. One for similarity measures between sets of items that takes the desired similarity measure and the items' IDs as parameters, and one for similarity measures between attributes that takes the attribute's ID as well as the desired similarity measure as parameter. The differentiation between attributes and item sets allows the touring process to use one similarity measure for both types.

The route for similarity measures between item sets is defined as `<server-address>:<port>/api/similarity/group/<method>?range=<selection>`, with similarity being the namespace of the plugin and *method* an identifier for one of the measures described in Section 3.3.1. The identifiers can be seen in Table 4.1. The IDs of the items to be compared are passed by a range parameter. The selection handling of the Phovea platform takes care of translating the passed ID ranges to a full list of the individual IDs. An example would be: `http://localhost:9000/api/similarity/group/jaccard/?range=(350:357,1072:1082,1084,1202,1553)` which passes a total of 22 IDs.

The similarity measures between attributes take the selected attribute's name as parameter. It is defined as `<server-address>:<port>/api/similarity/attribute/<method>/<attribute_id>` with *method* being an identifier for attribute similarity measures (see Table 4.1 and Section 3.3.1) and the *attribute id* to identify and retrieve the attribute from the dataset. To calculate the Pearson correlation coefficients for the attribute *Days to Death* in the TCGA KIRC dataset the task could be started with: `http://localhost:9000/api/similarity/attribute/pearson/tcgaKircClinical_patient.daystodeath`.

With these parameters the asynchronous Celery tasks, in which the data is processed and scores are calculated, are created. As mentioned above, the tasks run asynchronously and start immediately. Celery creates an ID for every task upon creation. With this ID the results can be retrieved from `<server-address>:<port>/api/processing/res/<ID>`. Furthermore an event stream displays the status of all tasks. The event stream's route is `<server-address>:<port>/api/processing/stream`. The routes to

query the results and status differ from those of the similarity measures, as they are defined in the *phovea_processing_queue* plugin.

4.2.2 Similarity Measures

The similarity measures discussed in Section 3.3 are each implemented in an own class. All similarity measures classes derive from the common abstract base class *ASimilarityMeasure* shown in Listing 4.1.

Listing 4.1: Definition of class ASimilarityMeasure.

```
class ASimilarityMeasure(object):
  __metaclass__ = abc.ABCMeta

  @abc.abstractmethod
  def __call__(self, set_a, set_b):
    pass

  @staticmethod
  @abc.abstractmethod
  def is_more_similar(measure_a, measure_b):
    return measure_a > measure_b

  @staticmethod
  @abc.abstractmethod
  def matches(name):
    return False
```

At runtime, all subclasses of *ASimilarityMeasure* are retrieved. Each subclass is compared to the declared similarity measure parameter with the `matches` function In this way, it is sufficient to implement a new subclass, to add a similarity measure to the plugin and use it via the API.

The `is_more_similar` function prefers higher scores by default, but can be overridden to support either distance measures or negative values as in the case of PCC, where the absolute values are compared.

Item Set Similarity

Similarity measures between sets of items, treat the given item IDs as first set. For every attribute, each categories' item IDs are compared to this given ID set. For the following example, let's assume there is a fictional dataset of cities in Austria that contains the categorical attributes *State* and *Is state capital?*, with nine and two (yes/no) categories respectively. The user selected item set U is first compared to each of the categories in attribute *State*, and then to the two categories of attribute *Is state capital?*. The more

items of a category have the same ID as the items from U, the higher is the similarity score of this category.

The data structures of Phovea can export their data as NumPy [76] arrays. NumPy is a Python library that adds support for large arrays and matrices to the language and offers a wide range of functions for them.

For **categorical attributes**, the values of the exported NumPy array are the categories of the individual items, and an item's ID can be deduced by its index in the array. When iterating over all attributes, the indices of each category's items are retrieved. With NumPy's set operations for one-dimensional arrays, *intersect1d* [74] and *union1d* [75], and the arrays' sizes, the scores of Section 3.3.2 are calculated.
Hierarchical attributes that are split into categorical ones (see Section 4.2.2) are treated equally.

The result of the similarity task is stored in a Python dictionary and contains the highest scoring category's name of each attribute together with its score and, if it is a numerical attribute, the threshold to categorize it. When the dictionary is returned over the REST API it is converted to JavaScript Object Notation (JSON) with Flask [22].

After all categorical and hierarchical attributes have been processed, the **numerical attributes** are handled. The NumPy array of a numerical attribute contains the numerical values of the items. As the attribute's values will be split into groups below and above a certain threshold, the values need to be sorted in ascending order. The change of order would prevent the deduction of an item's ID, so a two-dimensional matrix with the item IDs in the first column and the items' values in the second column is formed. The matrix is then ordered by the numerical values in the second column. Two additional empty columns, which will contain the similarity scores, are added afterwards. One column for the similarity score between the given IDs and the IDs with values less or equal the respective value in column two, and one column for the score between the given IDs and the IDs of values greater or equal the value in column two. After each distinct numerical value in the matrix's second column is used to form and score these two groups, the maximum score is retrieved. The corresponding numerical value will be used to split the attribute's values in two groups. For reasons of consistency, the value the attribute will always be split to form a category with values less or equal, and a category with values greater than the threshold. This means that if the highest similarity score is reached by a group greater or equal a certain value, the next lower value in the dataset will be returned. The threshold is returned inside an array to support future discretization methods that create more bins.

Attribute Similarity

For similarity measures between attributes, the values in the given attribute are compared with the values in every other attribute. Attributes whose type is invalid for the comparison are excluded, e.g., categorical attributes are excluded from the comparisons with the Pearson correlation coefficients. An attribute's values are retrieved as one-

dimensional NumPy array just as described above. For every attribute that is compared by the similarity measure, its value array and that of the given attribute are input to the similarity measure.

With Scipy [47], a library for scientific computing that builds on NumPy, the PCC of the two arrays is calculated. Measurement pairs, where one of the attributes has no value, are omitted from the Pearson correlation coefficient calculation due to PCCs sensitivity to outliers and data distribution. The score of each attribute is stored in a Python dictionary together with the attribute's ID and is converted to JSON as well.

Splitting of Hierarchical Attributes

As already described in Section 3.3.4, hierarchical attributes are split into categorical attributes. Each subtree of the hierarchical attribute becomes a categorical attribute, with one category for each child node and an additional optional category for items in the remainder of the whole tree.

The tree's structure is defined in a JSON configuration file underneath the `hierarchies` key. Listing 4.2 shows the basic structure. A configuration excerpt for the ICD-O Morphology ontology can be seen in Listing A.2.

For each hierarchy that should be converted into multiple categorical attributes, the hierarchical attribute's name is included in the configuration and serves as key for an object that further describes the tree's structure (`Hierarchical_Attribute` in Listing 4.2). This object has the same structure for the tree's root and any child nodes. It contains a label that is used if that node is turned into a category. It may contain a list of child nodes (`children`) that will also become categorical attributes and a flag that determines whether to add another category for items in the rest of the tree (`includeOthers`). If `includeOthers` is not defined, the value of the nearest parent node is used. The default is `false`. The JSON object further contains an array of subcategories that are the tree's leaf nodes (`startWith`). These will not become categorical attributes (compare Figure 3.5).

Listing 4.2: JSON Configuration for hierarchical attribute categorization.

```
"hierarchies": {
  "Hierarchical_Attribute": {
    "includeOthers": true/false
    "children": {
      "ChildNode1": {
        "includeOthers": true/false,
        "label": "Label for categorical attribute",
        "startWith": []
      },
```

```
            "ChildNode2": {
              "includeOthers": true/false,
              "label": "Label for categorical attribute",
              "startWith": []
            }
          }
        }
      }
```

CHAPTER 5

Integration of the Touring Process

We demonstrate the touring process in Ordino and StratomeX, two web applications that are plugins of the Phovea Platform. The client-side additions in these two applications were implemented in Typescript and make use of the tools provided by the Caleydo Phovea Platform.

To start the touring process, users either select items or attributes and specify a similarity method. As already described in Section 4.1, every item has a numerical ID. While a selected attribute is passed to the backend by its ID, item selections are represented by ID ranges. The similarity method is specified by a string keyword, e.g., *jaccard* or *pearson* for the respective scores (see Table 4.1). This information is then passed to the server-side component discussed in Chapter 4. The start of the touring process over the REST API returns a Promise [68], which is fulfilled upon task completion.

While the similarity scores are calculated, users can continue their analysis and are informed of the ongoing server-side processing. When the server-side processing is done the results are retrieved automatically, by the Promise, and visualized.

The attributes are shown together with their own or their categories' similarity score. The user can then continue the analysis by adding new attributes or a change of selection for another touring iteration.

While the touring process offers to start from item and attribute selections, only the implementation in Ordino provides both options. In the StratomeX implementation users can only start the touring process with an item set as input.

5.1 StratomeX

StratomeX was reimplemented as part of the Caleydo Phovea Platform by Gratzl et al. [37] and has its user interface split in the following three parts: the main area where the data is displayed, a data browser, and a provenance graph (compare Figure 5.1). The provenance graph is part of Capture, Label, Understand, Explain (CLUE) [38], which records a user's actions during exploration and can be used to share and present findings together with their derivation. The data browser is a LineUp [35] table in which each attribute is displayed as a row and columns relate to properties of the attributes. With a plus symbol on the left, attributes can be added to the main view for analysis (see Figure 5.1).

The attributes in the main view are displayed as columns, built by individual bricks that relate to the attribute's categories. Categories of adjacent attributes that share subsets of items are connected by ribbons. The ribbons width is proportional to the number of items in the shared item subset and the total number of items in the category (compare Figures 5.1 and 2.15).

The first step of the touring process is done in the main area by selecting a set of items. This can either be done by clicking on an attribute's category or on a ribbon that connects categories. With the selection of ribbons, the user is able to start the touring process from a subset of a category's items.

After the items are selected, the user chooses the similarity measure to be used from a drop-down menu shown by the *Calc Similarity* button in the data browser (bottom left in Figure 5.1). By selecting the similarity measure the server-side task is started and two new columns are added to the data browsers. These columns are initially empty and indicate the pending data with spinning wheels and the text *Loading...*. While the server-side task is running, users can continue the analysis, change selections or the similarity measure, and start further tasks, which will add additional columns every time.

As soon as the server-side task has finished, the results are loaded in the corresponding columns of the data browser. The first column displays the score of the highest scoring category as a bar and the exact numerical value if users place their mouse over the table cell. In the second column the attribute's category that achieved this score is printed. In this way, interesting attributes and categories can be identified and included in the analysis while attributes that score high in an irrelevant category can be seen at a glance. Subsequent calculations will add additional columns in the data browser.

The table's rows are also sorted in descending order by the score to quickly see the most similar attributes and their highest scoring categories.

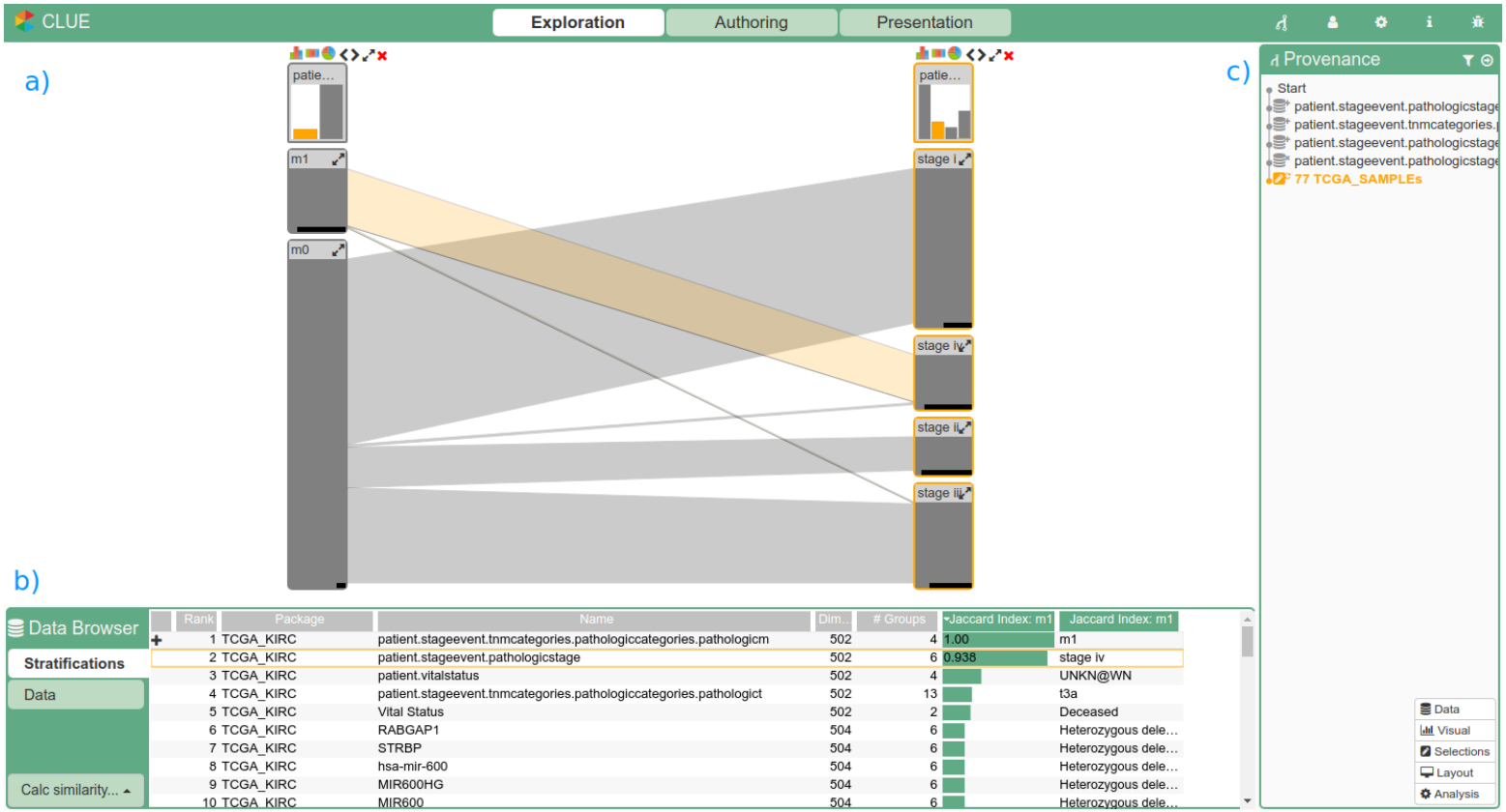


Figure 5.1: The StratomeX user interface is split in three parts: the main area *a)* where the data is displayed, a data browser *b)*, and a provenance graph *c)*. The data browser's table displays a + symbol to add the attribute to the analysis, if the mouse is placed over a row. The data browser also contains the button to start the touring process *Calc similarity...* in the bottom left corner.

5.2 Ordino

Ordino's approach is to provide users with the option to analyze the data on item level, i.e., in detail mode, and on attribute level, with the overview mode (compare Section 2.1.7). The data is shown in tabular form, where each column corresponds to an attribute and the rows to the items (see Figure 5.2). Each table cell contains the value of an item for the specific attribute.

Groups of items, e.g., categories, can be aggregated. In overview mode, items are decreased down to a minimum size of 1 px. Selected items are displayed in their full height (see Figure 5.2). Selections of items can be made by brushing in overview and detail mode. In detail mode, the user can also select items by clicking on a row or via checkboxes. The items in aggregated groups can also be selected at once.

Additional attributes can be added from a searchable drop-down menu. Right above this drop-down menu, a button to start the touring process is added adjacent to existing ones for data export, zooming, and saving the current list of items (see Figure 5.3). As these buttons have no label text but only use icons, an *align-left* button has been chosen as it resembles the similarity score bars that will be the result of the touring process (see Figure 5.3a). If a selection exists, the button above the drop-down menu can be clicked to start the server-side process with the Jaccard index as similarity measure. While a server-side task for similarity calculation is active, this icon will change to a spinning wheel (see Figure 5.3b). Both icons are part of the *Font Awesome* [23] icon set.

Attribute's can not be selected per se in Ordino. To start the touring process with an attribute rather than an item set, a button *Find similar...* has been added to the list of actions in the attributes' header (see Figure 5.4). As currently only the PCC is implemented as attribute-based similarity score (see Section 3.3.1), the button is added to numerical columns. By clicking the button, the similarity measure and the attribute's name is sent to the backend and the task is started. The button, which is used to start the task for item selections, will change to a spinning wheel to indicate the work by the new task, as this is the only constantly visible guidance component. While the server-side task is running, users can continue the analysis, change selection or the similarity measure, and start further tasks, just like in StratomeX.

When the task is finished, the button will change from the animated spinning wheel back to the default icon (see Figure 5.3). The similarity scores are shown as bars beneath each attribute in the drop-down menu (compare Figure 6.4). Bars have been used because their length can be compared quicker than the similarity score as text. They have a light green background that indicates their maximum size. If hovering over an attribute, the attribute's category that achieved this score and its exact value are shown.

Ordino provides the feature to group the values of numerical attributes into categories. Therefore, numerical attributes added after an item set similarity score was calculated are automatically stratified according to the thresholds determined by the server-side similarity scoring described in Section 3.3.3.

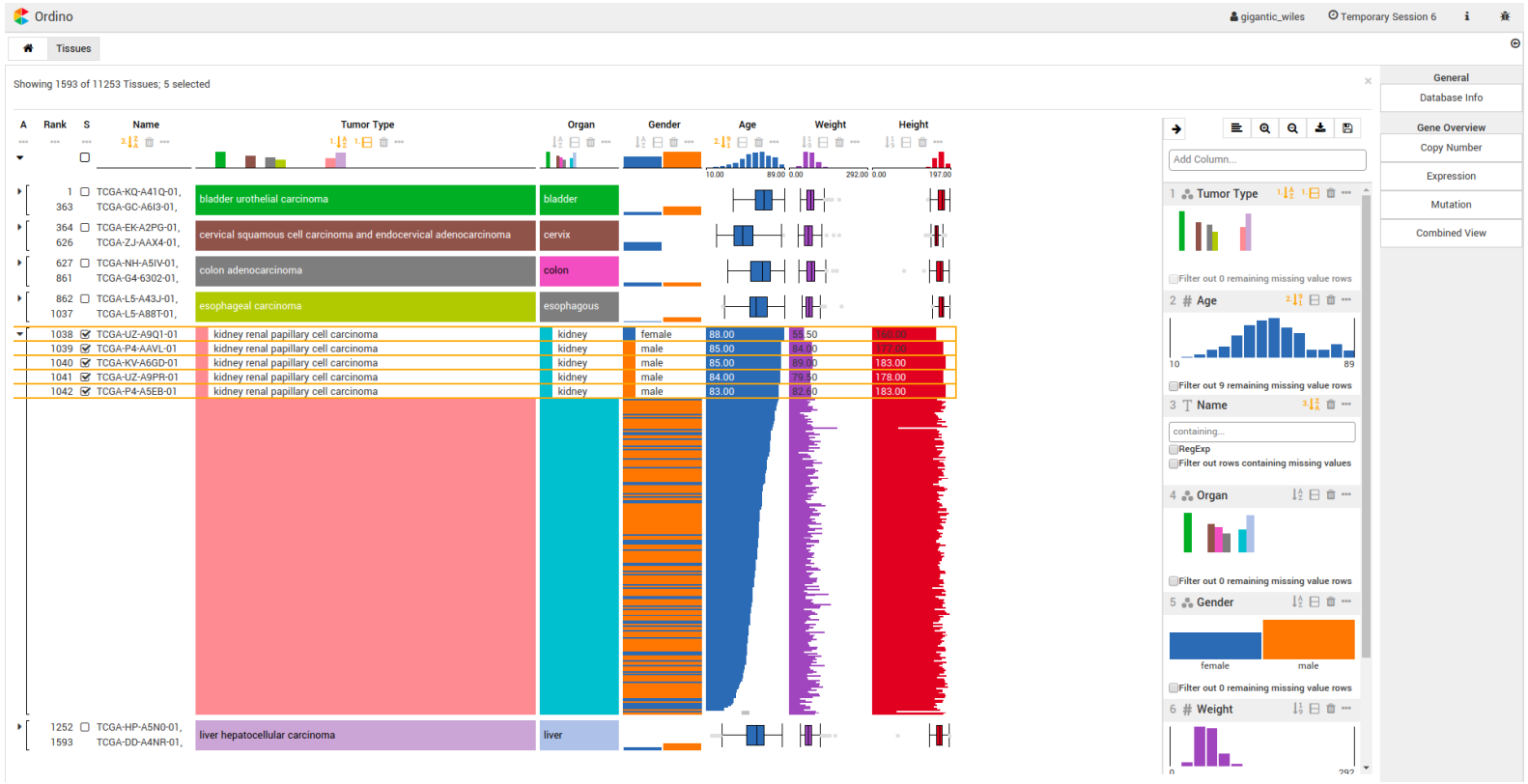


Figure 5.2: Ordino in overview mode with TCGA data for six tumor types. In addition to the column headers, the side bar on the right shows information on the attributes. The items are stratified by tumor type and items of tumors in bladder, cervix, colon, esophagous, and liver, are collapsed into a group representation. Because Ordino is in the overview mode, the items with kidney renal papillary cell carcinoma are represented by only a few pixels. The five oldest patient with a kidney tumor are selected and displayed in full height. The rightmost bar contains a list of detail views.

(a) Left-align icon which resembles the bars of the similarity scores.

(b) Spinning wheel to indicate ongoing computation of similarity scores.

Figure 5.3: Button to start the touring process and display its current state.

Due to Ordino's different data browser, the result of only one touring process can be shown at a time. This will always be the one that has finished last, which means that additionally started touring processes will overwrite the results of previous ones in the drop-down menu.

Based on the similarity scores beneath each attribute and the tooltips, users can continue their analysis by choosing an attribute, or, due to the results not reflecting the user's hypothesis, by varying input data and restarting the touring process.

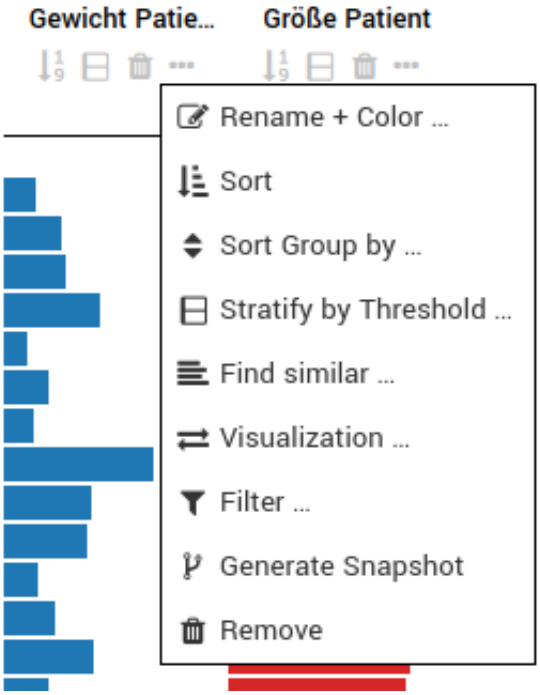


Figure 5.4: While the three main options of an attribute,i.e., sort, stratify, and remove, are always visible, additional actions are displayed via the more options button (...). The *Find similar...* button to start the touring process for an attribute was added to the menu. It uses the same icon as the button to start item set touring in Figure 5.3a.

CHAPTER 6

Results and Feedback

In the following section, the computational and visual results of the touring process are shown, followed by the feedback of future users and its implications.

6.1 Results

The touring process can compare attributes and item sets, for which we have implemented the most common similarity measure respectively—the Pearson correlation coefficient (PCC) and the Jaccard index. As seen in Chapter 2, state-of-the-art similarity measures compare data of the same type, or simply item set sizes. Item set comparisons are independent of the data type, but only categorical attributes can be used directly. We have resolved this restriction by discretizing numerical attributes and converting the subtrees of hierarchical attributes into separate categorical attributes. The touring process can therefore compare a given item set with every attribute from a dataset and return the most similar categories.

The touring process has been tested on a dataset of the Kepler University Hospital and two datasets of The Cancer Genome Atlas (TCGA), for *glioblastoma multiforme* [110] and *clear cell renal cell carcinoma* [111]. All three contain numerous categorical and numerical attributes. The dataset of the Kepler University Hospital also contains hierarchical attributes in the form of ICD-10 and ICD-O codes.

The touring process has been implemented in the two systems StratomeX and Ordino. Both are augmented with controls to trigger the touring and visualizations for the resulting similarity scores. Google's Chrome Browser [32] has been used during development to control correct representation and to test functionality. The implementations differ in terms of input selection and output representation. While with StratomeX, users can select item sets by clicking on categories or ribbons, selection of individual items is possible in Ordino, as well as starting the touring from numerical columns. In StratomeX,

users can choose the similarity measure to be used with a drop-down menu. In Ordino, the similarity measure is chosen automatically, with the PCC for numerical attributes, and the Jaccard index for items sets.

In StratomeX the touring process' output is added as additional column for an attribute in the data browser. One column is added with the score of the attribute's highest scoring category and one column with this category's name. The data browser uses LineUp [35] to display the data. By triggering the touring multiple times, multiple columns are added and due to LineUp's feature to combine attributes, similarity scores can be combined by the user to create new ones. In Ordino, the output is shown in the filterable drop-down menu used to add attributes. The similarity scores are presented by bars beneath each attribute. The exact values are shown as tooltip, if the user hovers the mouse cursor over an attribute.

6.1.1 Server-Side Performance

The computationally heavy work is executed server-side and in the following some performance measures are presented. As the selected input, be it an item set or an attribute, is compared with each available attribute, the performance of the similarity computation is directly dependent on the dataset's size and the data retrieval performance.

As the TCGA datasets are retrieved significantly faster than the data from the Kepler University Hospital's tumor dataset, despite having much more attributes, the TCGA KIRC dataset [111] is used for the following performance measurement. The dataset contains 35 010 attributes, of which 35 002 are categorical and four are numerical. The remaining four attributes are matrices that have been converted into multiple categorical attributes by clustering. The 35 002 categorical attributes have 177 532 categories in total.

From the attribute *M Stage*, which indicates whether patients have metastases, the category *m1* (patients with metastases) was selected in StratomeX and the touring process started by selecting the Jaccard index. In average, computation of the Jaccard index for category *m1* with all 177 532 categories, plus the discretization of numerical attributes, took 15.4 seconds, with 15.02 seconds for the fastest, and 16.06 for the slowest of 20 runs. The durations are taken from Celery's log messages, which report the needed time upon task completion. The task ran inside a Debian [15] Docker [16] container. The task queries the data from the Phovea Server, which is also running in a Debian Docker container. The measures were taken on the author's personal laptop with eight gigabytes of memory and an Intel i5-3210M with a base frequency of 2.5 gigahertz. Instruction to setup the Docker environment are available on Github [83; 85].

6.1.2 StratomeX

After the scores of Section 6.1.1 are computed, the attributes in the data browser are automatically sorted in descending order by similarity score, guiding users to the most similar ones. Figure 6.1 shows the attributes *M Stage* and *Stage*, which indicates the

severeness of a cancerous disease, and the similarity scores. The Jaccard index for the group *m1* of M Stage is of course 100 % and thus highest. With 93.8 %, the second highest score comes from the *Stage 4* category of the Stage attribute. As a tumor that generated one or more metastases is considered worst in nearly all cases, this relationship is also medically plausible. With StratomeX's ribbons, it can be seen that these two categories share nearly all items (see Figure 6.1).

Only some of the items (two) of category *m1* are classified as Stage 3, about as few items from category *m0* in category stage 4 (three items). In total, both categories share 75 items but contain 80 distinct items, resulting in the Jaccard index of $Sim_{Jaccard}(m1, Stage4) = \frac{75}{80} = 0.938$ (compare Equation 2.7).

The dataset of the Kepler University Hospital is used to show the touring process for hierarchical attributes. Two columns from the ICD-O hierarchy are shown in Figure 6.2. The highlighted part of the ribbons are patients that received chemotherapy and have a tumor grading of three. This item set was used to calculate similarity scores with the *Shared/Selected* measure. The results can be seen in Figure 6.3.

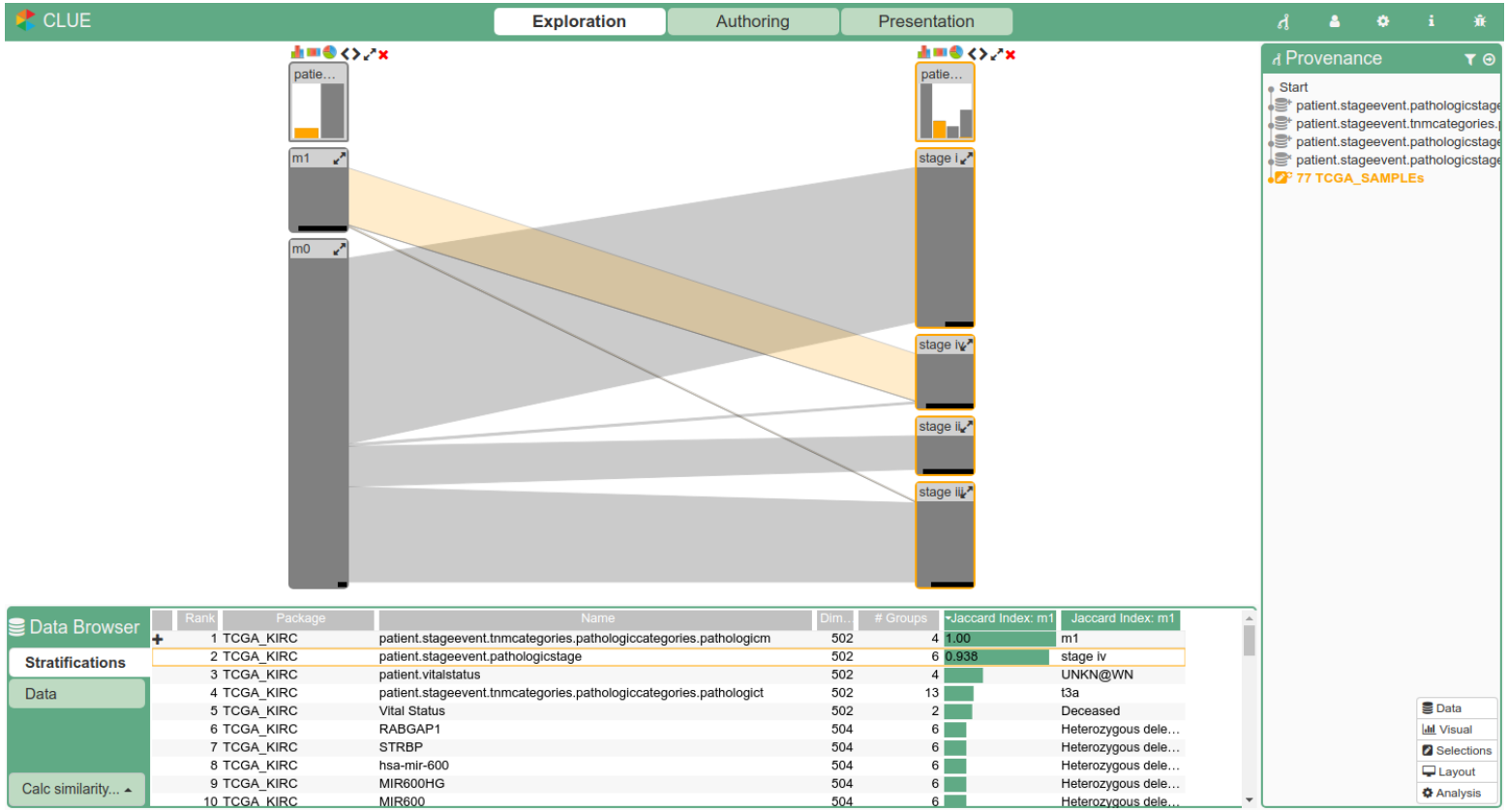


Figure 6.1: StratomeX with the attribute *Stage M* on the left and the highest scoring attribute *Stage* on the right. The similarity scores for the category *m1* can be seen in the data browser. The column headers shows the selected similarity measure (Jaccard) and the category (m1). The highlighted ribbon connects the two similar categories.

Figure 6.2: StratomeX with two columns of the hierarchical attribute *ICD-O Morphologie*. The first column represents the tree's root, while the second column contains all items of the subtree with epithelial neoplasms (ICD-O Morphology codes 801–804). The hierarchical structure can be seen by the ribbons that connect one category of the first column with the categories of the second column.

Rank	Dataset	Name	Dimensions	# Groups	Shared/Selected	Shared/Selected: True
1	Exploration Set Tumor	ICD-O Morphologie 814-838	1558	9	1.00	8230:3, Solides Karzinom o.n.A.
2	Exploration Set Tumor	N-Stadium	1558	10		3b
3	Exploration Set Tumor	ICD-O Morphologie 801-804	1558	6		8010:6, Karzinom-Metastase o....
4	Exploration Set Tumor	N-Stadium des Ausgangstumors (g...	1558	12		1b
5	Exploration Set Diagnose	Diagnose	9749	1738		L93.0, Diskoider Lupus erythema...
6	Exploration Set Tumor	T-Stadium posttherapeutisch postop...	1558	12		4a
7	Exploration Set Patient	Familienstand	1493	6		getrennt
8	Exploration Set Tumor	ICD-O Morphologie 856-857	1558	2	0.800	8575:3, Metaplastisches Karzin...
9	Exploration Set Tumor	Operationstechnik	1558	5		andere OP-Technik?
10	Exploration Set Tumor	ICD-O Morphologie 850-854	1558	19		8530:3, Inflammatorisches Karzin...
11	Exploration Set Tumor	ICD-O Morphologie	1558	12		ICD-O Morphologie 856-857

Figure 6.3: *Shared/Selected* Similarity scores for patients that received chemotherapy and have a tumor grading of three. The numerical values of the similarity scores are only shown for the select row (attribute *ICD-O Morphologie 856-857*) and the row first row, as the mouse cursor is placed over this row. From the hierarchical attribute *ICD-O Morphologie*, two subtrees have a category with a similarity of 100 % (ICD-O Morphologie 814-838 and 801-804). The categorical attribute *ICD-O Morphologie* that represents the root level of the hierarchy has a similarity of 66.7 % and is ranked eleventh..

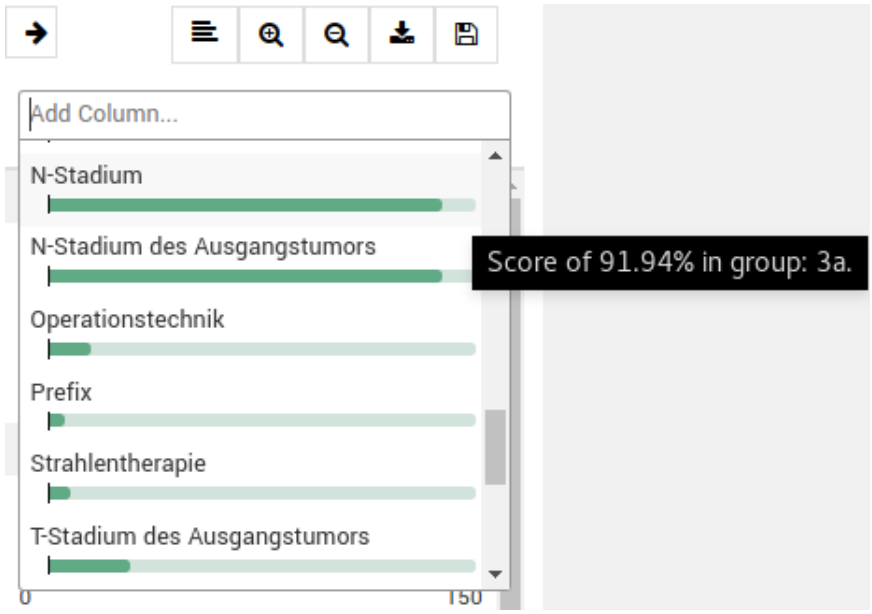


Figure 6.4: Similarity scores for six attributes in Ordino's drop-down menu for data selection. The actual score is visualized by a green bar on light green background to show the highest possible score in comparison. If the users hovers over an attribute with the mouse, a tooltip will appear showing the exact score and the category for which this score was calculated. The group *3a* of attribute *N Stage* (German: *N Stadium*) scored highest with a similarity score of 91.94 %.

6.1.3 Ordino

As mentioned above, items can be selected individually in Ordino. This can be done by simple clicks, brushing, or by selecting a group of items after aggregation. After selection, the touring process can be started and the results are shown in the filterable drop-down menu (see Figure 6.4).

The Kepler University Hospital's tumor dataset is used to display the results in Ordino. This dataset also contains stages of the cancer's spread into the surrounding tissues (Stage T) and lymph nodes (Stage N), similar to the *M Stage* from the TCGA KIRC dataset in the presented StratomeX result.

As selection of more than one category is possible in Ordino, we do so in the following example and select the most severe cases with N Stage of three or more, thus patients who have affected mammary and axilla lymph nodes [115]. These are three categories, namely *N3a* with 57 items, *N3b* with one item and *N3c* with four items. In Ordino, the Jaccard index is used as similarity measure for item sets. Due to the unequal distribution, one can foresee that the selected items will score a high similarity with the category *N3a*. With $Sim_{Jaccard}((N3a \cup N3b \cup N3c), N3a) = \frac{57}{62} = 0.9194$.

Figure 6.4 shows some of the available attributes in the drop-down. The attribute N

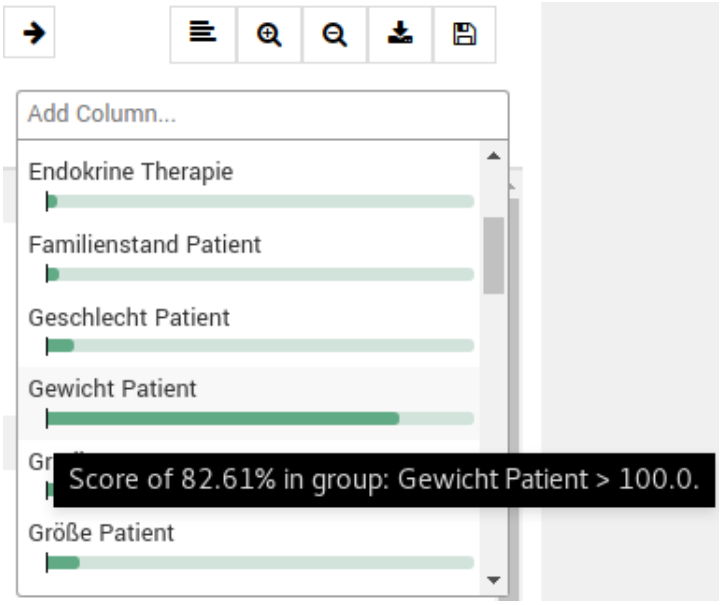


Figure 6.5: Similarity score for the 38 items with a body weight between 101 kg and 131 kg. The numerical attribute body weight (German *Gewicht Patient*) was categorized into a group of items with body weight less or equal 100 kg and a category with more than 100 kg. The latter achieved a Jaccard index of 82.61 %.

Stage occurs twice, once for the latest assessment and once for the original classification of lymph nodes. By the bar's width, an almost maximal score can be seen for both attributes. The score of the attribute N Stage, 91.94 %, is shown as tooltip, together with the category 3a. With Jaccard indices below 0.2, all other attributes showed no significant similarity.

To demonstrate the numerical splitting for similarity calculation (see Section 3.3.3), all 38 items with a body weight value between 101 kg and 131 kg are selected and the touring process is started. Like before, the Jaccard index is used as similarity measure and therefore determines the resulting threshold. In Figure 6.5 the resulting scores are shown and attribute body weight (German: *Gewicht*) shows a Jaccard index of 0.8261, if the attribute is split at a body weight value of 100 kg. As there are 38 items with weight between 101 kg and 131 kg and eight items with a weight higher than 131 kg, the Jaccard index can be checked with: $Sim_{Jaccard}(weight = [101, 131], weight = [101, \infty)) = \frac{38}{46} = 0.8261$. With Jaccard indices below 10 %, none of the dataset's other categories showed any significant similarity.

The third feature implemented in Ordino is the search for similar attributes. With the PCC a comparison method for numerical attributes was implemented. Therefore numerical attributes have the additional button *Find similar...* in their header (see Figure 5.4). The attribute's header also displays the distribution of the attribute's values.

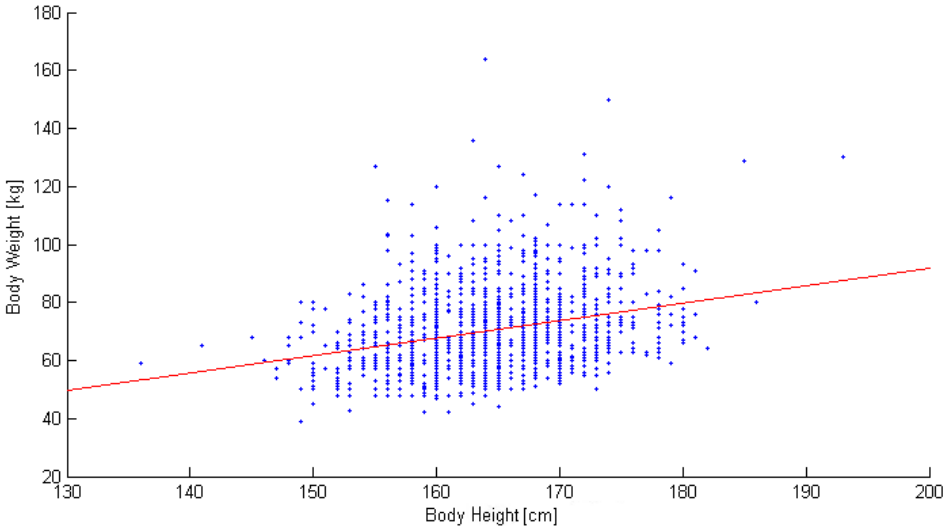


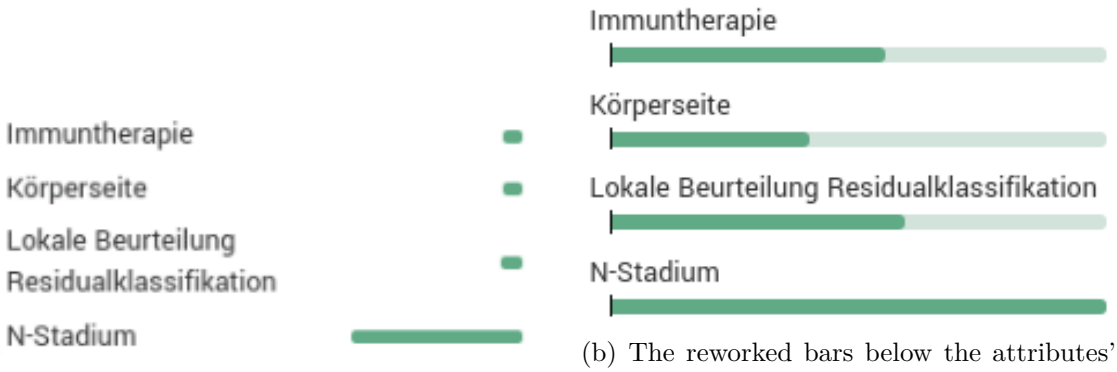
Figure 6.6: Scatter plot with the two attributes body weight and body height. Four outliers, which most probably have erroneously exchanged weight and height, are not included for a better representation of the majority of the values. A least-squares line is superimposed in red.

The attribute body weight is used again to demonstrate the similarity scores. As only numerical attributes are compared, only those yield a similarity score. From the Kepler University Hospital's dataset, the attribute body height scored highest with a PCC of 0.6826, indicating a medium to major correlation (compare Table 2.1 and Figure 6.6). The PCCs of the other attributes are below ± 0.06, i.e., show no correlation.

6.2 Feedback

During the development of the touring process, the achieved progress was reported to clinicians of the Kepler University Hospital on a regular basis. Their feedback was gladly accepted, integrated, and presented at later meetings. The changes that resulted from this agile development are presented in this section.

Apart from the minor visual adaptions, users were interested in the touring process right away. Finding potential causes for irregular attribute combinations, such as cases that would normally receive endocrine therapy, but had none, or whether adipose patients are over-represented in categories of other attributes (as in the example above), are some of the potential use cases.



(a) The initial similarity score bars next to the attribute names. The bars take up one third of the horizontal space in the drop-down menu.

(b) The reworked bars below the attributes' names. They are slightly indented and have a light background to show their maximum length.

Figure 6.7: Bars for similarity scores before and after user feedback.

6.2.1 Feedback to StratomeX Integration

During the initial integration in StratomeX, only one column containing the similarity score was added to the data browser. To quickly judge the relevance of a high scoring attribute, the second column to display the category was added.

Additionally, the two proportion scores *Shared/Selected* and *Shared/Compared* were introduced for an intuitive comparison with the selected item set (also see Section 3.3.2).

6.2.2 Feedback to Ordino Integration

As Ordino does not show the attributes, which can be added into the view, in the main interface like StratomeX, an alternative visualization of the ongoing server-side computation was necessary. As a result the touring process button is animated while similarity scores are calculated.

Another change concerned the bars in the search box that represent the similarity score. These were originally right aligned in the drop-down to save space, but the users could not determine how high a score could be, and the difference between bars was harder to see due to their small width.

In Figure 6.7 one can see the original and reworked versions. We added a light background to the bars to show their potential maximum. The bars are also slightly indented to indicate where they belong (like subitems in lists). A small vertical mark at the beginning of the bars serves as visual support to estimate and compare low similarity scores.

CHAPTER 7

Conclusion and Discussion

Healthcare and numerous other research fields face an ever increasing amount of data. Through the ongoing digitization, more data is available. Scientific advances discover new attributes that need to be recorded to improve patient care. Improvements in existing techniques finally enhance the resolution or extend the set of collected data.

While the collected amount and the ability to store data have increased rapidly, the means to process this data did not [31]. The idea of VA is to combine an automated analysis with analysis by humans. While humans can quickly identify relationships and patterns in a visualization, the computational power surpasses them in the calculation and comparison of defined measures across whole datasets. Given that researchers are experts in the domain of the data, they have hypotheses to test and need assistance while using the VA tool to do so.

This thesis has described a universal process to find attributes of interest for the users to continue their exploration. Similar attributes can be queried based on item sets or attributes and various different similarity measures. The similarity measures for item sets also use attributes that have no inherent categorization of their values, as well as hierarchical attributes with nested categories.

The touring process uses a client-server architecture. The data subset for which guidance is needed and the similarity measure are selected on the client-side. The server-side retrieves the whole dataset and starts to compare it with the given subset.

Starting from an item set, it is compared to each category of a categorical attribute. Numerical attributes are categorized to achieve maximal similarity and hierarchical attributes are traversed in order to find the most similar subtree of every parent node. Therefore the prominence of items with certain characteristics that are assumed to be relevant, can be checked across all attributes to prove hypotheses or find potentially interesting relationships.

If the touring process is started with an attribute as input, that attribute is also compared to every other attribute in the dataset. The Pearson correlation coefficient has been implemented to compare numerical attributes. The resulting similarity scores give insights on possible dependencies, without the need to visually compare all attributes.

While the server-side is processing, the users are informed of the ongoing computation until the results can be retrieved. The similarity scores are presented as bars alongside their respective attributes. The user can utilize this information to pick an attribute with high similarity for further analysis, or vary the selected data subset based on the reported relationships.

The presented touring process is independent of the exploratory visualization technique used, which was demonstrated by the integration in the two tools StratomeX and Ordino. Furthermore, the data domain is irrelevant to the touring process, which is demonstrated by the usage of two TCGA datasets as well as the dataset provided by the Kepler University Hospital. Processing flat numerical and categorical as well as hierarchical data, it is also generalized for different data types.

7.1 Discussion

A touring process that compares a multi-attribute dataset and reports similarity measures to the users was implemented in the VA tools StratomeX and Ordino. Similar categories and attributes can be found for item sets and attributes respectively. As both tools are web applications and thus share a client-server architecture, the user only selects the input and receives the output on the client, while the computationally heavy work is done server-side. As a side effect, this also favors the connection to the database and does not impact the client-side system requirements.

The described solution is generalized for data types and domains and independent of the employed visual analysis tool. While the implemented similarity measures are among the most commonly used ones, knowledge of the data types and domain can be used to select, add, or configure similarity measures and optimize the guidance. By providing guidance not only in the data domain, but also in the domain of VA methods, the hurdle of choosing from more methods and configurations could be overcome by assisting users in selecting a similarity measure.

As this thesis' idea originated from an ongoing project with the Kepler University Hospital to visualize data of patients with mamma carcinoma, the main target of the touring process are the hospital's medical professionals. Advancements during development were presented to them to obtain and integrate feedback.

Data exploration, quick, intuitive tests of hypotheses, and the retrieval of attributes to which a data subset shows relationships, are the main tasks. As the usage of a VA tool is at best only a facet in the daily routine, simplicity is more important than feature richness. That is why the implementation in Ordino uses predefined similarity measures for the two types of input data.

The touring process we present in this thesis tries to find attributes that have relationship to a given data subset. Instead of attributes, the touring process can also try to find items with similar characteristics to a given item set. If the data of new patients is continuously integrated, the touring process could support clinicians in their treatment decisions by finding similar old cases to reason about new ones.

Gower's similarity coefficient, discussed in Section 2.3.2, would be an appropriate similarity measure to find similar items. However, through the sort, stratify, and filter operations in Ordino, adjacent items already show high similarity.

In contrast to the guided StratomeX [106] discussed in Section 2.2, the presented touring process is detached from the visualization and its data scope and can therefore be used by any VA tool. It also supports hierarchical attributes and can discretize numerical attributes into groups to compare them to a given item set.

7.2 Future Work

Besides the guidance to relevant attributes from the user's input, a guidance towards similar items would be worth striving for. Selecting a set of reference items to find similar ones in the data may be used for the above mentioned reasoning by similarity. In the case of patients, a new patient's condition could be compared to the original condition of past cases to find promising treatments. Similarity measures for clustering items such as Gower's, discussed in Section 2.3, can be used to compare items by multiple attributes. Many more measures to cluster items can be found in the literature [2; 5; 18; 119; 134].

As StratomeX and Ordino both display all stored items at any time, similar items cannot simply be added to the reference one. Appropriate means to communicate the similarity of items to users are needed. One simple solution would be to create a new attribute that contains the similarity score for each item, either as a numerical value or divided into multiple categories.

Following this thought, more similarity measures can be added. The PCC can be calculated only on the selected set of items to see whether a subset correlates with other attributes. Similarity measures to compare categorical attributes as a whole are still absent and a measure that handles numerical as well as categorical attributes would be desirable. Knowledge of the data domain can be used as an input to tailor similarity measures at the cost of general applicability.

Enhancement of numerical attribute discretization is also a potential candidate for further refinements. Instead of creating two groups for values above and below a threshold, the most similar value window could be retrieved from an attribute. Ordino already offers partitioning of numerical attributes by multiple thresholds and the server-side discretization already returns the threshold inside an array. Methods described in Section 2.4 also create multiple bins, e.g., by simply moving an equal amount of items into each bin (equal frequency discretization) or by splitting at points where the entropy gain is highest (MDL).

The guidance to similar items, additional similarity measures, and the specification of a discretization method add further complexity to the guidance process, contrary to its purpose. If the proposed enhancements are integrated, the scope of the touring process should also be extended. By giving guidance on the domain of VA methods, the touring process can also support users specifying the inputs appropriately.

APPENDIX A

Appendix

Listing A.1: MATLAB Code to create a random numerical attribute *Age* and split it into groups where the similarity is highest. Creation of the individual plots is omitted for the sake of clarity, but the resulting groups are printed

```
% get two normal distributions to create attribute 'Age'
fem = floor(normrnd(66,8,1,100)); % 100 items of cat 'female'
male = floor(normrnd(58,5,1,80)); % 80 items of cat 'male'

% count each age. shortens array to the respective maximal age
% first element is age 1, last element is max(fem) / max(male)
binnedFem = accumarray(fem(:),1);
binnedMale = accumarray(male(:),1);

%lowest/highest age of female and male
first = min(min(fem),min(male))-1;
last = max(max(fem), max(male))+1;

% make array lengths equal by padding with zeros
binnedFem(length(binnedFem)+1:last) = 0;
binnedMale(length(binnedMale)+1:last) = 0;

% remove leading zeros (indexes below minimal age)
binnedFem = binnedFem(first:last);
binnedMale = binnedMale(first:last);

% aggregated histograms (for age >= x)
aggHistFem = cumsum(binnedFem); % sum up
aggHistMale = cumsum(binnedMale);
```

```
25
26 % aggregated histograms (for age <= x)
27 % reverse vectors to sum up from end to start
28 aggHistFem_rev = cumsum(flipud(binnedFem));
29 aggHistMale_rev = cumsum(flipud(binnedMale));
30
31 x = first:last; % x axis
32
33 femJaccards = aggHistFem./(max(aggHistFem)+aggHistMale);
34 femJaccards_rev = flipud(aggHistFem_rev./(max(aggHistFem_rev)+
      aggHistMale_rev)); %reverse back to first:last ordering
35 [femJaccard, femSplitIndex] = max(femJaccards);
36 [femJaccard_rev, femSplitIndex_rev] = max(femJaccards_rev);
37 if (femJaccard >= femJaccard_rev)
38     femJaccard = femJaccard;
39     femStart = first; % from first
40     femSplit = x(femSplitIndex);  % to split
41 else
42     femJaccard = femJaccard_rev;
43     femStart = last; % from last
44     femSplit = x(femSplitIndex_rev);  %to split
45 end
46
47 maleJaccards = aggHistMale./(max(aggHistMale)+aggHistFem);
48 maleJaccards_rev = flipud(aggHistMale_rev./(max(aggHistMale_rev
      )+aggHistFem_rev)); %reverse back to first:last ordering
49 [maleJaccard, maleSplitIndex] = max(maleJaccards);
50 [maleJaccard_rev, maleSplitIndex_rev] = max(maleJaccards_rev);
51 if (maleJaccard >= maleJaccard_rev)
52     maleJaccard = maleJaccard;
53     maleStart = first; % from first
54     maleSplit = x(maleSplitIndex); %to split
55 else
56     maleJaccard = maleJaccard_rev;
57     maleStart = last; % from last
58     maleSplit = x(maleSplitIndex_rev); %to split
59 end
60
61 disp(sprintf('Female: Jaccard Score of %f with region from %d
      to %d.', femJaccard, femStart, femSplit))
62 disp(sprintf('Male: Jaccard Score of %f with region from %d to
      %d.', maleJaccard, maleStart, maleSplit))
```

Listing A.2: An excerpt of the configuration for the ICD-O Morphology (German version) to convert it into categorical attributes.

```
"hierarchies": {
    "ICD-O␣Morphologie": {
      "includeOthers": true,
      "children": {
        "800-800": {
          "includeOthers": false,
          "label": "Neoplasien o.n.A.",
          "startWith": ["800"]
        },
        "801-804": {
          "includeOthers": false,
          "label": "Epitheliale Neoplasien o.n.A.",
          "startWith": ["801", "802", "803", "804"]
        },
        "805-808": {
          "includeOthers": true,
          "label": "Plattenepithelneoplasien",
          "startWith": ["805", "806", "807", "808"]
        },
        ...
        "959-972": {
          "label": "Hodgkin- und Non-Hodgkin-Lymphome",
          "startWith": ["959", "960", "961", "962", "963",
             "964", "965", "966", "967", "968", "969", "970",
             "971", "972"],
          "children": {
            "959-959": {
              "label": "Maligne Lymphome, o.n.A. oder diffus",
              "startWith": ["959"]
            },
            "965-966": {
              "label": "Hodgkin-Lymphome",
              "startWith": ["965", "966"]

            },
            "967-972": {
              "label": "Non-Hodgkin-Lymphome",
              "startWith": ["967", "968", "969", "970", "971",
                 "972"],
              "children": {
                "967-969": {
```

```
                "label": "Reifzellige B-Zell-Lymphome",
                  "startWith": ["967", "968", "969"]

                },
                "970-971": {
                "label": "Reifzellige T- und NK-Zell-Lymphome",
                  "startWith": ["970", "971"]

                },
                "972-972": {
                "label": "Lymphoblastische Lymphome der
                            Vorläuferzellen",
                  "startWith": ["972"]
                }
              }
            }
          }
        },
        ...
      }
    }
  }
```

List of Figures

2.1 Line- and point-based visualizations from Flexible Linked Axes (FLINA) 8
2.2 Variations of stacked-bar charts 10
2.3 Stream graph with movie revenues 11
2.4 Examples of overview visualization techniques. 12
2.5 Fisher's Iris flower data set and three dimensionality reduced visualizations 14
2.6 Probing Projections visualization for dimensionality reduced data 15
2.7 ProxiLens visualization for dimensionality reduced data 15
2.8 LineUp comparing two rankings of universities 16
2.9 Taggle in detail mode. 17
2.10 Taggle in overview mode. 18
2.11 Integrative Genomics Viewer application window 20
2.12 Circos' circular data visualization 21
2.13 Gene-gene interaction network visualized with Cytoscape.js 22
2.14 Example of a heatmap with genomic data 23
2.15 StratomeX with three heatmap columns 25
2.16 LifeFlow combined with LifeLines2 27
2.17 Characterization of guidance in visual analytics 28
2.18 Example of a tree with 10 nodes 36

3.1 Schematic representation of the touring process 40
3.2 Van Wijk's model of visualization extended by guidance components . . . 42
3.3 Visual representation of the implemented item set similarity methods . . 45
3.4 Discretization of a numerical attribute 48
3.5 Conversion of a hierarchical attribute to categorical attributes 50

4.1 Architecture overview 55

5.1 StratomeX user interface 63
5.2 Ordino user interface 65
5.3 Touring process button in Ordino 66
5.4 Button in Ordino to start the touring process for an attribute 67

6.1 StratomeX with computed similarity scores 72
6.2 Columns from a hierarchical attribute in StratomeX 73

6.3 StratomeX data browser with scores for hierarchical attribute subtrees . . 73
6.4 Similarity scores for six attributes in Ordino 74
6.5 Similarity score of a categorized numerical attribute 75
6.6 Scatter plot of body weight and height 76
6.7 Bars for similarity scores before and after user feedback. 77

List of Tables

2.1 Correlation coefficient ranges and their significance 31
2.2 Gower's similarity coefficient for binary data 34
2.3 Similarity and distance scores of tree node pairs 36

3.1 Guidance characteristics of the proposed touring process 42

4.1 Identifiers for similarity measures 56

List of Listings

4.1 Base class for similarity measures 57
4.2 Configuration for hierarchical attribute categorization 59

A.1 MATLAB Code to split numerical groups 83
A.2 JSON configuration for ICD-O Morphology 85

List of Equations

1.1 Binomial coefficient for combinations of two attributes 3

2.1 Conversion of distance to similarity measures 30
2.2 Pearson correlation coefficient 30
2.3 Spearman's rank correlation coefficient 31
2.4 Kendall rank correlation coefficient 31
2.5 Minkowski distance 32
2.6 Szymkiewicz-Simpson or Overlap coefficient 32
2.7 Jaccard index with sets 33
2.8 Binary Jaccard index 33
2.9 Sørensen–Dice coefficient 33
2.10 Rand index 33
2.11 Gower similarity coefficient 34
2.12 Gower similarity for numerical attributes 34
2.13 Gower similarity for categorical attributes 34
2.14 Similarity of two items in a hierarchical attribute by Wu and Palmer . . . 35
2.15 Distance of two items in a hierarchical attribute by Girardi et al. 35
2.16 Distance of hierarchical item sets 35
2.17 Relation of the similarity measure by Wu and Palmer and the distance measure by Girardi et al.. 36
2.18 Heuristic to choose number of bins for discretization 37

3.1 Pearson correlation coefficient 43
3.2 Item set similarity with Jaccard index 46
3.3 Shared/Selected item set similarity 46
3.4 Shared/Compared item set similarity 46

Acronyms

API Application Programming Interface. 4, 56, 57, 59, 60, 63

bp base pairs. 22

CGM CompuGroup Medical. 27

ChIP-seq chromatin immunoprecipitation DNA-sequencing. 21

CLUE Capture, Label, Understand, Explain. 64

DICOM Digital Imaging and Communications in Medicine. 2

DICON Dynamic Icons. 28, 29

EHR Electronic Health Record. 1, 27, 28, 30

FHIR Fast Healthcare Interoperability Resources. 2

FLINA Flexible Linked Axes. 8, 11

HDF Hierarchical Data Format. 56, 57

HER2 human epidermal growth factor receptor 2. 53

ICD International Classification of Diseases. 2, 3, 35, 43, 51, 71

ICD-O ICD for Oncology. 2, 3, 35, 61, 71, 75, 87

ICGC International Cancer Genome Consortium. 20

ICU Intensive Care Unit. 27

IGV Integrative Genomics Viewer. 20, 21

JSON JavaScript Object Notation. 60, 61

KRCC Kendall rank correlation coefficient. 32, 46

LOINC Logical Observation Identifiers Names and Codes. 2

MDL Minimum Description Length. 38, 83

MDS Multidimensional Scaling. 13, 14

MTSA Multivariate Time Series Amalgam. 27

NGS next-generation sequencing. 21

PCA principal component analysis. 13–15

PCC Pearson correlation coefficient. 4, 31, 32, 43, 45, 46, 58–61, 66, 71, 72, 77, 82, 83

PCP parallel coordinates plot. 8, 9, 12

REST Representational State Transfer. 4, 56, 60, 63

SNOMED CT Systematized Nomenclature of Medicine – Clinical Terms. 2

SQL Structured Query Language. 56, 57

SRCC Spearman's rank correlation coefficient. 31, 32, 46

t-SNE t-distributed stochastic neighbor embedding. 13, 14

TCGA The Cancer Genome Atlas. vii, 1, 3, 20, 58, 67, 71, 72, 75, 82

UCSC University of California, Santa Cruz. 20

VA Visual Analytics. ix, xi, 2–5, 7, 29, 30, 41, 43, 44, 51, 53, 81–84

VISITORS Visualization of Time-Oriented Records. 28

Bibliography

[1] H. Abdi and L. J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010. ISSN 1939-0068. doi: 10.1002/wics.101. URL http://onlinelibrary.wiley.com/doi/10.1002/wics.101/abstract.

[2] M. Alamuri, B. R. Surampudi, and A. Negi. A survey of distance/similarity measures for categorical data. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 1907–1914. IEEE, 2014.

[3] B. Alpern and L. Carter. The hyperbox. In *Proceedings of the 2nd conference on Visualization'91*, pages 133–139. IEEE, 1991.

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

[5] S. Boriah, V. Chandola, and V. Kumar. Similarity measures for categorical data: A comparative evaluation. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 243–254. SIAM, 2008.

[6] B. Boutsinas and T. Papastergiou. On clustering tree structured data with categorical nature. *Pattern Recognition*, 41(12):3613–3623, 2008.

[7] L. Byron and M. Wattenberg. Stacked Graphs - Geometry & Aesthetics. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '08)*, 14(6):1245 –1252, 2008. ISSN 1077-2626. doi: 10.1109/TVCG.2008.166.

[8] M. Bärtschi. Health Data Visualization - A review. Seminar Report, University of Fribourg, Switzerland, 2011. URL http://human-ist.unifr.ch/seminars/collaborative-data-visualization-seminar. Accessed: 2018-03-13.

[9] Caleydo. Homepage. URL http://caleydo.org/. Accessed: 2018-08-11.

[10] Celery. Distributed Task Queue. URL http://www.celeryproject.org/. Accessed: 2018-08-11.

[11] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H.-J. Schulz, M. Streit, and C. Tominski. Characterizing Guidance in Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics (VAST'16)*, 23(1):111–120, 2017. doi: 10.1109/TVCG.2016.2598468.

[12] E. Cerami, J. Gao, U. Dogrusoz, B. E. Gross, S. O. Sumer, B. A. Aksoy, A. Jacobsen, C. J. Byrne, M. L. Heuer, E. Larsson, Y. Antipin, B. Reva, A. P. Goldberg, C. Sander, and N. Schultz. The cBio Cancer Genomics Portal: An Open Platform for Exploring Multidimensional Cancer Genomics Data. *Cancer discovery*, 2(5): 401–404, May 2012. ISSN 2159-8274. doi: 10.1158/2159-8290.CD-12-0095.

[13] N. S. Chok. *Pearson's versus Spearman's and Kendall's correlation coefficients for continuous data.* PhD Thesis, University of Pittsburgh, 2010.

[14] J. H. Claessen and J. J. van Wijk. Flexible Linked Axes for Multivariate Data Visualization. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '11)*, 17(12):2310–2316, 2011. doi: 10.1109/TVCG.2011.201.

[15] Debian. The Universal Operating System. URL `https://www.debian.org/index.en.html`. Accessed: 2018-08-12.

[16] Docker. Build, Ship, and Run Any App, Anywhere. URL `https://www.docker.com/`. Accessed: 2018-08-11.

[17] Docker Hub. MariaDB. URL `https://hub.docker.com/_/mariadb/`. Accessed: 2018-08-17.

[18] T. dos Santos and L. Zárate. Categorical data clustering: What similarity measure to recommend? *Expert Systems with Applications*, 42(3):1247–1260, 2015.

[19] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *Machine Learning Proceedings 1995*, pages 194–202. Elsevier, 1995.

[20] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[21] Flask. A Python Microframework. URL `http://flask.pocoo.org/`. Accessed: 2018-08-11.

[22] Flask 0.12.4 documentation. API. URL `http://flask.pocoo.org/docs/0.12/api/#flask.json.jsonify`. Accessed: 2018-08-11.

[23] Font Awesome. Icons. URL `https://fontawesome.com/icons`. Accessed: 2018-08-11.

[24] M. Franz, C. T. Lopes, G. Huck, Y. Dong, O. Sumer, and G. D. Bader. Cytoscape.js: a graph theory library for visualisation and analysis. *Bioinformatics*, 32(2): 309–311, 2015.

[25] K. Furmanova, S. Gratzl, H. Stitz, T. Zichner, M. Jaresova, M. Ennemoser, A. Lex, and M. Streit. Taggle: Scalable Visualization of Tabular Data through Aggregation. *arXiv preprint*, 2018. URL `https://arxiv.org/abs/1712.05944`.

[26] J. Gama, L. Torgo, and C. Soares. Dynamic discretization of continuous attributes. In *Ibero-American Conference on Artificial Intelligence*, pages 160–169. Springer, 1998.

[27] N. Gehlenborg, S. I. O'Donoghue, N. S. Baliga, A. Goesmann, M. A. Hibbs, H. Kitano, O. Kohlbacher, H. Neuweger, R. Schneider, D. Tenenbaum, and A.-C. Gavin. Visualization of omics data for systems biology. *Nature Methods*, 7(3):56–68, 2010. ISSN 1548-7105. doi: 10.1038/nmeth.1436.

[28] D. Girardi, S. Wartner, G. Halmerbauer, M. Ehrenmüller, H. Kosorus, and S. Dreiseitl. Using concept hierarchies to improve calculation of patient similarity. *Journal of biomedical informatics*, 63:66–73, 2016.

[29] Gitools 2.3.0 documentation. Tutorial 4.1. TFBS enrichment analysis. URL `http://www.gitools.org/docs/Tutorials_Tutorial41.html`. Accessed: 2018-08-04.

[30] S. Glenn. Kendall's Tau (Kendall Rank Correlation Coefficient). URL `http://www.statisticshowto.com/kendalls-tau/`. Accessed: 2018-03-19.

[31] P. Godfrey, J. Gryz, and P. Lasek. Interactive visualization of large data sets. *IEEE Transactions on Knowledge and Data Engineering*, 28(8):2142–2157, 2016.

[32] Google. Chrome-Browser. URL `https://www.google.com/intl/de_ALL/chrome/`. Accessed: 2018-08-11.

[33] D. Gotz, J. Sun, N. Cao, and S. Ebadollahi. Visual cluster analysis in support of clinical decision intelligence. In *AMIA Annual Symposium Proceedings*, volume 2011, page 481. American Medical Informatics Association, 2011.

[34] S. Gratzl. *Visually Guiding Users in Selection, Exploration, and Presentation Tasks*. PhD thesis, Johannes Kepler University Linz, Linz, Mar. 2017.

[35] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. LineUp: Visual Analysis of Multi-Attribute Rankings. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 19(12):2277–2286, 2013. doi: 10.1109/TVCG.2013.173.

[36] S. Gratzl, N. Gehlenborg, A. Lex, H. Pfister, and M. 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. ISSN 1077-2626. doi: 10.1109/TVCG.2014.2346260.

[37] S. Gratzl, N. Gehlenborg, A. Lex, H. Strobelt, C. Partl, and M. 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.

[38] S. Gratzl, A. Lex, N. Gehlenborg, N. Cosgrove, and M. Streit. From Visual Exploration to Storytelling and Back Again. *Computer Graphics Forum*, 35(3): 491–500, 2016. ISSN 1467-8659. doi: 10.1111/cgf.12925.

[39] Health Level 7. Fast Healtcare Interoperability Resources. URL `https://www.hl7.org/fhir/`. Accessed: 2018-06-30.

[40] J. Heer, M. Bostock, and V. Ogievetsky. A tour through the visualization zoo. *Commun. ACM*, 53(6):59–67, June 2010. ISSN 0001-0782. doi: 10.1145/1743546.1743567. URL `http://doi.acm.org/10.1145/1743546.1743567`.

[41] N. Heulot, J.-D. Fekete, and M. Aupetit. Proxilens: Interactive exploration of high-dimensional data using projections. In *VAMP: EuroVis Workshop on Visual Analytics using Multidimensional Projections*. The Eurographics Association, 2013.

[42] K. Hinum, S. Miksch, W. Aigner, S. Ohmann, C. Popow, M. Pohl, and M. Rester. Gravi++: Interactive Information Visualization to Explore Highly Structured Temporal Data. *J. UCS*, 11(11):1792–1805, 2005. URL `http://jucs.org/jucs_11_11/gravi_interactive_information_visualization/jucs_11_11_1792_1805_hinum.pdf`.

[43] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

[44] J. Iavindrasana, G. Cohen, A. Depeursinge, H. Müller, R. Meyer, and A. Geissbuhler. Clinical data mining: a review. *Yearbook of medical informatics*, 18(01):121–133, 2009.

[45] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(4): 69–91, 1985. doi: 10.1007/BF01898350.

[46] P. Jaccard. The Distribution of the Flora in the Alpine Zone. *New Phytologist*, 11 (2):37–50, Feb. 1912. ISSN 1469-8137. doi: 10.1111/j.1469-8137.1912.tb05611.x. URL `http://dx.doi.org/10.1111/j.1469-8137.1912.tb05611.x`.

[47] E. Jones, T. Oliphant, and P. Peterson. *SciPy: Open source scientific tools for Python*. 2001. URL `http://www.scipy.org/`. Accessed: 2018-08-13.

[48] M. Kern, A. Lex, N. Gehlenborg, and C. R. Johnson. Interactive visual exploration and refinement of cluster assignments. *BMC bioinformatics*, 18(1):406, 2017. doi: 10.1101/123844. URL `http://biorxiv.org/content/early/2017/04/04/123844`.

[49] A. Kirk. Making sense of streamgraphs, Aug. 2010. URL http://www.visualisingdata.com/2010/08/making-sense-of-streamgraphs/. Accessed: 2018-08-05.

[50] D. Klimov and Y. Shahar. A framework for intelligent visualization of multiple time-oriented medical records. In *AMIA Annual Symposium Proceedings*, volume 2005, page 405. American Medical Informatics Association, 2005.

[51] G. Kopanitsa, C. Hildebrand, J. Stausberg, and K. Englmeier. Visualization of medical data based on EHR standards. *Methods of Information in Medicine*, 52 (01):43–50, 2013.

[52] R. Kosara, F. Bendix, and H. Hauser. Parallel Sets: Interactive Exploration and Visual Analysis of Categorical Data. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):558–568, 2006. ISSN 1077-2626. doi: 10.1109/TVCG.2006.76.

[53] M. Krzywinski, J. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra. Circos: An information aesthetic for comparative genomics. *Genome Research*, 19(9):1639–1645, 2009. doi: 10.1101/gr.092759.109.

[54] A. Lex. *Visualization of Multidimensional Data with Applications in Molecular Biology.* Doctoral Thesis, Graz University of Technology, Graz, Mar. 2012. URL http://www.icg.tugraz.at/Members/alex/2012_phd_thesis_lex.pdf.

[55] A. Lex, M. Streit, E. Kruijff, and D. Schmalstieg. Caleydo: Design and Evaluation of a Visual Analysis Framework for Gene Expression Data in its Biological Context. In *Proceedings of the IEEE Symposium on Pacific Visualization (PacificVis '10)*, pages 57–64. IEEE, 2010. doi: 10.1109/PACIFICVIS.2010.5429609.

[56] A. Lex, M. Streit, H.-J. Schulz, C. Partl, D. Schmalstieg, P. J. Park, and N. Gehlenborg. StratomeX: Visual Analysis of Large-Scale Heterogeneous Genomics Data for Cancer Subtype Characterization. *Computer Graphics Forum (EuroVis '12)*, 31 (3):1175–1184, 2012. ISSN 0167-7055. doi: 10.1111/j.1467-8659.2012.03110.x.

[57] LibreOffice. Calc. URL https://de.libreoffice.org/discover/calc/. Accessed: 2018-03-13.

[58] M. Lind, J. Johansson, and M. Cooper. Many-to-many relational parallel coordinates displays. In *Information Visualisation, 2009 13th International Conference*, pages 25–31. IEEE, 2009.

[59] S. Liu, D. Maljovec, B. Wang, P. T. Bremer, and V. Pascucci. Visualizing High-Dimensional Data: Advances in the Past Decade. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1249–1268, Mar. 2017. ISSN 1077-2626. doi: 10.1109/TVCG.2016.2640960.

[60] C. T. Lopes, M. Franz, F. Kazi, S. L. Donaldson, Q. Morris, and G. D. Bader. Cytoscape Web: an interactive web-based network browser. *Bioinformatics*, 26(18): 2347–2348, 2010.

[61] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[62] D. M. Maslove, T. Podchiyska, and H. J. Lowe. Discretization of continuous features in clinical datasets. *Journal of the American Medical Informatics Association : JAMIA*, 20(3):544–553, June 2013. ISSN 1067-5027 1527-974X. doi: 10.1136/amiajnl-2012-000929.

[63] Mathworks. MATLAB. URL `https://de.mathworks.com/products/matlab.html`. Accessed: 2018-08-11.

[64] K. Meurer. A Simple Guide to Binning Data Using an Entropy Measure. URL `http://kevinmeurer.com:80/a-simple-guide-to-entropy-based-discretization/`. Accessed: 2018-03-28.

[65] Microsoft. Excel 2016. URL `https://products.office.com/en-us/excel/`. Accessed: 2018-03-13.

[66] J. Miguel. Quantitative Methods: Some Basic Techniques in Data Mining. URL `http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/MetQ/Talk6.pdf`. Accessed: 2018-03-20.

[67] V. K. Mootha, C. M. Lindgren, K.-F. Eriksson, A. Subramanian, S. Sihag, J. Lehar, P. Puigserver, E. Carlsson, M. Ridderstråle, E. Laurila, N. Houstis, M. J. Daly, N. Patterson, J. P. Mesirov, T. R. Golub, P. Tamayo, B. Spiegelman, E. S. Lander, J. N. Hirschhorn, D. Altshuler, and L. C. Groop. PGC-1α-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature Genetics*, 34(3):267–273, 2003.

[68] Mozilla Developer Network. Promise - JavaScript. URL `https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise`. Accessed: 2018-08-11.

[69] M. Mukaka. A guide to appropriate use of Correlation coefficient in medical research. *Malawi Medical Journal : The Journal of Medical Association of Malawi*, 24(3):69–71, Sept. 2012. ISSN 1995-7262 1995-7270.

[70] National Cancer Institute. Genomic Data Commons Data Portal. URL `https://portal.gdc.cancer.gov`. Accessed: 2018-06-28.

[71] National Electrical Manufacturers Association. DICOM Standard. URL `https://www.dicomstandard.org/`. Accessed: 2018-06-30.

[72] National Institutes of Health. The Cancer Genome Atlas Home Page. URL https://cancergenome.nih.gov/. Accessed: 2018-06-30.

[73] R. Newson. Parameters behind "nonparametric" statistics: Kendall's tau, Somers' D and median differences. *Stata Journal*, 2(1):45–64, 2002. URL http://hdl.handle.net/10044/1/27164.

[74] NumPy. numpy.intersect1d — NumPy v1.13 Manual, 2017. URL https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.intersect1d.html. Accessed: 2018-08-11.

[75] NumPy. numpy.union1d — NumPy v1.13 Manual, 2017. URL https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.union1d.html. Accessed: 2018-08-11.

[76] NumPy. Homepage, 2018. URL http://www.numpy.org/. Accessed: 2018-08-11.

[77] P. Ordóñez, M. desJardins, M. Lombardi, C. U. Lehmann, and J. Fackler. An animated multivariate visualization for physiological and clinical data in the ICU. In *Proceedings of the 1st ACM International Health Informatics Symposium*, pages 771–779. ACM, 2010.

[78] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[79] C. Perez-Llamas and N. Lopez-Bigas. Gitools: analysis and visualisation of genomic data using interactive heat-maps. *PloS one*, 6(5):e19541, 2011.

[80] phovea_data_hdf. Data provider plugin for loading data stored as Hierarchical Data Format (HDF). URL https://github.com/phovea/phovea_data_hdf. Accessed: 2018-08-11.

[81] phovea_data_mongo. Data provider plugin for loading (graph) data stored in a MongoDB. URL https://github.com/phovea/phovea_data_mongo. Accessed: 2018-08-11.

[82] phovea_data_sql. Data provider plugin for loading tabular data stored in an SQLite database. URL https://github.com/phovea/phovea_data_sql. Accessed: 2018-08-11.

[83] phovea_processing_queue. Process long-running tasks in the background using Celery. URL https://github.com/phovea/phovea_processing_queue. Accessed: 2018-08-11.

[84] phovea_processing_similarity. Processing queue plugin to compute similarities for categorical data of matrices, tables, and stratifications. URL https://github.com/phovea/phovea_processing_similarity. Accessed: 2018-08-11.

[85] phovea_server. Python server implementation of Phovea. URL https://github.com/phovea/phovea_server. Accessed: 2018-08-11.

[86] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. LifeLines: visualizing personal histories. In *Proc. CHI 1996*, pages 221–227. ACM, 1996. ISBN 0-89791-777-4. doi: 10.1145/238386.238493.

[87] C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, and B. Shneiderman. LifeLines: using visualization to enhance navigation and analysis of patient records. *Proceedings of the AMIA Symposium*, pages 76–80, 1998. ISSN 1531-605X. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2232192/.

[88] Python Software Foundation. Python.org. URL https://www.python.org/. Accessed: 2018-08-11.

[89] W. Raghupathi and V. Raghupathi. Big data analytics in healthcare: promise and potential. *Health information science and systems*, 2(1):3, 2014.

[90] Regenstrief Institute. LOINC — The freely available standard for identifying health measurements, observations, and documents. URL https://loinc.org/. Accessed: 2018-06-30.

[91] M. Rester, M. Pohl, S. Wiltner, K. Hinum, S. Miksch, C. Popow, and S. Ohmann. Mixing evaluation methods for assessing the utility of an interactive InfoVis technique. In *International Conference on Human-Computer Interaction*, pages 604–613. Springer, 2007.

[92] S. Ribecca. Stream Graph. URL https://datavizcatalogue.com/methods/stream_graph.html. Accessed: 2018-08-05.

[93] A. Rind, T. D. Wang, W. Aigner, S. Miksch, K. Wongsuphasawat, C. Plaisant, and B. Shneiderman. Interactive Information Visualization to Explore and Query Electronic Health Records. *Foundations and Trends® in Human–Computer Interaction*, 5(3):207–298, 2013. URL http://www.nowpublishers.com/articles/foundations-and-trends-in-humancomputer-interaction/HCI-039?journal=Foundations+and+Trends%C2%AE+in+Human%E2%80%93Computer+Interaction.

[94] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[95] J. Rissanen. An introduction to the MDL principle. 2005. URL http://www.mdl-research.net/. Accessed: 2018-08-14.

[96] M. P. Schroeder, A. Gonzalez-Perez, and N. Lopez-Bigas. Visualizing multidimensional cancer genomics data. *Genome Medicine*, 5(1):9, 2013. ISSN 1756-994X. doi: 10.1186/gm413. URL http://genomemedicine.com/content/5/1/9/abstract.

[97] Y. Shahar and C. Cheng. Intelligent visualization and exploration of time-oriented clinical data. In *Systems Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*, pages 12–pp. IEEE, 1999.

[98] Y. Shahar, D. Goren-Bar, D. Boaz, and G. Tahan. Distributed, intelligent, interactive visualization and exploration of time-oriented clinical data and their abstractions. *Artificial intelligence in medicine*, 38(2):115–135, 2006.

[99] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–2504, 2003.

[100] B. Shneiderman, C. Plaisant, and B. W. Hesse. Improving healthcare with interactive visualization. *Computer*, 46(5):58–66, 2013.

[101] SNOMED International. SNOMED CT - The Global Language of Healthcare. URL https://www.snomed.org/snomed-ct. Accessed: 2018-06-30.

[102] J. Stahnke, M. Dörk, B. Müller, and A. Thom. Probing Projections: Interaction Techniques for Interpreting Arrangements and Errors of Dimensionality Reductions. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):629–638, 2016. ISSN 1077-2626. doi: 10.1109/TVCG.2015.2467717.

[103] C. Stanfill and D. Waltz. Toward memory-based reasoning. *Communications of the ACM*, 29(12):1213–1228, 1986.

[104] M. Streit and N. Gehlenborg. Points of View: Bar charts and box plots. *Nature Methods*, 11(2):117–117, 2014. ISSN 1548-7091. doi: 10.1038/nmeth.2807. URL http://www.nature.com/nmeth/journal/v11/n2/full/nmeth.2807.html.

[105] M. Streit, H. Stitz, S. Gratz, K. Eckelt, D. Girardi, I. Leitner, S. Wartner, M. Wiesinger-Widi, M. Fridrik, and D. Fuchs. TourGuide: Visual Analysis of Large and Heterogeneous Scientific Workflows for Analytical Provenance. URL http://tourguide.caleydo.org/. Accessed: 2018-07-18.

[106] M. Streit, A. Lex, S. Gratzl, C. Partl, D. Schmalstieg, H. Pfister, P. J. Park, and N. Gehlenborg. Guided visual exploration of genomic stratifications in cancer. *Nature Methods*, 11(9):884–885, 2014. ISSN 1548-7091. doi: 10.1038/nmeth.3088.

[107] M. Streit, S. Gratzl, H. Stitz, A. Wernitznig, T. Zichner, and C. Haslinger. Ordino: visual analysis tool for ranking and exploring genes, cell lines, and tissue samples.

bioRxiv, 2018. doi: 10.1101/277848. URL http://biorxiv.org/lookup/doi/10.1101/277848.

[108] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. 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, Oct. 2005. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.0506580102. URL http://www.pnas.org/content/102/43/15545.

[109] V. Thada and V. Jaglan. Comparison of Jaccard, Dice, Cosine Similarity Coefficient To Find Best Fitness Value for Web Retrieved Documents Using Genetic Algorithm. *International Journal of Innovations in Engineering and Technology*, 2(4):202–205, 2013.

[110] The Cancer Genome Atlas Research Network. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455 (7216):1061–1068, 2008. ISSN 0028-0836. doi: 10.1038/nature07385. URL http://dx.doi.org/10.1038/nature07385.

[111] The Cancer Genome Atlas Research Network. Comprehensive molecular characterization of clear cell renal cell carcinoma. *Nature*, 499(7456): 43–49, July 2013. ISSN 0028-0836. doi: 10.1038/nature12222. URL http://www.nature.com/nature/journal/v499/n7456/full/nature12222.html#supplementary-information.

[112] The New York Times. The Ebb and Flow of Movies - Box Office Receipts 1986—2008 - Interactive Graphic. URL http://archive.nytimes.com/www.nytimes.com/interactive/2008/02/23/movies/20080223_REVENUE_GRAPHIC.html. Accessed: 2018-08-05.

[113] H. Thorvaldsdóttir, J. T. Robinson, and J. P. Mesirov. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. *Briefings in bioinformatics*, 14(2):178–192, 2013. ISSN 1477-4054. doi: 10.1093/bib/bbs017.

[114] H. Tipney and L. Hunter. An introduction to effective use of enrichment analysis software. *Human Genomics*, 4(3):202–206, 2010. ISSN 1473-9542. doi: 10.1186/1479-7364-4-3-202. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3525973/.

[115] Tiroler Arbeitskreis für Onkologie. Mammakarzinom: Empfehlungen zu Diagnostik, Therapie und Nachsorgeuntersuchungen in Tirol. URL http://www.tako.or.at/wp-content/uploads/Mamma_2016.pdf. Accessed: 2018-03-25.

[116] C. Tominski, J. Abello, and H. Schumann. Axes-based visualizations with radial layouts. In *Proceedings of the ACM Symposium on Applied Computing (SAC '04)*,

SAC '04, pages 1242–1247, New York, NY, USA, 2004. ACM. ISBN 1-58113-812-1. doi: 10.1145/967900.968153. URL http://doi.acm.org/10.1145/967900.968153.

[117] TypeScript. JavaScript that scales. URL https://www.typescriptlang.org/. Accessed: 2018-08-11.

[118] C. Viau and M. J. McGuffin. ConnectedCharts: Explicit Visualization of Relationships between Data Graphics. In *Proceedings of Eurographics/IEEE Conference on Visualization*, volume 31, pages 1285–1294, 2012. doi: 10.1111/j.1467-8659.2012.03121.x.

[119] M. Vijaymeena and K. Kavitha. A survey on similarity measures in text mining. *Machine Learning and Applications: An International Journal*, 3(2):19–28, 2016.

[120] S. Wagner and D. Wagner. Comparing Clusterings - An Overview. URL https://publikationen.bibliothek.kit.edu/1000011477. Accessed: 2018-04-02.

[121] T. D. Wang, C. Plaisant, A. J. Quinn, R. Stanchak, S. Murphy, and B. Shneiderman. Aligning Temporal Data by Sentinel Events: Discovering Patterns in Electronic Health Records. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 457–466, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-011-1. doi: 10.1145/1357054.1357129. URL http://doi.acm.org.ezp-prod1.hul.harvard.edu/10.1145/1357054.1357129.

[122] M. Ward, G. Grinstein, and D. A. Keim. *Interactive Data Visualization: Foundations, Techniques, and Application*. A.K. Peters, Natick, MA, USA, 2010. ISBN 978-1-56881-473-5. URL http//www.idvbook.com.

[123] V. L. West, D. Borland, and W. E. Hammond. Innovative information visualization of electronic health record data: a systematic review. *Journal of the American Medical Informatics Association*, 22(2):330–339, 2014.

[124] J. J. v. Wijk. Views on Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):1000–433, 2006. URL http://portal.acm.org/citation.cfm?id=1137498.

[125] Wikipedia. Sørensen–Dice coefficient. URL https://en.wikipedia.org/w/index.php?title=S%C3%B8rensen%E2%80%93Dice_coefficient&oldid=828477181. Page Version ID: 828477181. Accessed: 2018-08-11.

[126] W. Willett, J. Heer, and M. Agrawala. Scented Widgets: Improving Navigation Cues with Embedded Visualizations. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '07)*, 13(6):1129–1136, 2007. doi: 10.1109/TVCG.2007.70589.

[127] K. Wongsuphasawat, J. A. Guerra Gómez, C. Plaisant, T. D. Wang, M. Taieb-Maimon, and B. Shneiderman. LifeFlow: Visualizing an Overview of Event Sequences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1747–1756, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979196. URL http://doi.acm.org.ezp-prod1.hul.harvard.edu/10.1145/1978942.1979196.

[128] World Health Organization. International Classification of Diseases for Oncology, 3rd Edition (ICD-O-3), 2000. URL http://www.who.int/classifications/icd/adaptations/oncology/en/. Accessed: 2018-06-30.

[129] World Health Organization. International Classification of Diseases 10 Version: 2016, 2016. URL http://apps.who.int/classifications/icd10/browse/2016/en. Accessed: 2018-06-30.

[130] Z. Wu and M. Palmer. Verbs Semantics and Lexical Selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics. doi: 10.3115/981732.981751. URL https://doi.org/10.3115/981732.981751.

[131] D. R. Zerbino, P. Achuthan, W. Akanni, M. R. Amode, D. Barrell, J. Bhai, K. Billis, C. Cummins, A. Gall, C. G. Girón, L. Gil, L. Gordon, L. Haggerty, E. Haskell, T. Hourlier, O. G. Izuogu, S. H. Janacek, T. Juettemann, J. K. To, M. R. Laird, I. Lavidas, Z. Liu, J. E. Loveland, T. Maurel, W. McLaren, B. Moore, J. Mudge, D. N. Murphy, V. Newman, M. Nuhn, D. Ogeh, C. K. Ong, A. Parker, M. Patricio, H. S. Riat, H. Schuilenburg, D. Sheppard, H. Sparrow, K. Taylor, A. Thormann, A. Vullo, B. Walts, A. Zadissa, A. Frankish, S. E. Hunt, M. Kostadima, N. Langridge, F. J. Martin, M. Muffato, E. Perry, M. Ruffier, D. M. Staines, S. J. Trevanion, B. L. Aken, F. Cunningham, A. Yates, and P. Flicek. Ensembl 2018. *Nucleic Acids Research*, 46(D1):D754–D761, 2018. doi: 10.1093/nar/gkx1098. URL http://dx.doi.org/10.1093/nar/gkx1098.

[132] W. Zifferer. CGM G3 Timeline, 2015. URL http://www.cgm-media.at/share/detail/cgm-g3-timeline/. Accessed: 2018-03-21.

[133] S. Zillner, T. Hauer, D. Rogulin, A. Tsymbal, M. Huber, and T. Solomonides. Semantic visualization of patient information. In *Computer-Based Medical Systems, 2008. CBMS'08. 21st IEEE International Symposium on*, pages 296–301. IEEE, 2008.

[134] Z. Šulc and H. Řezanková. Evaluation of recent similarity measures for categorical data. In *Proceedings of the 17th International Conference Applications of Mathematics and Statistics in Economics. Wydawnictwo Uniwersytetu Ekonomicznego we Wrocławiu, Wroclaw*, pages 249–258, 2014.