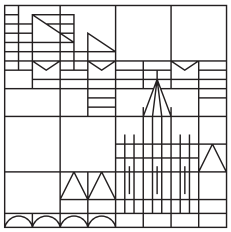
Visual Analytic Methods for Exploring Large Amounts of Relational Data with Matrix-based Representations

Dissertation zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften

vorgelegt von

Michael Behrisch

an der



Mathematisch-Naturwissenschaftliche Sektion
Informatik und Informationswissenschaft

Konstanz Juli 2016

Für Ben und Maline,
durch die mein Leben,
meine Wertevorstellungen
und Ansichten eine
neue Perspektive bekommen haben.

Danksagung

In meiner Danksagung zur Masterarbeit habe ich meiner damaligen *Freundin* Susanne mit den Worten gedankt: “Danke, dass [...] du mir eine unendlich wertvolle Stütze bist und mir mein Leben versüßt. Ich freue mich auf alles Kleine und Große was noch vor uns liegt.” Heute hat sich daran geändert, dass diese wundervolle Frau meine *Ehefrau* geworden ist und ich jeden Tag stolz darauf bin. Zudem hat sie mir zwei großartige Kinder – Maline und Ben – geschenkt, die mich daran erinnern, wie wichtig das Leben neben der Arbeit ist. Sie ist der Rückhalt und die Quelle meiner Kraft und Energie, die diese Arbeit ermöglicht hat.

Ein besonderer Dank gilt meinen Eltern und Schwiegereltern. Danke Mama und Papa für jegliche Unterstützung und den Zuspruch in schwierigen Zeiten. Danke Beate und Hans-Peter für die Denkanstöße, die uns immer wieder in eine richtige Richtung stupsen.

Ein großer Dank gilt Prof. Tobias Schreck, mein Mentor, Lehrer und Wegweiser. Durch ihn habe ich das Handwerkszeug an die Hand gegeben bekommen und er hat es geschafft sein gutes Gespür ein Stück weit weitergeben.

Ein weiterer Dank geht an Prof. Daniel Keim, der uns eine herausragende Infrastruktur zur innovativen und kreativen Arbeit ermöglicht, die seines Gleichen sucht. Danke auch der ganzen Arbeitsgruppe: großartige Kollegen, die mir sehr ans Herz gewachsen sind. Sie alle sind der Grundpfeiler für diese positive, offene und herzliche Arbeitsatmosphäre.

Abstract

Relational data is omnipresent in our computerized society and has found its way into our everyday life: Circumstances in social networks, in the transport- and public mains supply, as well as in politics or academics can be modeled with relational data.

However, together with the ever growing amount of this data type also novel analysis techniques have to be developed that are able to cope with its demanding size and complexity properties. Typical tasks include not only to visualize the often large and dense data, but also to help the analyst to understand relationships if the data set is multivariate or dynamic in nature.

Several well-known visualization techniques for relational data exist. For example, node-link diagrams display relationship attributes by drawing edges between nodes with respect to the relationship strength. The layout of nodes helps users to perceive groupings, central items or highly connected items. Matrix-based representations are another means to visualize relational data. This compact representation reaches its technical scalability limit not until all display pixels are occupied.

In this doctoral thesis we will present novel visual interactive techniques, algorithmic approaches and integrated visual analytics systems to support users in navigating and exploring large amounts of relational data. One central research objective is, amongst others, to automatically assess the interestingness of matrix views and show only potentially important matrices from a large exploration space to reduce the users' cognitive overload.

Zusammenfassung

Relationale Daten sind omnipräsent in unserer computerisierten Gesellschaft und sind in unserem täglichen Leben nicht mehr wegzudenken: Sachverhalte in sozialen Netzwerken, im Transport- und Versorgungswesen, als auch in der Politik oder im Forschungsumfeld lassen sich mit Hilfe von relationalen Daten beschreiben.

Doch zusammen mit dieser immer weiter anwachsenden Datenmenge werden auch neuartige Analysetechniken benötigt, die mit den Größen- und Komplexitätseigenschaften meistern können. Heutzutage sind die typischen Analyseaufgaben nicht nur die oftmals großen und dichten Datenmengen zu visualisieren, sondern auch dem Analysten den Einblick in die multivariaten und dynamischen Datengegebenheiten zu ermöglichen.

Es existieren einige bekannte Visualisierungstechniken für relationale Daten. Zum Beispiel, das Node-Link Diagramm stellt die relationalen Attribute durch Verbindungen zwischen Knotenpaaren dar. Das Layout der Knoten hilft dem Nutzer Gruppierungen, zentrale Schnittstellen oder logische Zusammenhänge zu erkennen. Matrix-basierte Repräsentationen sind ein weiteres Mittel um relationale Daten zu visualisieren. Diese kompakte Repräsentation erreichte erst ihre technische Skalierbarkeit wenn alle Pixel des Bildschirms gefüllt sind.

In dieser Doktorarbeit werden wir neuartige visuell-interaktive Techniken, algorithmische Ansätze und integrierte Visual-Analytics Systeme präsentieren, die den Benutzer in der Navigation und Exploration von großen Mengen an relationalen Daten unterstützen. Ein zentrales Forschungsziel ist hierbei die Interessantheit von Matrix Bildern automatisch zu quantifizieren und potentiell interessante Matrizen vom einen –möglicherweise großen– Explorationsraum zu extrahieren. Mit diesem Ansatz kann die kognitive Aus-/Überlastung des Nutzers reduziert werden.

Table of Contents

1 | Introduction

Contents

This chapter of the dissertation motivates our research with respect to matrix-based representations. Especially, we will highlight, enumerate and discuss in Section 1.1 the primary research questions that build the cornerstones of our work and this Ph.D. thesis. Specifically, we will motivate in Section 1.2.2 the need for a quality-metric and pattern-driven data exploration and summarize our scientific contributions in Section 1.3.

The main contribution of this chapter is a theoretical discussion in Section 1.2 on the usefulness of data visualizations, which we consider to be influenced from (1) the contained dataset information, (2) algorithmic processing, especially the (visual) pattern generation processes and (3) the user's analysis task at hand.

Relational data is omnipresent in our computerized society and has found its way into our everyday life. With the advent of social networking websites, such as Facebook, even new research fields have emerged that explore characteristics of relational data types. Furthermore, relational data is present in network security scenarios, in the analysis of biological experiments, and in academic research (co-authorship and citation networks) only to name a few examples. With the growing amounts of relational data the need for analysis techniques dealing with those data sets increases likewise. Typical tasks include not only to visualize the often large and dense data, but also to help the analyst to understand relationships if the data set is multivariate or dynamic in nature.

However, relational data is growing significantly in size. To take again the social networking example, the average Facebook user has 338 friendship connections and the median friend count is 200 [Cen14]. In general, visualizing relational data can be challenging, since the data is either globally or locally dense and in nearly every application scenario large in size. Several well-known visualization techniques for relational data exist. *Node-Link diagrams*, for example, display relationship attributes by drawing edges between nodes with respect to the relationship strength. The layout of nodes helps users to perceive groupings, central items or highly connected items. *Matrix-based Representations* are another means to visualize relational data. Here, N columns and M rows are displayed to show simultaneously the relationships between all items. Each pair-wise relationship is drawn at the intersection of the corresponding items' indices. This compact representation reaches its technical scalability limit not until all display pixels are occupied. A comparison on the effectiveness of the both visualization techniques has been conducted in [GFC05]. The overarching result of that user study is that matrices show to be more effective than node-link diagrams whenever the underlying data has large and dense characteristics (for all nine tasks, except of path finding). If the data has on top of that a dense aspect[1], matrices can help to answer the graph-related tasks significantly better than node-link diagrams.

In this doctoral thesis we will present novel visual interactive techniques, algorithmic approaches and integrated visual analytics systems to support users in navigating and exploring large amounts of relational data. One central research objective is, amongst others, to automatically assess the interestingness of matrix views and show only potentially important matrices from a large exploration space to reduce the users' cognitive overload.

With my studies we want contribute to the matrix visualization research by enlarging the scope to data sets that have on top of its large and dense characteristics, also multivariate and/or dynamic aspects. In case of multivariate data one matrix can be constructed for every data type. In the case of dynamic datasets one matrix can be retrieved for every time instance. In both cases, large amounts of matrices lead to both processing- and visualization challenges.

[1] Graphs with a density more an average than 0.4d

1.1 | Research Questions and Approach

Related to the motivation we are deriving several research questions, which will be described in detail in the following.

1. How can we support the exploration process for relational data with the help of matrix-based representations?
   (a) How can enhance the expressiveness and effectiveness of matrix visualizations?
   (b) Which interaction concepts help the user in exploring relational data in matrix visualizations?
2. How can we describe and quantify the interestingness of matrices wrt. its contained patterns?
   (a) How can we measure the occurrence of specific visual features (i.e., patterns) contained in matrices?
   (b) How can we derive interestingness scores depending on pattern descriptions for matrix-based representations?
3. How can we help the user in navigating and exploring large matrix spaces?
   (a) How can we compare matrices, e.g., to allow for 'more-like-this' queries?
   (b) How can we support the user in defining queries for matrix patterns?
   (c) How can we train computer systems to reflect an analyst's notion of interestingness?

The first set of research questions focuses on the *effectiveness* and *usefulness* of matrix visualizations. While the standard row-/column matrix layouting paradigm already allows to encode with every screen pixel a distinct data value –an outstanding visualization characteristic only shared with a few other visualization techniques– more sophisticated interaction and exploration mechanisms allow a visual encoding of even more information. We therefore *experimentally explored* different glyph designs for matrices that "appear" based on a semantic exploration zoom level. This semantic zoom metaphor allows the user to gain iteratively more and more insight and information during the exploration process.

The second set of research questions addresses the problem of retrieving potentially *interesting* matrix views to support the exploration of networks. For this purpose, we developed Matrix Diagnostics (or MAGNOSTICS), a conceptual framework to *evaluate empirically* the usefulness of image feature descriptors for the retrieval of matrix patterns. In spirit of related approaches for rating and ranking other visualization techniques, such as Scagnostics for scatter plots, the MAGNOSTICS feature descriptor ranks matrix views according to the appearance of specific visual patterns, such as blocks and lines, indicating the existence of topological motifs in the data, such as clusters, bi-graphs, or central nodes.

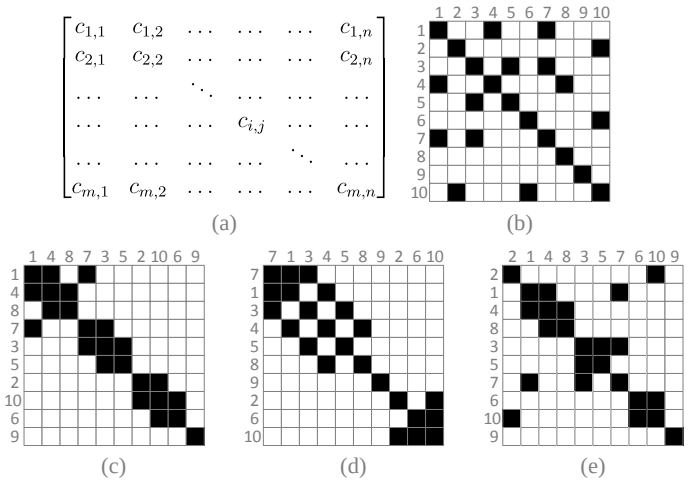


Figure 1.1 Visual matrix of numerical data (a) ordered randomly (b) and with three algorithms (c-e) revealing different patterns.

As an extension to the work of MAGNOSTICS and to contrast the approach of engineered (image)features, this thesis presents a learned feature approach, based on convolutionary neural network (cf. Section 4.6) and compares both pattern retrieval approaches with respect to their efficiency and effectiveness (cf. Section 4.7).

While the first two sets of research questions relate to patterns and the visual appearance of a single matrix, the third set of research questions focuses on the analysis of large sets of matrices and especially the pattern-driven navigation within these large view spaces. As one example, we developed the FDIVE (Feedback-Driven Interactive View Exploration), a conceptual and theoretical framework for the relevance feedback-driven exploration of large view spaces, which helps the user to intuitively define and refine his/her current notion of interest.

To be of practical use, we will present throughout this thesis several application scenarios in which our approaches help the analysts to get a better insight into their (matrix) data sets. As an example, we will show in Section 4.8 how MAGNOSTICS helps exploring the temporal evolutionary changes in brain-connectivity scans from the biological domain. Another example will be presented in 5.8 where we show how an interactive similarity steering helps to understand the specificities of denial-of-service attacks on computer networks.

Generally, our work can be subdivided into *Single Matrix Analysis* and *Multi Matrix Analysis*. But, one has to note that the e.g., a comparative analysis of multiple matrices would not be possible if we neglect single matrix aspects, such as matrix ordering. Therefore, we present in Chapter 2 theoretical considerations on patterns in matrices and more generally, the visual appearance of matrices. Specifically, we will report in Section 2.3 on

the State-of-the-Art of matrix reordering algorithms with the analytic question “Which matrix reordering algorithm tends to produce which specific matrix patterns?”.

1.1.1 | Single Matrix Analysis

Related to the question of visual quality of matrix views is how a matrix is ordered. If a matrix is ordered “appropriately” interpretable visual structures are outstanding, as Figure 1.1 depicts. We conducted a survey [Beh+16b] to describe the impact and characteristics of matrix reordering algorithms depending on the dataset’s characteristics. This helps to solve parts of the question, which matrix reordering algorithm to choose for which analysis task at hand.

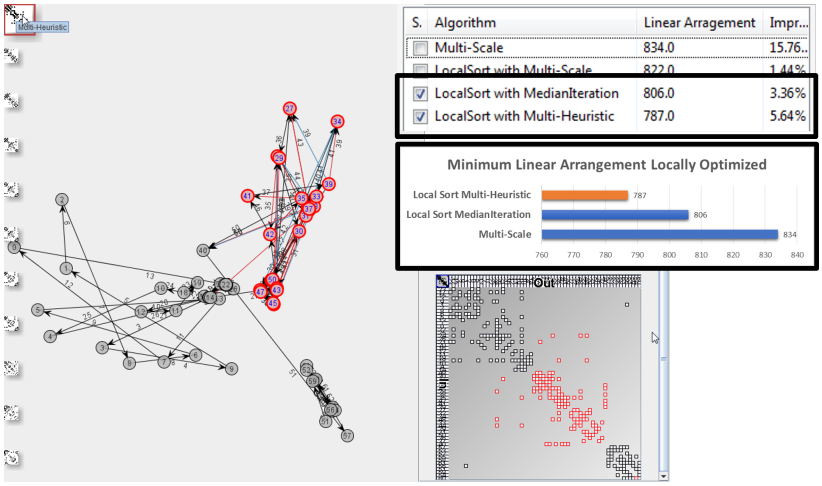


Figure 1.2 Interactive Matrix Reordering: In an interactive user-guided approach the user can steer the reordering process by invoking a localized reordering algorithm. Ordering thumbnails on the left side allow the anticipation of localized reordering results without applying the transformation to the data. Here, the user selection leads to an improvement of the linear arrangement quality measure (5.64%).

However, most matrix reordering algorithms solve an optimization problem based on predefined local or global target criteria. They are more-or-less black-box algorithms; the user has no control over results beyond the choice and parameterization of quality criteria. Due to the large search space, the algorithms use heuristics and may return a local optimum in certain circumstances. Additionally, their runtime and/or memory complexity is such that multiple runs with different parameterizations can be very time consuming. We therefore investigated means to interactively steer and guide the matrix reordering process during its progression. We therefore introduced in [Beh+14a] interactive visualizations (see: Figure 1.2) that help to improve quantitatively measurable matrix ordering criteria and the qualitative user satisfaction.

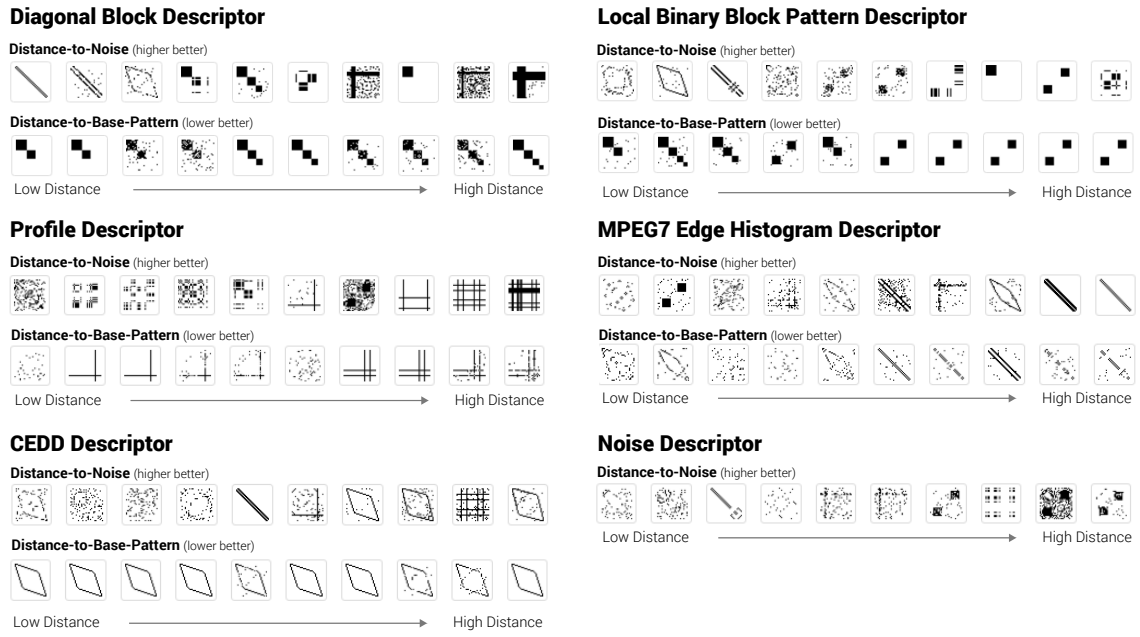


Figure 1.3 Final selection of MAGNOSTICS feature descriptors for a quantification of the primary visual patterns in matrix plots.

In line with the question of visual quality we also investigated the consensus of multiple matrix sorting algorithms following the hypothesis that if multiple sorting algorithms "agree" on local substructures these submatrices might contain interesting patterns. Hence, we presented in [Beh+13] a visual approach for the comparison of sequentially ordered (or ranked) data, such as a matrix's permutation of rows and columns. The approach relies on a small-multiple view of glyphs each of which visually contrasts a pair of rankings. The glyph in turn is defined on a radial node-link representation which allows effective perception of agreements and differences in pairs of rankings. With this visualization we are able to spotting patterns of similarity and differences in sets of orderings.

1.1.2 | Multi Matrix Analysis

The exploration and navigation in large matrix spaces is another central research focus which we subsume under the term "Multi Matrix Analysis". Therefore, we investigated methods to interactively and (semi-)automatically support users during the exploration, e.g., occurring in dynamic application scenarios. Since matrices are mostly perceived as a static visualization technique, little research has been conducted in the field of dynamic and multivariate matrix spaces. We developed, on the one hand, clustering and classification approaches and on the other hand information retrieval approaches, which support the user facilitating navigation and exploration tasks.

However, a pattern-driven exploration is not possible without measures that allow to quantitatively assess the presence or salience of matrix patterns. Quantifying patterns in visualizations typically requires heuristic feature-based approaches that respond to the (potentially) interesting structural characteristics of a visualization. These methods try to mimic human perception in that they distinguish one or more visual patterns from noise. While many feature descriptors (FDs) for image analysis exist, there is no evidence how they perform for detecting patterns in matrices. In order to make an informed choice for the primary visual patterns in matrices, we evaluate in [Beh+16a] 30 FDs, including three new descriptors that we specifically designed for detecting matrix patterns. Using a controlled benchmark data set of 5,570 artificially generated matrix images , we evaluated each FD with respect to four criteria: pattern response, pattern variability, pattern sensibility, and pattern discrimination.

As the final result of MAGNOSTICS we derived a set of six FDs that helps us to quantify the presence of matrix patterns as depicted in Figure 1.3.

In [Beh+14a] we also investigated the question: Can we develop visual analytic methods that support the user in a comparative analysis of large sets of matrices? In contrast to the image space approach of MAGNOSTICS, our approach here considers the row and/or column vectors of a matrix as the basic elements of the analysis. We project these data vectors for pairs of matrices into a low-dimensional space which is used as the reference to compare matrices and identify relationships among them. Bipartite graph matching is applied on the projected elements to compute a measure of distance. A key advantage of this measure is that it can be interpreted and manipulated as a visual distance function, and serves as a comprehensible basis for ranking, clustering and comparison in sets of matrices. We present an interactive system (see: Figure 1.4) in which users may explore the matrix distances and understand potential differences in a set of matrices. A semantic zoom mechanism enables users to navigate through sets of matrices and identify patterns at different levels of detail.

Another line of research tackles the question how computers can effectively support users in exploration tasks. This question originates from the fact that users are often confronted with the problem of identifying interesting views in which a manual exploration of the entire view space is ineffective or even infeasible. While certain quality metrics have been proposed to identify potentially interesting views, these often are defined in a heuristic way and do not take into account the application or user context. To tackle some of these challenges we introduced in [Beh+14b] a framework for a feedback-driven view exploration, inspired by relevance feedback approaches used in Information Retrieval. The basic idea is that users iteratively express their notion of interestingness when presented with candidate views. From that expression, a model representing the user's preferences, is trained and used to recommend further interesting view candidates. A decision support system monitors the exploration process and assesses the search process for convergence

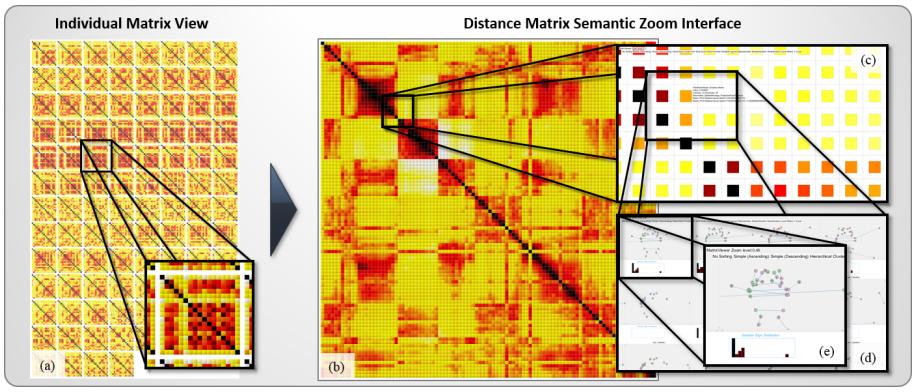


Figure 1.4 Projection-based Matrix Comparison: In a semantic zoom interface users can explore distances between matrices (a) (here: 100 matrices; ordered by time stamp). Starting from an overview distance meta-matrix (b) showing the pairwise distances between matrices, users can identify patterns (e.g. strong groups or outliers). Having found such patterns, users can investigate the impact of matrix size variations on the distance calculation (c) and steer it using a simple set of interactions (d) and (e).

and stability. We presented our approach with an instantiation of our framework for the exploration of large scatter plot spaces based on visual features and demonstrated the effectiveness by a case study on two real-world datasets.

1.2 | On the Usefulness of Data Visualizations

Usefulness and effectiveness are central keywords describing the visual quality of a visualization. A quotation that defines effectiveness stems from Mackinlay [Mac86] and says:

> Effectiveness criteria identify which of these graphical languages *[that are expressive]*, in a given situation, is the most effective at exploiting the capabilities of the output medium and the human visual system.

For the purpose of characterizing the visual quality and interestingness of matrix-based representations we are using the related term usefulness, which we define as follows:

> The *usefulness* of a visualization is influenced and bounded by three distinct characteristics: (1) the contained dataset information, (2) algorithmic processing, especially the (visual) pattern generation processes and (3) the user's analysis task at hand.

Figure 1.5 visually depicts this usefulness dependency triangle, which we will describe in detail in the following. Most generally, all major analysis tasks have the focus to retrieve certain patterns in the data. Fayyad, Piatetsky-Shapiro, and Smyth state, that "extracting

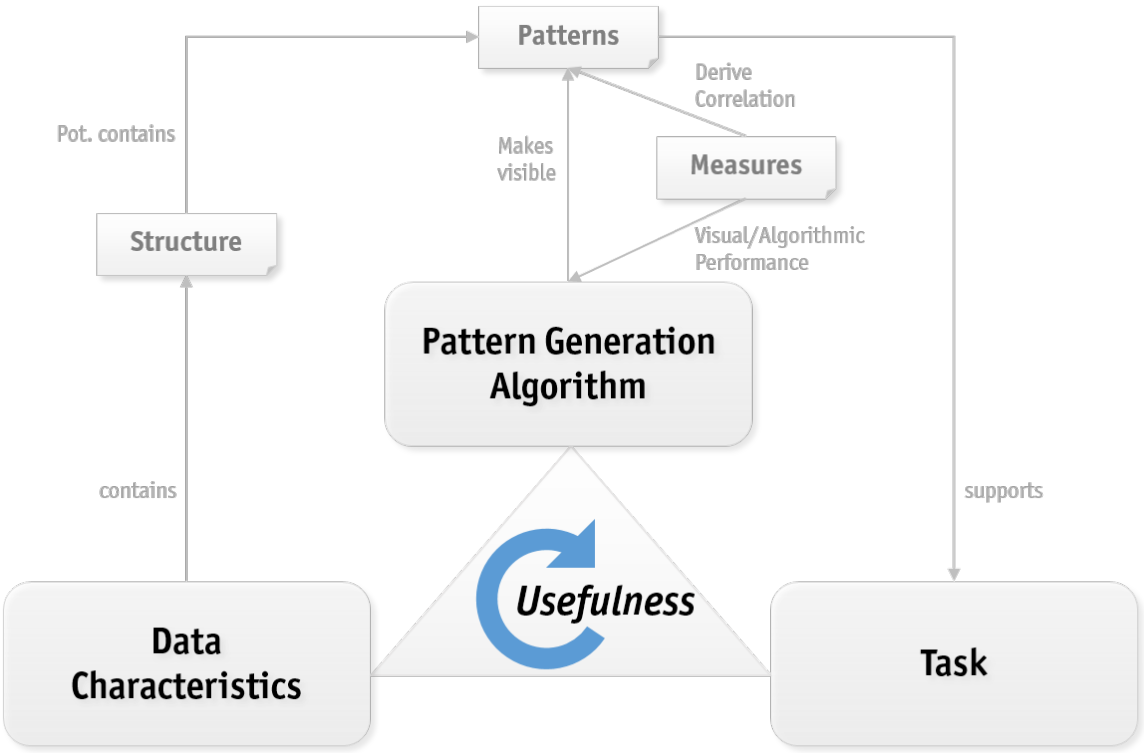


Figure 1.5 On the Usefulness of Data Visualizations: A dependency triangle.

a pattern designates fitting a model to data; finding structure from data; or, in general, making any high-level description of a set of data" [FPS96a, p. 41].

Bertin [Ber73; Ber81] developed several important ideas about the distinct levels of information contained in data displays and the user tasks –he uses the term questions— that refer to the respective levels [Ber73, p. 141]. He mentions (i) an elementary level, comprised of individual graphic elements and the task to understand their specificities; (ii) an intermediate level, for the comparisons among subsets of graphic elements and the discovery of homogeneous information parts; and (iii) an overall level, comprised of overall trends and relations. As a result, the analysis of visual patterns, esp. in matrices, is important, since these patterns can be interpreted in the user's analysis context, i.e., they relate to an analysis question and task at hand, and second, since they constitute the core information of a matrix plot, they allow the analyst to interpret and reason about their presence or salience. It is suspect to an explorative analysis to retrieve these patterns and reason about their existence or absence. Sacha et al. elaborate in [Sac+14] on the knowledge generation model for visual analytics.

Yet, the term usefulness refers to less abstract considerations. Let us imagine an analyst tries to explore the inherent information –i.e., patterns– contained in a dataset, then several cases might arise:

1. The most obvious and desired case is, that an analyst has a specific task, such as retrieving similarly performing entities or validating the hypothesis of a trend, and is able to find a visual representation of the data pattern, which allows accomplishing the current task at hand.
2. The opposite cases are more problematic and require a more careful consideration. The analyst has a specific task, but is not able to accomplish a specific task, because he/she finds no evidence. Then two circumstances could be influencing the usefulness of the visualization:
   (a) The data does not contain the expected pattern or
   (b) The pattern generation process is not able to translate this data structure into the expected visual pattern. One specific instance of this case is whenever a visual language/mapping is not expressive enough to represent the complex data patterns.
3. As opposed to the last item, the analyst could also have a specific task, which cannot be brought in line with the visualized data pattern. In this case, the pattern generation algorithm was able to capture the data characteristics, but is inappropriately chosen for the task at hand; i.e., is potentially designed for a different focus.
4. The following case is even more problematic: The data does not contain a pattern, but the pattern generation process introduces visual artifacts that lead the analyst to wrong findings, hypothesis or knowledge [Sac+14].

All problematic cases may occur for multi matrix analysis, where potentially multiple data sets, i.e., networks, need to be visualized and on top of that a multitude of matrix reordering algorithms can be applied to extract/reveal the expected visual matrix patterns.

In summary, knowing which patterns are artifacts of the algorithms and which patterns are due to the data is crucial for the data analysis and exploration process and may support the analyst to facilitate his/her analytic task at hand. However, we believe that this general considerations can also be applied to other visualizations, i.e., the usefulness of a node-link diagram is likewise influenced by the layouting algorithm that may show or hide the presence of visual patterns in the data.

1.2.1 | Matrix Analysis Tasks

In his studies on the “re-orderable matrix” Bertin [Ber73; Ber81] underlined the importance of general purpose data exploration techniques, which allow the user retrieve interesting patterns. Specifically for matrix-based representations of data tables, a simple row-/column permutation allows bringing together similar observations and variables.

Unlike most other relational data visualizations, matrices allow depicting global and local data characteristics (or visual patterns) simultaneously. Ghoniem et al. [GFC04]

found that a range of overview tasks, such as estimating the amount nodes/edges or finding the most connected node, can be answered with matrices independent of the matrix ordering. On the other hand, higher level tasks, such as summarized in the following, require an appropriate reordering of rows and columns. On top of Ghoniem's separation into lower and higher level tasks, we distinguish matrix analysis tasks based on the amount of matrices to be analyzed. Furthermore, we assume an appropriately reordered matrix (see also: Figure 2.3.3) for more information on matrix reordering).

Single Matrix Analysis Tasks

In cases where a single matrix is in the analysis focus, e.g., one snapshot in time of a social network or the correlation relationships between two variables, generally the task focus is to investigate the relationships of between one or multiple entities.

Partitioning and Grouping: One of the central tasks to be accomplished with matrix-based representations is partitioning and grouping of data items. Therefore, the general goal of most matrix reordering algorithms is to establish an ordering in which similar items will be placed close to each other, while dissimilar items will be farther apart. A matrix form that allows perceiving partitioning and grouping information is the block-diagonal form, such as depicted in Figure 1.1 (c).

Outlier Analysis: If the task is to retrieve dissimilar items, i.e., data outliers, then –by definition– a matrix reordering algorithm will separate outstanding items notably from the rest of the items. This makes a matrix-represenation to a valuable analysis tool for outlier analysis.

Depiction of High-Dimensional Structures: Although matrices are an inherently two-dimensional representation they allow perceiving complex data patterns, such as depicted in Figure 1.6. These high-dimensional data relationships are often a mixture and variation of multiple base patterns, and thus not easy to describe. But, not only high-dimensional data patterns stick out in matrices, but also circular structures are clearly visible, as already mentioned by Wilkinson in [Wil05, pp. 518].

Avoiding Clustering Artifacts: Unlike clustering, matrix reordering avoids "forcing" a vertex into a particular cluster if it does belong to this group. In other words, if you have a set of vertices that belong clearly to a group and a distinct set vertices that are close to that group but do not share the group membership then matrices will allow perceiving that progression/variation.

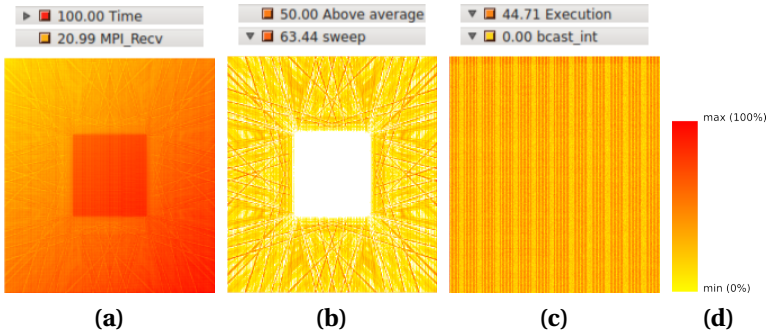


Figure 1.6 Examples of matrix views for the performance analysis in High-Performance Computing (HPC) runs on the IBM Blue Gene/P system at the Jülich Supercomputing Centre [Rüd+15a]. The matrices show virtual-topology views (2D projections of the n-dimensional computing grid) from the Sweep3d performance data set for several performance measures.

Multi Matrix Analysis Tasks

In cases where several matrices are generated, e.g., in the temporal analysis of social networks, the single analysis tasks shift to even more complex analysis scenarios.

Comparative Analysis: Whenever a degree of similarity (or distance) between matrices can be calculated, a pairwise comparison of alternatives can be facilitated. For example, in a search and retrieval task a ranking of matrices can help to understand how similar data snapshots are. Similarly, large amounts of matrices can be clustered to understand the overall data set's interrelations.

Temporal/Evolutionary Analysis: A noteworthy special case for multi matrix analysis is the temporal analysis of matrices. In these scenarios an analyst wants to retrieve evolutionary trends, outliers and –more general– temporal correlations between successively occurring matrix instances.

Pattern Analysis: The retrieval and understanding of descriptive patterns is one primary goal of multi matrix analysis. An analyst who can (semi-)automatically separate descriptive patterns from anti-patterns or task-unrelated patterns will have significantly more time to focus on the core question: "What does this pattern mean in my application context?".

1.2.2 | Need for a Quality-Metrics Driven Visual Interactive Data Exploration

The extraction of relevant and meaningful information from relational data, or more general high-dimensional data, is complex and time consuming. In that respect the notion

curse of dimensionality represents a whole set of issues encountered in the analysis of these data sets: finding relevant data attributes, selecting meaningful and descriptive dimensions, removing noise represent just a few of them. High-dimensional data visualization also carries its own set of challenges like, above all, the limited capability of any technique to scale to more than a handful of data dimensions. Relation data shares many inherent properties of high-dimensional data in that, for example, ever entity of a graph could be described by a multitude of descriptive attributes –imagine for example the name, age, size, weight, etc. in a social network scenario. Additionally, the relationships between the entities can be characterized with descriptive attributes –e.g., since when do people share a connection, how often do they communicate, is there a family relationship degree; to stay with the social network analysis example.

Researchers have been trying to solve the aforementioned analysis problems through either automatic data analysis or interactive visualization approaches. However, what is needed is an integrated *visual analytics* approach, where the machine –based on *quality metrics*– automatically searches through a large number of potentially interesting data transformations and mappings, and the user interactively steers the process and explores the output through visualizations. One specific example for the aforementioned data transformation would be the selection of a "good" matrix reordering (algorithm), such that an interpretable and useful matrix plot is generated.

This Ph.D. thesis aims at advancing the area of *quality-metric-driven visual analytics*. While many early approaches [PWR04a; BS06a; WAG05a; KC03] were focusing on the removal and detection of noise and clutter from visualizations, we are advancing the state-of-the-art to analyze, find and retrieve *visual patterns* and *anti-patterns*, such that the user may –for example– develop an intuition how patterns are distributed in the dataset. Consequently, an important research question is how to quantify the quality of data transformations and visual mappings with reference to the resulting visual (anti-)patterns. The main purpose of quality-metrics-based algorithms is to aid the user in the selection of promising data transformations and visual mappings. The algorithms search through large sets of configurations and suggest one or more solutions to the users, who evaluate them and use their insights to steer the analysis process. Since the automation aims at easing the work of the user, we have developed interactive exploration approaches that help to model reflect users' intentions.

1.2.3 | Exemplified Quality-Metric driven Exploration Workflow for Matrix-based Representations

In an exemplified usage scenario for quality-metric driven exploration a user would have to analyze a large set of matrix plots/images for the visual patterns they contain. These scenarios occur regularly, e.g. in the medical data domain, where brain activity maps are

generated in millisecond time intervals and –for example– deviations from the baseline brain activity maps are to be retrieved. Another application scenario is the performance analysis of massive parallel computing systems. One application run on a HPC cluster can produce many time-dependent performance measures in (sub-)millisecond intervals for many clusters and many computing nodes. One standard data transformation approach is to map one performance measure correspondence on a virtual 2D grid, where every row/column corresponds to one computing node in the network. This spans a large multiplicative exploration space, which needs to be assessed for the patterns it contains.

Two information visualization related approaches can come into play to explore these large exploration spaces: (1) Overview-First approaches would show the distribution of patterns in the data set at hand and (2) Explore-First approaches would show a sampling of all images and request feedback whether the chosen samples are of interest or not. In both cases it is beneficial for the user and/or the system to maintain information about the distribution of (anti-)patterns in the data. Thus, the user can be guided to interesting findings, while the information that, for example, a great share of the data set contains anti-patterns is still accessible and informative.

1.3 | Scientific Contributions and Structure of the Thesis

In the following Section we will list the primary contributions of this thesis. After enumerating the contributions we will elaborate on the impact of the individual contributions in their specific research domains.

C1: A survey of the state-of-the-art for matrix reordering approaches
Section 2.3 focuses on a description of algorithms to reorder visual matrices of tabular data and adjacency matrix of networks. The goal is to provide a comprehensive list of reordering algorithms published in different fields. We are describing the reordering algorithms in a unified manner to enable a wide audience to understand their differences and subtleties. Also we tackle the general question “What is a good reordering?”, and give practical guidance on which algorithm to choose for a specific data set at hand.

C2: Glyph-representations for enhancing the effectiveness and expressiveness of matrix-based representations
In the Chapter 3 and specifically in Section 3.4 we collect several approaches to enhance the exploration of multi-dimensional data. As a common ground, these approaches are mainly based on the combination of (complex) glyph designs and matrix representations. Specifically, we will show a sunburst glyph for the exploration of multi-dimensional numerical data, a glyph to compare rankings and orderings and a time-series statistic glyph and evaluate their usefulness in respective case studies.

C3: Visualization and Navigation approaches for large matrix spaces
We present in Chapter 3 and specifically in Section 3.5 our approaches for navigating and exploring large matrix spaces. We developed an interactive Overview-First and Detail-on-Demand interface based on the semantic zoom metaphor, which is contrasted by the Small Multiple approach.

C4: Automatic support for pattern retrieval in matrix-based representations
As one of the core chapters of this dissertation, Chapter 4 contributes with (a) engineered feature descriptor approaches, (b) learned feature extraction approaches, and (c) a data-space feature descriptor approach to the pattern-driven exploration vision. Section 4.4 presents and analyzes several new and established feature extraction approaches designed to model specific visual patterns. As an alternative to image-space measures, we present in Section 4.5 an approach that extracts structural information solely from the data space. Section 4.6 contrast engineered feature extraction approaches with a convolutionary neural network that learns in an unsupervised manner, which of the structural matrix image characteristics map to which matrix pattern. In Section 4.7, we critically compare all developed approaches with respect to their retrieval and runtime performance.

C5: Relevance feedback-driven exploration framework for large view spaces
In Section 5.7 we will present one potential solution to the interesting view problem in large view spaces. Especially, in the analysis of multivariate, high-dimensional or relational data one challenging problem is that the number of possible representations, which might contain relevant information, grows exponentially with the amount of data dimensions. In contrast to Focus+Context or semantic zoom interfaces (cf. Section 3.5), we propose a framework for a feedback-driven view exploration, inspired by relevance feedback approaches used in Information Retrieval. Our basic idea is that users iteratively express their notion of interestingness when presented with candidate views. From that expression, a model representing the user's preferences, is trained and used to recommend further interesting view candidates. A decision support system monitors the exploration process and assesses the relevance-driven search process for convergence and stability.

C6: User-guided interactive similarity steering
Adapting the similarity calculation is a core user interaction in the visual analytics pipeline and has a direct impact on the algorithm and model performance. We present in Section 5.6.1 theoretical considerations and practical implementations for a user-guided similarity adaption. Specifically, the presented approach considers the rows and/or columns of a matrix as the basic elements of the analysis. We project these vectors for pairs of matrices into a low-dimensional space which is used as the reference to compare matrices and identify relationships among them. Bipartite graph matching

Contributions	C1 Reordering STAR	C2 Glyph Designs	C3 Matrix Space Navigation	C4 Pattern Analysis	C5 Relevance Feedback	C6 Steerable Similarity	C7 Steerable Reordering
Computer Science	●●●	●○○	○○○	●●○	●○○	●○○	○○○
Data Analysis Domain	●○○	○○○	○○○	●●●	●○○	●○○	○○○
Information Visualization	●●●	●●●	●●●	●●○	●○○	●●○	●●○
Visual Analytics	●●○	●○○	○○○	●●●	●●○	●●●	●●●

Table 1.1 Mapping of relative importance of the thesis contributions to their respective research domain. Rating schema: No relevance ○○○, some relevance ●○○, largely relevant ●●○, highly relevant ●●●

is applied on the projected elements to compute a measure of distance, which can be interpreted and manipulated as a visual distance function. The projection space gives rise to a steering mechanism to control the fuzziness in inexact graph matching problems. We introduce a set of interactions to steer the similarity computation and perceive its outcome visually.

C7: User-steerable iterative matrix reordering
In line with the fundamental goal of Visual Analytics to increase the transparency of black-box algorithms, we present in Section 5.4 an approach to interactively guide and understand the complex processes in matrix reordering algorithms. We modularize the reordering process by enabling users to select groups of similar rows (or columns) and to apply local sorting algorithms to those rows. In this way, users can apply their knowledge to locally optimize the results of global reordering algorithms.

1.4 | Citation- and Contribution Clarifications Rules

As it is the accepted scientific practice and guidelines of the research community in computer science, all the major contributions of this thesis are published in journals and conference proceedings. I retain the copyright of all my publications that are used in this thesis. In order to be as transparent as possible, I state the origin of the text I produced. This serves also the goal to avoid any suspicion about plagiarism and self-plagiarism. Generally, I follow the current understanding of the citation rules as indicated by the German Research Foundation (DFG).

This resulting thesis is a trade-off between a nicely readable thesis (rewriting of all my peer-reviewed articles) and a thesis following the strictest citation rules (quoting all sections being related to a publication). I decided to put a specific focus on the content, contributions, and the reader, as I believe these to be most important.

For transparency reasons, I will state at the beginning of each chapter from which publication the content is taken from. In this thesis, I follow the subsequent citation rules:

- For each cited own publication, I list the contributions of all authors in the Reference section. In order to be transparent about the work of the co-authors I will give a *contribution clarification* for every written/co-authored paper. The intuition here is to split up between: (a) conceptualization (e.g., ideas and research approach) (b) implementation/instantiation effort (c) paper writing and (d) supervising the efforts.
- For each chapter we state the primary publication(s) from which the text and figures are adapted or taken. In the prominent boxes in the beginning of every chapter the title, all author names, the publisher and additional information are given. Individual (sub-)sections are not further marked as adapted or taken over.
- For sections that are adapted from my authored or co-authored research proposal text I state the name of the funded project and my involvement. Most of the text was written or co-authored by myself, but may have been textually revised by colleagues. These are not classical publications in the sense that a bibliographic reference can be given. However, all works can be accessed on request.
- I differentiate between three different kinds of integrating already published works into this thesis:
  - Quoted paragraphs are not written by myself and contain contributions of other authors.
  - Sections “taken from” my publications are copied and differ only in slight wording changes. These sections contain my own contributions and I did all writing myself or rephrased the sections during the paper writing process.
  - Sections “adapted from” a publication are mostly rephrased and the content has been modified. These sections contain my own contributions, but were changed to fit nicely into this thesis.
  - Every section that is “adapted” from a research funding proposal was written, co-authored or inspired by my ideas. These sections are marked by a specific footnote.

2 | Matrix-based Representations

Contents

The focus of this chapter is to collect and interlink all necessary background information for a pattern-driven analysis in matrices. Section 2.1 motivates pattern-driven analysis in general and in particular for matrix visualizations and discusses its relationship with quality-metric driven analysis for visualizations. One important section in this chapter is Section 2.1.3 where visual matrix patterns are systematically collected and discussed in relationship with their graph-theoretic interpretation and potential analysis tasks. In Section 2.2 we will give definitions for the primary terms used

throughout this work and Section 2.3 will show the current state-of-the-art for matrix visualizations.

The main contribution of this chapter is a discussion about the patterns in matrices, considerations why these patterns should be focused in matrix analysis scenarios and which pattern generation processes –in our case primarily matrix reordering algorithms– tend to produce which patterns. Therefore, we survey in Section 2.3.3 the state-of-the-art for matrix reordering and give practical guidelines when to use which algorithm.

This chapter is based on the following publication:

"Matrix Reordering Methods for Table and Network Visualization"

Behrisch, Bach, Riche, Schreck, and Fekete.

Computer Graphics Forum, The Eurographics Association and John Wiley & Sons Ltd., 2016, 35, 693-716. [Beh+16b]

Parts of the introductory Section 2.1 are adapted and/or taken from the text/figures I have written/developed for the German Research Foundation (DFG) research proposal "Transregional Collaborative Research Center 161 Quantitative Methods for Visual Computing."

2.1 | Towards a Pattern-Driven Analysis of Matrix-based Representations

In this Ph.D. thesis we aim to advance the area of *quality-metric-driven visual analytics* to analyze, find and retrieve *visual patterns* and *anti-patterns*, such that the exploration process can be enhanced or improved with this information (see also: Section 1.2.2).

The original information visualization pipeline [Shn96; CMS99b] models the main steps required for transforming data into visualizations. However, when we focus on the visualization of data patterns a practical problem arises: while the process as a whole is still valid, the number of possible combinations of the given options in each step is so high that it is impractical to interactively find the most effective ones. For example, if the original data has dimensionality n = 10 (still a quite low number) and the number of available visual parameters is k = 3 (e.g., a matrix plot with the following visual primitives: x-index, y-index, and color), the number of alternative mappings is already more than $n!/2 * |colors| = 1.814.400 * |colors|$ (the number of sequences without repetition).

In past decades, various quality metrics for data visualization have been investigated. These approaches try to assess the quality of a visualization by abstractly quantifying its information content [BTK11b]. Thus, one of their primary goals is to group and rank visualizations according to their potential task relevance. Early attempts to calculate quality metrics on top of visualizations can be traced back to the work of Tufte [TG83], where he proposed metrics such as the *data-ink ratio* and the *lie factor*, which optimize the use of the visualization space and reduce the distortions that visualizations may introduce. Later in 1997, Brath proposed a rich set of metrics to characterize the quality of business visualizations [Bra97] and, around the same period, Miller et al. advocated the use of visualization metrics as a way to compare visualizations [Mil+97].

Many different aspects of visualizations have been considered for measuring the quality of visualizations. Ware et al. [War+02] worked on cognitive measurements of graph aesthetics and defined a novel set of metrics for graph drawing. In the visualization community, several papers propose some form of quality measures. Examples are measures for clutter reduction in visualizations [PWR04a] and [BS06a], graph-theoretic measures for scatter plot matrices [WAG05a], and metrics based on class decomposition in linear projections [KC03]. Quality metrics specifically for the context of clutter reduction and visualization scalability were applied in [JJ09]. In recent years extensive research has been conducted in the field of visual analytics, which strives to combine computational methods and visualizations (see [Kei+10b] for an overview). Quality metrics are one promising possibility of such a combination as demonstrated in studies on pixel visualization techniques (e.g., [Kei00], [SSK07], [Kei+10a]) and high-dimensional data visualizations (cf. [Tat+11c], [Tat+10]). Also, quality metrics for high-dimensional data visualizations have been studied extensively. In two separate papers [Tat+11c; Tat+10] Tatu et al. compare the

quality metrics for scatter plot visualizations and propose automatic analysis methods to extract potentially relevant visual structures from a set of candidate visualizations. Sips et al. [Sip+09a] introduce a measurement for scatter plot ranking with classified and unclassified data. They propose two additional quantitative measures, one based on the distance to the cluster centroids and another based on the entropy of the spatial class distributions. Dasgupta et al. investigated the quality metrics for parallel coordinates [DK11]. Furthermore, prior studies showed how visualizations can support feature selection and optimization in 3D models [SFK08] or exploration of chemical compounds [Str+12a].

Many prior papers categorize existing work in the visualization area. To name just some recent ones, Tory and Möller [TM04] provide a taxonomy to describe scientific visualization and information visualization under the same structure. Ellis and Dix propose a clutter reduction taxonomy for a large number of existing clutter reduction techniques [ED07a]. Segel et al. [SH10] identify common design patterns using a large number of story telling visualizations. Bertini et al. [BTK11a] investigate overview and systematization results of quality metrics in high-dimensional data. All these prior works organize aspects of data visualization by starting with a detailed analysis of the prior work.

Since our proposed research framework for a pattern-driven analysis of matrix-based representations (as described in Section 2.1.4) is related to the standard visualization pipeline, we briefly discuss existing data processing pipelines. The standard information visualization pipeline has been presented by Card et al. [CMS99a] and is widely accepted in the community. This pipeline includes four data stages: *raw data*, *table data*, *visual structures*, and *views* to transform the data. Chi proposes a new way to taxonomize information visualization techniques by using the Data State Model [Chi00], which is largely based on the information visualization pipeline. This model classifies visualizations according to how they use the operators in the pipeline. The KDD pipeline [FPS96b] developed in the early nineties describes the data processing steps in several stages: *selection*, *preprocessing*, *transformation*, *data mining*, *interpretation/evaluation*, leading to a final stage of *knowledge generation*.

2.1.1 | Quality Metrics Derived from Image Space.

While quality metrics derived from the data space deal with data characteristics, such as (statistical) noise or cluster properties, quality metrics derived from the *image space*try to reflect the human pattern recognition process. *Image-space quality metrics* work with the assumption that the algorithm selects what the user would choose as interesting if he/she was able to visually inspect the whole set of transformations. Hence, *visual-quality analysis* aims at partially substituting human vision with image processing algorithms by closely matching the algorithm's results with the users' perception. Initial work to validate this approach, has been conducted to study the relationship between what the user sees and what the machine selects in [Tat+09].

Visual-quality analysis –and image-space quality metrics– can be used to reduce the vast exploration space size as it typically exists in relational and high-dimensional data analysis. Bertini, Tatu, and Keim [BTK11b] structure visual-quality analysis process into three distinct assessment and processing steps: (1) Creating alternatives, (2) Evaluating alternatives and (3) Producing a final representation. In the first step the system creates alternatives, which can be derived based on the specific application at hand. This can be different data subsets, mappings or views. In most of our cases, we vary the matrix reordering algorithm on one graph to take advantage of the different patterns they are able to reflect. In the second step these (potentially virtual) alternatives are evaluated by computing a measure of their information content. The third step comprises an analysis of the produced visual-quality scores. This can be the mere ranking of the alternatives, but could also incorporate more sophisticated data mining processes, such as clustering or classification. In this thesis we evaluate the produced alternative wrt. the contained visual patterns. In a visual analytics driven scenario, the user can interact with the process by setting parameters or by evaluating the resulting views.

Several papers have been published –mostly in the context of high-dimensional data analysis– that use image-space quality metrics as a way to *reduce* the search for interesting subspaces. The visualization of high-dimensional data is a beneficial study object for image-based quality metrics, because the number of dimensions that can be displayed in a visual representation at once is very limited. Well-known visualization techniques, such as parallel coordinates or scatter plot matrices, reach their limit as the number of dimensions exceeds 10 to 15. Accordingly, many projections –typically 2D– for the same dataset have to be rendered and evaluated. The basic idea behind the use of quality metrics is to let the system analyze this larger number of low-dimensional subspaces and to choose only those that contain interesting patterns.

Tatu et al. and Albuquerque et al. introduced the use of quality metrics with scatter plots and sampling [Tat+10; Alb+09a] and discussed their broader use in a number of publications. In a recent publication [Tat+12c], several metrics and algorithms were introduced to identify interesting subspaces in scatter plots and parallel coordinates. Bertini et al. classified quality metrics according to several factors, among others the applied visualization technique and their purpose [BTK11b]. The majority of quality metrics is designed for scatter plots or parallel coordinates. Histograms can be evaluated by the Rank-by-Feature framework [SS04; SS05]. Quality metrics designed for pixel-based visualization techniques (i.e., each data point corresponds to a pixel), particularly for JigSaw maps, are for example the Noise-Dissimilarity measure of Albuquerque et al. [Alb+10], or the entropy and standard deviation that are used in the Pixnostics framework of Schneidewind, Sips, and Keim [SSK06]. Both have the purpose of finding a clutter-free visual mappings/transformations.

2.1.2 | Pattern-Driven Visual Analytics

Most of the works in the field of quality-metric-driven visual analytics are focusing on the extraction and quantification of clutter or noise in a visualization. Yet, in this work we are enlarging the scope of separating inappropriate –since noisy– visualizations from the useful visualizations by describing the information content with respect to the *visual patterns* they contain. With this extension to the standard quality-metrics-driven exploration we allow analysts to explicitly state his/her task in the beginning of the analysis process and retrieve only the information that fits his/her needs.

More specifically, we are broadening the scope of the initial work by investigating novel and a significantly larger amount of engineered image processing techniques and validate the usage of learned feature extraction mechanisms for a pattern-driven visual analytics process. We constructed a ground truth data set of 5570 artificially generated matrix images to validate our pattern assessment techniques for matrix-based representations. This ground-truth data set allows us to numerically quantify the performance of each investigated image analysis method and helps us to tackle part of the research question: "Which quality metric to use for which task at hand?" (c.f. Chapter 4).

2.1.3 | Structures and Visual Patterns in Matrix Visualizations

Core of analysis goal of a pattern-driven visual exploration is to determine and quantify the occurrence of visual patterns. Therefore, we are focusing in this section on the central question: "Which visual patterns and anti-patterns should be retrieved for matrix-based representations?" and "How do these structural features relate to the human perception?".

As already noted in the Introduction Chapter (Section 1.1.1) visual patterns in matrices are "generated" by choosing an appropriate permutation of rows and columns (*matrix ordering*). Figure 1.1 shows the same data set, but with distinct orderings that differ in their visual characteristics; distribution of cells, number of blocks, size of blocks, clarity of blocks. Each of the orderings highlights or hides certain characteristics of the underlying data set such as the number of clusters, similar elements, and outlier. The matrix in Figure 1.1(b) shows an equal distribution of cells which implies no particular structure in the data (random data). Figure 1.1(c) shows a continuous band along the matrix diagonal, with a single isolated block. Finally, Figure 1.1(d), (e) show two isolated blocks, but at different corners of the matrix.

Formally, reordering an undirected network G consists in computing one permutation π from the set S of all possible row-/column permutations that maximizes or minimizes an objective function $q(\pi, G)$, such that:

$$\arg\min_{\pi \in S} q(\pi, G) \tag{2.1}$$

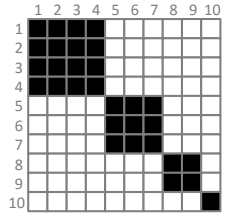
For example, q can compute the sum of distances $d(v_i, v_j)$ between vertices according to the order π; Equation 2.1 would find $\pi \in S$ that minimizes this sum.

A brute-force approach to return an optimal solution for the permutation problem on a symmetric matrix would require $n!$ computations, which renders impractical when n gets large. Since a directed network requires two permutations, one π_r for the rows and one π_c for the columns, a brute-force approach would actually require $n! \times m!$ computations.

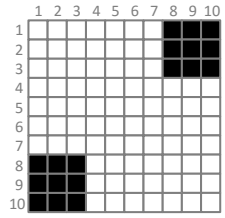
In addition, there is no consensus of an objective function q in the reordering literature. Therefore, we cannot understand the reordering problem as a pure optimization problem, and need to consider reordering algorithms according to the structures they reveal visually.

Therefore, the goal of matrix reordering is to make *visual patterns* emerge, which represent data properties of the underlying network. To understand why this is possible, it is essential to realize that the order of matrix rows and columns can be freely changed without changing the data in the matrix.

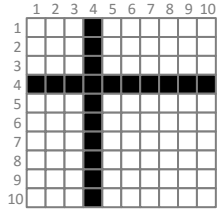
Extending Wilkinson's [Wil05] and Mueller et al. [MML07] work, we list below main *visual patterns* in matrix-based representations, along with their graph-theoretic interpretations.



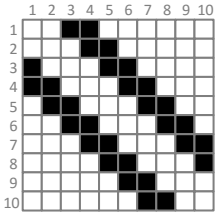
Block Pattern (*P1*): Coherent rectangular areas appear in ordered matrix plots whenever strongly connected components or cliques are present in the underlying topology. The figure shows 4 disconnected cliques (complete sub-graphs) containing 4, 3, 2, and 1 vertices. Mathematically, these matrices are called *block-diagonal matrices*.
Block-diagonal forms are central in the analysis of matrices, since they directly relate to *partitioning* and *grouping* tasks of the data. Blocks visually represent that their contained vertices share a similar connection characteristic. In a network analysis scenarios these blocks would be referred to as cohesive groups or clusters. Clear block patterns help counting clusters, estimate cluster overlap and identify larger and smaller clusters. Furthermore, many networks show block patterns with missing cells, meaning that clusters have missing connections (i.e., holes) or being connected to other clusters (i.e., off-diagonal dots).



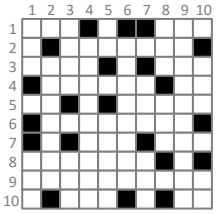
Off-diagonal Block Pattern (*P2*): Off-diagonal coherent areas correspond to either sub-patterns of a *block pattern* or relations in a bi-graph. In the first case, the off-diagonal pattern would be visible in addition to the previous block pattern, and show connections between cliques. Off-Diagonal blocks map to the user task of understanding how groups/entities are connected. In the graph task taxonomy of Lee [Lee+06], this pattern would allow approaching *adjacency assessment* and *overview* tasks. In the case of a bi-graph, the off-diag. pattern would show consistent mappings from e.g., a set of authors to a set of documents. Just like the diagonal block pattern, off-diagonal blocks can contain missing connections.



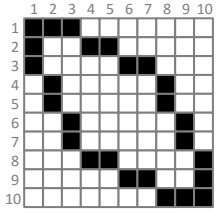
Line/Star Pattern (*P3*): Continuous horizontal and vertical lines are present in matrix plots if a vertex is strongly connected to several distinct other vertices.
This pattern helps the analysts to understand and reason on the general *connectivity* aspects within the network. In a network analysis scenario lines would refer to hubs, i.e., nodes with many connections. The length of a line thereby indicates the number of connections (node degree).



Bands Pattern (*P4*): Off-diagonal continuous lines refer to paths and cycles, or meshes in a network. They represent a set of vertices with a few connections to other vertices. Visually, this pattern can regarded as a special case of the line pattern and is useful whenever (adjacency) relationships and *connectivity* aspects are in the user focus [Lee+06]. In a network analysis scenario bands would refer to connection paths and transition chains, where the width of the band visually depicts how many distinct paths could be used.



Noise Anti-Pattern (*A1*): Noise (also called *salt-and-pepper*) is the classic anti-pattern for a matrix plot. It can be found whenever the row-/column ordering is not able to reveal the underlying graph topology or if simply no structure exists. However, submatrices can occur to be noisy, even if other submatrices show structure. Moreover, a matrix can be noisy or show structure on different levels: locally, i.e. for subgraphs (submatrices), and globally, i.e. the entire graph (matrix).
The distinction between anti-patterns and the mentioned (interpretable) visual patterns helps to develop an *overview* about the topological aspects of the network at hand.



Bandwidth Anti-Pattern (*A2*): Bandwidth- or sparsity patterns visually group non-zero elements (connections) within an enclosure around the diagonal. This pattern adds little interpretation asset to the matrix plot if the inner part of the bandwidth enclosure reveals no structure. Bandwidth patterns are typical for breadth-first search algorithms where the outer border depicts the stop criterion of the enumeration (cf. Section 2.3.3).
However, similarly to the noise anti-pattern, bandwidth patterns allow to reason on the absence of (expected) topological aspects and facilitate thus *overview* and *exploration* tasks [Lee+06].

Since any graph motif has a corresponding visual pattern in a visual matrix, we only describe the most important ones above. Real world graphs exhibit a mixture of overlapping patterns appearing at different scales. Hence, the visual patterns we describe are not always clearly discernible (Figure 1.1) and may appear merged together. Reordering algorithms take into consideration different aspects of the topology, inducing different pat-

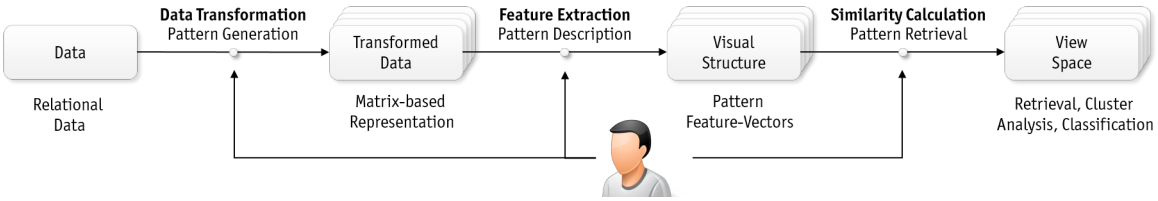


Figure 2.1 Research Framework for a Pattern-Driven Exploration of Matrix-based Representations.

terns. Others directly optimize for specific patterns. Note that several of these algorithms however fail to reveal any pattern or introduce visual artifacts.

2.1.4 | Research Framework for Pattern-Driven Exploration of Matrix-based Representations

To achieve our central goal of establishing a pattern-driven analysis of matrix-based representations we follow a research methodology as outlined in Figure 2.1 and described in the following. We motivate each of the steps and outline the main involved research challenges.

Data Transformation (Pattern Generation): The primary data transformation for matrix-based representations is to change the ordering of the rows and columns. To understand and validate the impact of this pattern generation process on the analysis outcome, we have surveyed the state-of-the-art of matrix reordering algorithms. While one research goal of the survey was to provide a comprehensive list of reordering algorithms along with a description of the algorithms' characteristics, another core research focus was to discuss and outline potential research directions. Although excerpts of the survey are presented throughout of this Ph.D. thesis, which also undermines the importance of the work for this thesis, the core considerations are presented in Section 2.3.3.

Feature Extraction (Pattern Description): The next step is the extraction of features from the visual representations as the basis for similarity computation. In general, two solution spaces can be feasible to compute descriptive features: (i) Data-space features are considering among others statistical distributions, clustering coefficients, or graph-theoretic measures to describe the relational data and have the advantage to abstract from a potentially mis-interpretable matrix reordering and (ii) Image-based features are considering solely the visual representative (matrix plot) of the relational data, which has the advantage that the feature engineering process may be understood as a modeling of the human's visual perception. Most existing image features have been proposed for retrieval of real-world images or the detection of real-world objects or scenes in images. In one of the core chapters of this thesis (Chapter 4) we systematically explore which existing

image features are applicable for retrieval of patterns in matrix-based representations. We therefore compare existing image features for their applicability, but also define new feature extractors especially designed and suitable for matrix plots.

A main research goal is to understand how image analysis methods can be applied or modified for the goal of feature extraction from image abstractions of complex data.

Similarity Calculation (Pattern Retrieval): Indispensable for an analytic reasoning in large matrix spaces are similarity functions, which measure the degree of similarity (or distance) between two given matrices. Similarity functions are the basis for any ranking method (for search tasks), or for clustering data by similarity or assigning descriptive labels in a classification task. A core scientific challenge is that the definition of similarity functions for a pattern-driven analysis can be complex and even subject to change during the analysis process. Even more challenging is that the knowledge of the user and/or the application context may influence the notion of similarity. To tackle some of the mentioned research questions, we are incorporating ideas from the Visual Analytics domain, where the user may interactively steer the similarity calculation algorithms to reflect the current needs. Chapter 5 presents these approaches and showcases their usefulness in application scenarios.

Application Design and User Interaction; Pipeline Extension towards VA: We study the above mentioned pattern generation, feature extraction and similarity steering approaches in the context of prototype applications and integrated prototype modules. As an example, we developed the "Matrix Projection Explorer", e.g., depicted in Figure 5.11, which serves as the basis for the projection-based similarity calculation between matrices (cf. data-space feature descriptor), ranking and clustering applications and ranking/ordering comparison visualizations (see: Section 3.4.2). Similarly, the user-guided matrix reordering approach (presented in Section 5.4), as well as an early prototype for MAGNOSTICS (image-space driven feature extraction for matrix patterns; see Section 4.4) are incorporated as modules in the Matrix Projection Explorer. Coupling these research ideas –or prototype implementations thereof– within one infrastructure allows establishing a *matrix analysis suite*, in which findings from one module may be transferred to the next module, thus contributing to the Visual Analytics idea. As an example, the findings from an initial ranking may be used in a user-driven feedback loop to adjust the matrix reordering process and/or the feature description process, which in turn allows to gradually improve the ranking results.

However, also a range of other less coupled prototypes where developed. For example, the MAGNOSTICS feature descriptor stands for his own, but can be integrated in retrieval applications (showcased in Section 4.8) for a pattern retrieval task, but also can be used as

a quality function for matrix reordering thus allowing a matrix reordering algorithm to be steered towards a specific analysis pattern.

2.2 | Background and Definitions

We introduce the concept of a matrix based on the more formal definition of graphs. Graphs are an accepted and well studied subject in computer science. Interestingly, many problems from graph visualization directly relate to the problems and challenged described here in this thesis (see also: Section 1.1).

A *graph* G is a couple (V, E) where V is a set of vertices, and E is a set of edges where:

$$\begin{aligned} V &= \{v_0, \cdots, v_n\}, \\ E &= \{e_0, \cdots, e_m\}, e \in V^2 \end{aligned} \tag{2.2}$$

A *directed graph* is a graph where the two vertices associated with an edge are considered ordered. An *undirected graph* is a graph where the vertices associated with an edge are not ordered.

We use the term *network* to describe the graph topology as well as attributes associated with vertices (e.g. labels), and attributes associated with edges (e.g. weights). Most networks used in this survey have names associated with vertices, and positive weights associated with edges. A weighted graph G_W adds a weight function $w(e)$ to G so that:

$$w(e_i) = w_i, \text{with } w_i \in \mathbb{R}^+ \tag{2.3}$$

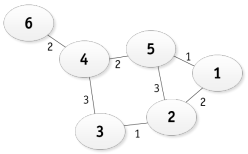
An *ordering* or *order* is a bijection $\varphi(v) \rightarrow i$ from $v \in V$ to $i \in N = \{1, \cdots, n\}$ that associates a unique index to each vertex. A network usually comes with an arbitrary ordering reminiscent of its construction or storage. We call that order the *initial order* noted $\varphi_0(v)$ to distinguish it from a computed order. A transformation from one ordering to another is called a *permutation* π. Formally, a permutation is a bijection $\pi(x) \rightarrow y$ such that:

$$\pi(x_i) = y_i, (x, y) \in N^2 \text{ where } y_i = y_j \Rightarrow i = j \tag{2.4}$$

It is usually implemented as a vector containing n distinct indices in N. We call S the set of the $n!$ possible permutations for n vertices. A permutation can also be represented as a $n \times n$ matrix P with all entries are 0 except that in row i, the entry $\pi(i)$ equals 1.

Alternatively to the representation by a tupel of sets (V, E), a graph can be represented by different matrices.

An *adjacency matrix* of a graph G is a square matrix M where the cell $m_{i,j}$ represent the edge (or lack of) for the vertices v_i and v_j. It is equal to 1 if there is an edge $e = (v_i, v_j)$ and



$$\begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 2 & 0 & 0 & 1 & 0 \\ 2 & 0 & 1 & 0 & 3 & 0 \\ 0 & 1 & 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 & 2 & 2 \\ 1 & 3 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 2.2 A simple labeled graph, its adjacency matrix, its weighted adjacency matrix, and its degree matrix

0 otherwise. When the graph is weighted, $m_{i,j}$ represents the weight (for clarity purposes, we restricted weights to be strictly positive in Equation 2.3).

$$M = \begin{bmatrix} m_{1,1} & \cdots & m_{1,n} \\ \vdots & \ddots & \vdots \\ m_{n,1} & \cdots & m_{n,n} \end{bmatrix} \tag{2.5}$$

Another less common possibility, since lossy, representation of a graph is to denote only the vertex degrees: The *degree matrix D* is defined as:

$$D = diag(degree(v_1), \ldots, degree(v_n)).$$

The *Laplacian matrix* representation, also known as Kirchhoff matrix or admittance matrix can be formulated with the help of the aforementioned equations:

$$L = D - A.$$

This matrix form has applications in spectral clustering approaches for graphs and spectral reordering methods for matrices, where this representation is used to find a partitioning of the underlying data (see also: Section 2.3.3).

A *bipartite-graph* or *bi-graph* is a graph $G = (V_1, V_2, E)$ where the vertices are divided into two disjoint sets V_1, V_2, and each edge e connects a vertex in V_1 to a vertex in V_2:

$$V = V_1 \cup V_2, V_1 \cap V_2 = \emptyset \text{ such that } e \in E = V_1 \times V_2 \tag{2.6}$$

The adjacency matrix of a bi-graph is generally rectangular, composed of V_1 in rows and V_2 in columns to limit empty cells. We consider a general *data table*, such as data presented in spreadsheet form, a valued bi-graph. A classic example of a bi-graph is a document-author network with a single relation *is-author* connecting authors to documents. The adjacency matrix of such bi-graph includes authors in rows (respectively in columns) and documents in columns (respectively in rows), a value of 1 marking the authoring relationship, and a value of 0 otherwise.

2.2.1 | Related Concepts

In addition to the previous definitions and notations, this work frequently bridges graph concepts with linear algebra concepts. Since readers might not be familiar with these relationships, we summarize them here, introducing concepts often used in this thesis.

Adjacency matrices typically bridge graph theory and linear algebra, allowing the interpretation of a graph as a multidimensional (n-dimensional for n vertices) linear system, and vice-versa. We list below several properties of networks when considered as adjacency matrices.

- When encoded as an adjacency matrix, a vertex becomes an n-dimensional **vector of edges** (or edge weights). When the network is undirected, the matrix is **symmetric** and the vectors can be read horizontally or vertically. Otherwise, two vectors can be considered: the vector of *incoming edges*, and the vector of *outgoing edges*.
- Since vertices are vectors, a **distance** measure $d(x, y)$ can be computed between two vertices $(x, y) \in V^2$ (or a *similarity* or *dissimilarity* measure $s(x, y)$). For example, the Euclidean distance L_2: $d(x, y)$ between vertices x and y is:

$$L_2(x, y) = \sqrt{\sum_{k \in [1,n]} (x_k - y_k)^2} \tag{2.7}$$

- Several reordering algorithms use a **distance matrix** (or *similarity matrix*) as input, which is a symmetric positive definite matrix D containing the pairwise distances between multiple vectors. From the $n \times n$ adjacency matrix of an undirected graph, one symmetric distance matrix can be computed of size $n \times n$. From a general n-rows $\times$ m-columns matrix, two distance matrices can be computed: one of size $n \times n$ for the rows (m-dimensional vectors we will call A), and one of size $m \times m$ for the columns (n-dimensional vectors we will call B). A distance matrix is always symmetric and positive (it is *positive-definite* mathematically speaking).
- A particularly important distance matrix is the **graph distance matrix**, which contains the length of the shortest path between every pair of vertices for an undirected graph. Note that a distance matrix or more generally a positive-definite matrix can also be interpreted as an adjacency matrix of a weighted undirected graph. Note also that any symmetric matrix can be interpreted as an adjacency matrix of a valued undirected graph (a graph where each edge has an associated value).
- From any undirected graph, or positive-definite matrix, many graph measures can be computed. These can serve as objective functions to minimize or as quality measures of reordering algorithms. We describe three key measures below: *bandwidth*, *profile*, and *linear arrangement*.

  Let us call $\lambda(u, v)$ the length between two vertices in G, given a one-dimensional

$\lambda(u, v)$ the length between two vertices in G, given an alignment of the vertices

  φ: $\lambda((u, v), \varphi, G) = |\varphi(u) - \varphi(v)|$.

Bandwidth BW is the maximum distance between two vertices given a order φ.

$$\mathrm{BW}(\varphi, G) = \max_{(u,v)\in E} \lambda((u,v), \varphi, G) \tag{2.8}$$

Intuitively and visually, when looking at the adjacency matrix of an undirected graph (a symmetric matrix), the bandwidth is the minimum width of a diagonal band that can encloses all the non-zero cells of the matrix. A small bandwidth means that all the non-zero cells are close to the diagonal. Therefore, a quality measure is MINBW, the *minimum bandwidth* of a graph $\mathrm{MINBW}(G) = \arg\min_{\varphi^*}(\mathrm{BW}(\varphi^*, G))$. Note that there can be multiple different orders that achieve that same minimum bandwidth.

Profile PR is:

$$\mathrm{PR}(\varphi, G) = \sum_{u\in V} \left(\varphi(u) - \min_{v\in\Gamma(u)} \varphi(v)\right) \tag{2.9}$$

where $\Gamma(u) = \{u\} \cup \{v \in V : (u,v) \in E\}$. Intuitively and visually, the profile is the sum, for each column i of the matrix, of the "raggedness": the distance from the diagonal (with coordinates (i,i)) to the farthest-away non-zero cell for that column (with coordinates (i,j)). It is a more subtle measure than the bandwidth because it takes into account all the vertices and not only the vertex with the largest length. The *minimum profile* is $\mathrm{MINPR}(G) = \arg\min_{\varphi^*}(\mathrm{PR}(\varphi^*, G))$.

Linear arrangement LA is the sum of the distances between the vertices of the edges of a graph:

$$\mathrm{LA}(\varphi, G) = \sum_{(u,v)\in E} \lambda((u,v), \varphi, G). \tag{2.10}$$

It is an even more subtle measure than the profile since it takes into account all the edges. The *minimum linear arrangement* is therefore formally defined as: $\mathrm{MINLA}(G) = \arg\min_{\lambda^*}(\mathrm{LA}(\varphi^*, G))$.

- In the context of matrix reordering several data "modes" are to be distinguished: (i) a *two-way one-mode* data set describes a matrix, which has columns and rows (two-way), but only represents one set of objects (one-mode). For example, symmetric dissimilarity matrices are of the form two-way one-mode. (ii) *two-way two-mode* data, such as in general non-negative matrices, represent two sets of objects. For two-way two-mode an optimal order of columns can depend of the order of rows and vice versa or it can be independent, i.e., allowing for breaking the optimization down into two separate problems, one for the columns and one for the rows [HHB08].

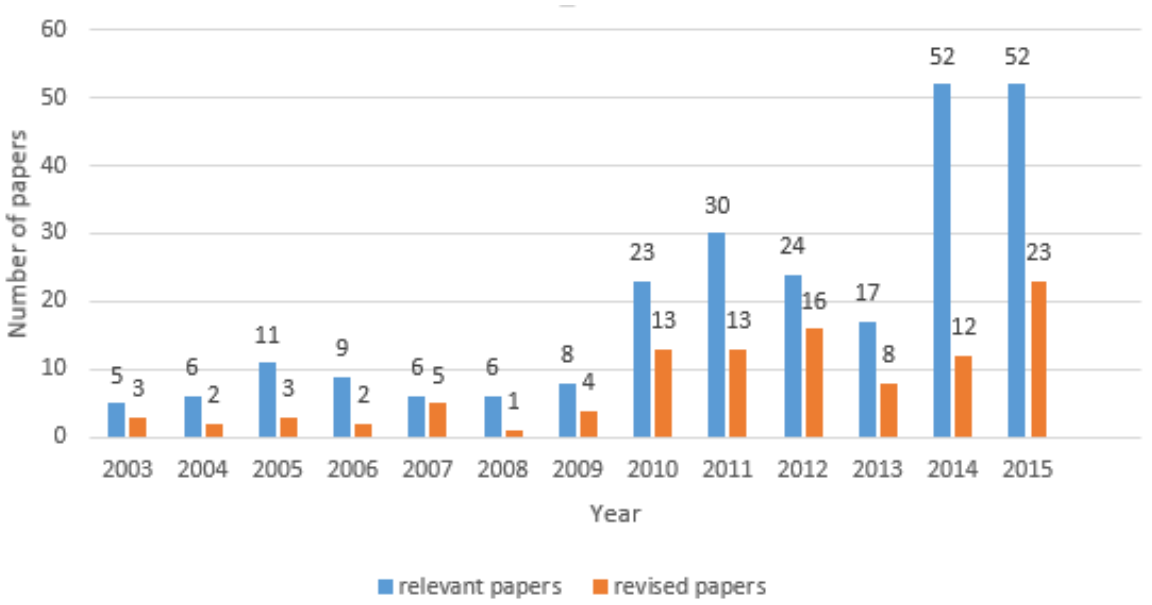


Figure 2.3 Distribution of papers using matrix visualizations as primary or meta analysis visualizations. Blue bars indicate the presence of matrix visualizations within the paper, orange bars present the usage of matrix for analytic purposes.

2.3 | State-of-the-Art for Matrix-based Visualizations

Matrix representations become increasingly applied in data visualization. This is not at least due to their primary advantage over node-link representations that they do not suffer from the occlusion for dense data sets [GFC04].

Today, over 100 papers can be found[1] showing either matrices as their primary- or as a meta visualization to support the analytic process. Matrices are used for the visualization of graphs and networks (*adjacency matrix*, e.g., [HF06; Bez+10a], similarity and correlation in data (*similarity matrix*, e.g., [Foo99; Beh+14b]), or as graphical representations for quantitative table data (*attribute matrix*, e.g., [Ber73; IML13; PDF14]).

In the following we will study the *modality* of matrix visualizations, their differences in the level of *algorithmic support* and their application/usage area, while a special focus lies on interaction concepts (see: Figure 2.4).

The visual appearance of matrix-based representations can be structured along two primary axis:

Matrix Layout Visualization approaches that modify, enhance or change the standard row-/column based layouts.

Cell Appearance Approaches that modify, enhance or change the standard uniform cell appearance.

[1] EuroVis and IEEE Vis publications between 2003 until 2015

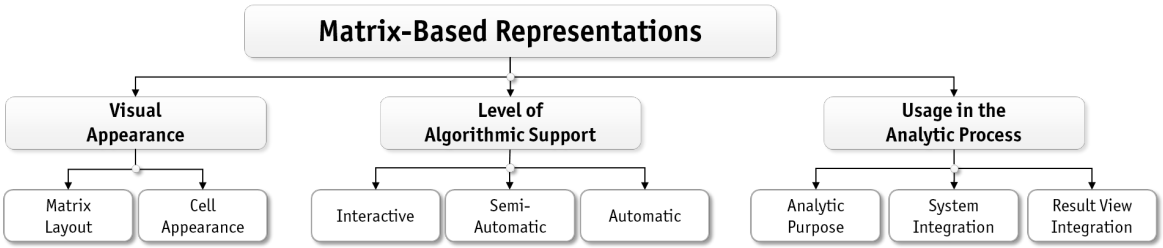


Figure 2.4 State-of-the-Art Categorization for Matrix-Based Representations: We are investigating (a) the visual appearance of matrix plots, (b) the level of algorithmic support users get for their analytic questions and (c) the usage scenarios in which matrices are used.

In the following, we will give examples and substructuring for each of these groups.

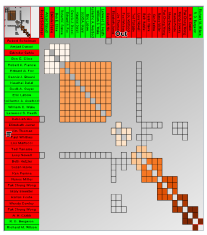
2.3.1 | Matrix Layout Approaches

The visual appearance of matrices is predetermined by its construction approach: Matrices assign a unique spatial unit (cell) for any value that relates two entities in the data set (two nodes in a network, an entity and/or its attributes, etc.). Mostly, the matrix appearance is predefined by a row-/column based layouting scheme, where each cell coordinate x, y represents the correspondence of dimension x versus the dimension y. We can find three subcategories:

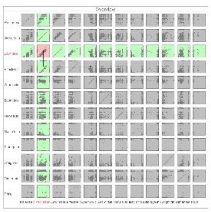
Standard Row-/Column Layout The *Standard Row-/Column Layout*, such as depicted in Figure 2.5a or Figure 2.5b, is the most frequent layouts of matrices, e.g., [Mac+03; BD10; BH11; Cha+07a; Ham03; Per13; Elm+08], which models each dimension as rows and columns. For example, in Figure 2.5a Henry and Fekete use a standard reorderable matrix to visualize large social networks.Another example is shown in Figure 2.5b, where a scatter plot matrix provides an overview about all dimensions for navigating and exploring multidimensional data [Elm+08].

Hybrid Layout A matrix can occur embedded in other visual layouts. For example, in Figure 2.5c Henry, Fekete, and McGuffin introduce the combination of node-link diagrams and matrices for the analysis of large social networks. In Figure 2.31c heat maps are placed inside a parallel coordinate visualization in order to represent multidimensional data [Lex+12]. Further examples can be found in [CMP09; Lex+10; Sil12; Via+10; WYM12; Yua+13].

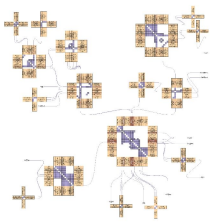
Layout Extensions/Adaptions Layout Extensions/Adaptions are a special subcategory in which the typical layout has been changed to facilitate special analysis purposes. For example, in Figure 2.5e, Dinkla, Westenberg, and Wijk present *Compressed Adjacency*



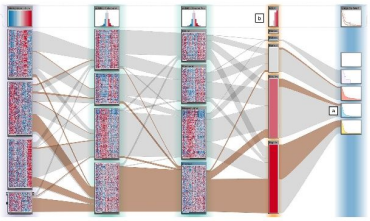
(a) Standard Row-/Column Layout [HF06]



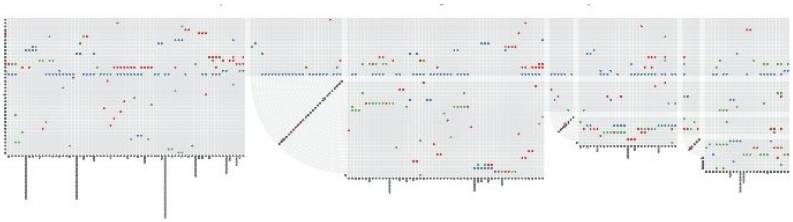
(b) Standard Row-/Column Layout [Elm+08]



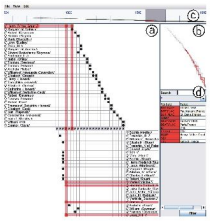
(c) Hybrid Layout [HFM07]



(d) Hybrid Layout [Lex+12]



(e) Layout Extensions/Adaptions [DWW12]



(f) Layout Extensions/Adaptions [Bez+10a]

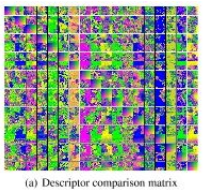
Figure 2.5 Examples of various Matrix Layout Approaches: (left) Standard row/column Layout, (middle) Hybrid Layout (right) Layout extensions/adaptions.

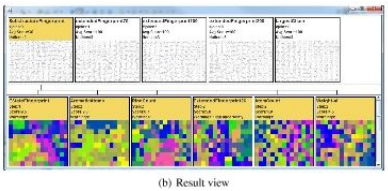
Matrices to represent and visualize the structure of gene regulatory networks. In another work, shown in Figure 2.5f, Bezerianos et al. use reoccurring matrix visualization to display large genealogy datasets. Further examples of this category can be found e.g., in [Bac13; Goo+05; HSW12; Sed+12; Sip+12].

2.3.2 | Matrix Cell Encodings

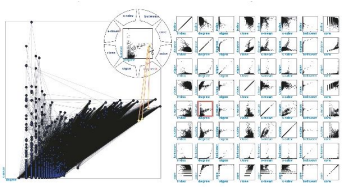
In contrast to the previous category, which focused on the layout of each matrix cell, this subcategory investigates the visual appearance of individual matrix cells. Again several subgroupings can be distinguished.

Uniform Matrix Cell Representations Matrix-representations with uniform cell representations encode the content of *all* cells in a uniform way, i.e., the data is mapped to color, glyphs or other visualizations. For example, in Figure 2.6a a color coding is used to represent similar classification results. Another example is depicted in Figure 2.6b where Scatter Plots representations are shown within each matrix cell. Further examples of this category can be found e.g., in [AH04; AAB07; Cha+07b; WYM12; Yan+99; You+13].



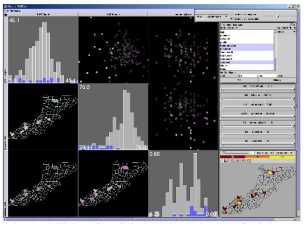


(a) Uniform Cell Representations in [Bre+11]

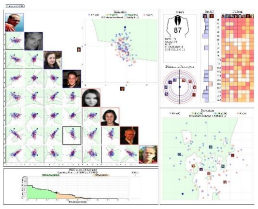


(b) Uniform Cell Representations in [Via+10]

(c) Heterogeneous cell representation in [IML13]



(d) Heterogeneous cell representation in [Mac+03]



(e) Heterogeneous cell representation in [MGW11]

Figure 2.6 Uniform and Heterogeneous Representations for Matrix Cells: Standard uniform cell representations are visually encoding every matrix cells with one selected visual encoding. In heterogeneous matrix representations every cell can show a distinct visual encoding.

Heterogeneous Matrix Cell Representations Matrices may take advantage of multiple distinct cell representations, i.e., within a matrix different visualization techniques are integrated. The most general instantiation of this idea is presented by Im, McGuffin, and Leung in their Generalized Plot Matrix *GPLOM* (depicted in Figure 2.6c). Here, three different visualization techniques are used for three different data types: Scatterplots for pairs of continuous variables, heat maps for pairs of categorical variables and bar charts for the combination of categorical and continuous variables. But already earlier in 2003, [Mac+03] presented a heterogeneous matrix cell display to show a correlation matrix and guide the user to interesting data dimension for a further analysis (cf. Figure 2.6d). Further examples of this category can be found e.g., in [HP11; Lex+12; MGW11; PBK10].

Figure 2.7 The taxonomy of the reviewed algorithms. For each algorithm the taxonomy reports first author, the year and the corresponding bibliographic reference is given.

2.3.3 | Automatic Support for Pattern Generation in Matrix-based Representations

The core algorithmic processing for matrix representations is *matrix reordering*. However, other automatic processing techniques may influence the visual appearance of matrices, too, and should be mentioned for the sake of completeness: For example, (semi-)automatic aggregation and -filtering techniques are presented e.g., in [Bre+16; Lex+14; Ale13; Bou+13; Ren+05]. Also, the matrix itself can have dynamic aspects and may change the cells' visual appearance or the even the layout during the exploration process. For example, in [DWW12] a compressed matrix iteratively expands to reflect the user interactions. Similarly, Guo, Ward, and Rundensteiner present an interactive system, in which the data space visualizations (in the form of a SPLOM) and model space visualizations (in the form of a pixel-based glyph matrix) change its appearance. Other approaches are automatic Focus+Context suggestions [Goo+05], automatic ranking [Ale13], automatic view adaption based on interaction gestures [Via+10].

The core visual appearance of matrices is defined by its matrix ordering. Therefore, we survey in the following algorithms for matrix reordering. Our coverage is not exhaustive

but biased towards impact publications in the respective sub-domains. There is also a large number of methods to speed-up or otherwise improve some algorithmic approaches, but details can be found in the original articles that are cited.

For the purpose of simplicity and understandability, we decided to group the algorithms into seven algorithmic families that arise from the *inherently shared reordering concept*, as depicted in our taxonomy in Figure 2.7. While all algorithms share the same objective of deriving an appropriate matrix reordering, every algorithm itself comprises its own design considerations and decisions.

Our classification of algorithms is tailored to the central goal of providing guidance on what algorithm to use depending on the dataset characteristics (e.g. size and structure). During our research we examined different taxonomies and orthogonal dimensions to describe algorithms in a comprehensive way; the domain they were developed in, the mathematical background, and the kind of information used to determine the distances between vertices (rows and columns). However, we found that none of those taxonomies was expressive enough while remaining simple to classify algorithms. Overall, we derived orthogonal taxonomy families/groups wherever possible, but also use the concept of overlapping and meta families to stress the importance of shared concepts or to emphasize particular features.

Multiple Ways of Reordering

We classify the algorithms for computing these permutations, depending on the stages and intermediate objects required to perform the computation. Figure 2.8 outlines the steps involved in reordering:

1. **Partition** the network into connected components and apply the reordering in each component separately. For the final matrix, the components are usually concatenated by decreasing sizes.
2. **Transform** the data into intermediate objects, such as distance matrices, Laplacian matrices, or Eigenvector spaces.
3. **Create a permutation** from those intermediate objects using some permutation operation,
4. **Assess the quality** of the obtained permutation. If it is unsatisfactory, create a new permutation, otherwise
5. **Apply permutation** to the matrix, i.e. reorder rows and columns in the visual matrix accordingly.

The following sections will explain which intermediate objects as well as permutation and quality assessment methods each algorithm group employs.

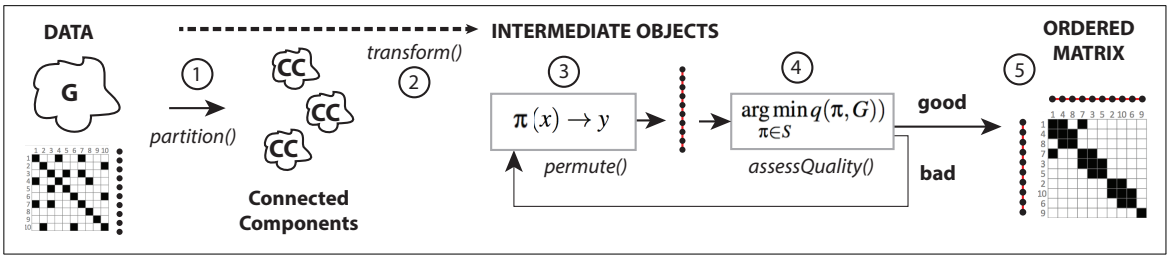


Figure 2.8 General process of reordering matrices.

However, other possibilities to structure the landscape exist, as well. An alternative classification approach is presented on our website: http://matrixreordering.dbvis.de, where we report on the design space considerations of the respective algorithms. This changes the focus to the overall question of which design combinations have never been explored. For this purpose, we derived for every algorithm its primary *feature vector* approach (e.g. row-/column vector or Eigenvector), its *objective/similarity function* (e.g. Minkowski distance, path length) and the applied *linearization method* (e.g. dynamic programming, scalar ordering). We present some interesting findings of this categorization in Figure 2.3.3.

Robinsonian (Statistic) Approaches

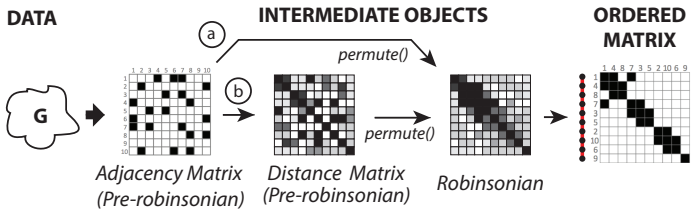


Figure 2.9 Robinsonian Matrix Reordering.

The fundamental idea of all Robinsonian approaches is to reorder a matrix so that similar rows (resp. columns) are arranged close and dissimilar rows (res. columns) are placed farther apart.

Robinsonian methods compute a similarity matrix from the A vectors (resp. B) of the (adjacency) matrix M. They then try to compute a permutation π_r (resp. π_c) to transform these similarity matrices into a Robinson matrix or R-matrix. A $n \times n$ symmetric matrix R is called a Robinson similarity matrix if its entries decrease monotonically in the rows and columns when moving away from the diagonal, *i.e.*, if:

$$\begin{cases} R_{i,j} \leq R_{i,k} & \text{for } j < k < i \\ R_{i,j} \geq R_{i,k} & \text{for } i < j < k \end{cases} \tag{2.11}$$

If instead of computing a similarity we compute a distance, then the entries should *increase* monotonically, but the principle remains identical. This property means that similar vertices are as close as possible in a consistent way.

When a similarity matrix can be permuted to become a R-matrix, it is called a *Pre-robinsonian matrix* (Pre-R) (see Figure 2.9 (middle)). The challenge is to find the permutation. When found, this permutation can be applied to the similarity matrix as shown in the Figure 2.9 (middle), but also to the original matrix M.

However, there are similarity matrices that are not Pre-R; in other words, not all the similarity matrices can be permuted to become an R-matrix. For real world cases, very few matrices are in fact pre-R. Therefore, two problems arise:

1. When a matrix is Pre-R, how to compute the permutation that transforms it into a R-matrix form?
2. When a matrix is not Pre-R, what is a good approximation of a R-matrix and how to compute it?

It turns out that there is a solution to the first question that has been ignored until recently by the statistical community [ABH98]. However, it does not address the second question at all. Therefore, the heuristics developed to approximate the general Robinsonian problem are still useful since they provide many solutions potentially applicable to the second question.

Distance/Similarity Measures

The first step to all the traditional Robinsonian algorithms consists in computing the similarity matrices from the (adjacency) matrix. Computing a distance—alike the similarity—matrix is always quadratic in time and space, and it implies choosing a measure. For distances, classical measures include the well known norms L_p:

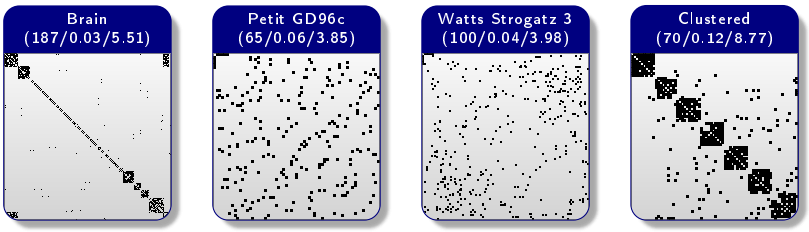
$$L_p(x,y) = (\sum_{k=1}^{k} (x_k - y_k)^p)^{1/p}, p > 0 \tag{2.12}$$

The most used are L_1, L_2, and L_∞ that simplify as (see Equation 2.7 for L_2) $L_1(x,y) = \sum_k |x_k - y_k|$ and $L_\infty(x,y) = \max_k(|x_k - y_k|)$.

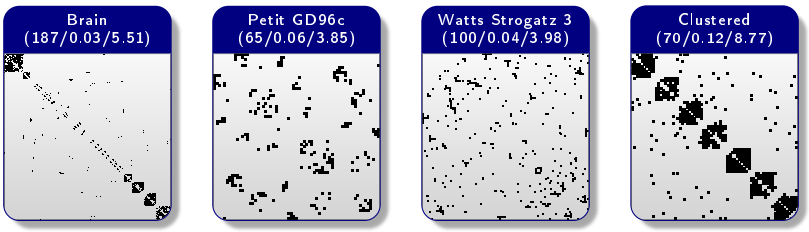
The choice of the measure may not be considered arbitrarily and has *significant impact* on the visual appearance of the matrix to be calculated. Since this aspect relates also to other matrix reordering families we will discuss distance metrics and parameterizing algorithms in Figure 2.3.3.

Algorithms and Variations

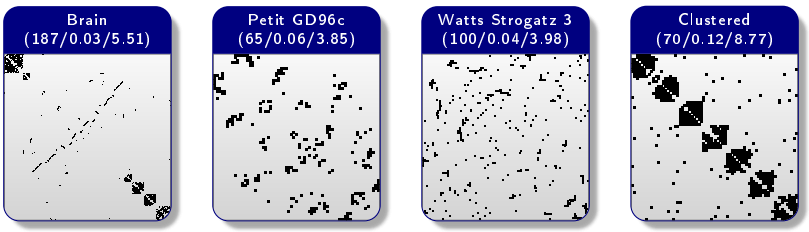
We now survey the three main approaches to compute a good permutation for the Robinsonian problem: greedy algorithms, clustering, and optimal-leaf ordering. Almost all algorithms in this group have to deal with the problem of potentially retrieving a *local optimal solution*, since a full enumeration of all permutations in the problem space is mostly infeasible. Few algorithms exist that are able to retrieve perfect anti-Robinson structures. These methods are not practical, due to their runtime, but provide an upper bound to



(a) Bipolarization [Hub74]



(b) Hierarchical Clustering [GW72]



(c) Optimal-Leaf-Ordering [BGJ01]

Figure 2.10 Examples for Robinsonian Matrix Reorderings.

this family of algorithmic methods. For example, the Branch-and-Bound algorithm or the Heuristic Simulated Annealing approach from Brusco and Stahl [BS05; BKS08] are able to retrieve global optimum solutions for up to 35 nodes and 35×35 connections with CPU times ranging from 20 to 40 min [BS05, p. 263].

(a) Greedy Algorithms: For large graphs with hundreds or thousands of nodes, on can enumerate permutations in a greedy fashion. For example, the *Bipolarization* algorithm [Hub74] alternates between rows and columns to reorganize a distance matrix, placing the highest dissimilarities in the remotest cells from the diagonal (Algorithm 1). Like most other greedy algorithms, Bipolarization (depicted in Figure 2.10(a)) is fast and yields good results whenever the underlying data structure contains a bi-polar organization (e.g., patterns *P1*, *P2*, *A1*). However, as already mentioned, greedy algorithms do not guarantee to find an optimal solution, but quickly yield a "reasonable good" solution. These solutions can then be improved further by seeking to optimize locally. Figure 2.10(c) shows examples of locally optimized matrices.

Algorithm 1 Greedy suboptimal enumeration of matrix permutations [Hub74; CP05].

```
procedure BIPOLARIZATION ALGORITHM
    D ← distMatrix(M).                                   ▷ Distance Matrix
    D_max ← max(upperTriangle(D)).
    π_col[1] ← colIndex(D_max).
    π_row[1] ← rowIndex(D_max).
    D ← applyPermutation(π_col,row, D).                  ▷ Reorganize Distance Matrix
    D_max ← maxFromIndex(row_i, col_i, i).
    if maxFromIndex(row_i, i) > maxFromIndex(col_i, i) then
        rowDirection ← true.
        π_row[2] ← rowIndex(D_max).
    else
        rowDirection ← false.
        π_col[2] ← colIndex(D_max).
    end if
    D ← applyPermutation(π_col,row, D).                  ▷ Reorganize Distance Matrix
    rIndex, cIndex ← 3.
    repeat
        if rowDirection then
            D_max ← maxFromIndex(row_i, i).
            π_row[rIndex] ← rowIndex(D_max).
            rIndex+ = 1.
        else
            D_max ← maxFromIndex(col_i, i).
            π_col[cIndex] ← colIndex(D_max).
            cIndex+ = 1.
        end if
        D ← applyPermutation(π_col,row, D).
        rowDirection ← notrowDirection.
    until rIndex = cIndex = |M|.
    M ← applyPermutation(π_col,row, M).                  ▷ Reorganize Data Matrix
end procedure
```

(b) Clustering Algorithms: Clustering algorithms, in the context of matrix reordering, are based on deriving clusters of "similar" data elements (e.g. nodes) and ordering each clusters individually. Building on this, Gruvaeus and Wainer [GW72] suggested to order clusters at different levels using an *hierarchical* clustering (dendrogram). Elements at the margin of each cluster, i.e. the first and last element in the obtained order for the respective clusters, should also be similar to the first (or last) element in the adjacent cluster. Figure 2.10(b) shows the result of the hierarchical clustering algorithm by Gruvaeus and Wainer (*RSeriationGW*).

Hierarchical clustering algorithms aim at produce grouping patterns (*P1*), however, groups are not necessarily placed along the matrix diagonal (Figure 2.10(b), *P2*). In biology, Eisen et al. [Eis+98] used agglomerative hierarchical clustering with average linkage to reveal interesting phenomena in gene expression data matrices. As also discussed by Wilkinson in "The Grammar of Graphics" [Wil05, p. 526-527] the choice of the average linkage method often yields visually good results, but represents a middle-ground between

two extremes: single and complete linkage. While complete linkage tends to produce spherical clusters, single linkage tends to produce snakelike clusters.

(c) Optimal-Leaf-Ordering Algorithms: In addition to the aforementioned clustering approaches, smoothing the clusters by ordering the vertices according to their neighborhood similarities reveals structures more clearly than walking the leaf of the hierarchical clusters in an arbitrary order. Finding an ordering consistent with the hierarchical binary tree is known as the *Optimal-Leaf-Ordering* problem. An optimal ordering is computed globally, so as to minimize the sum of distances between successive rows (columns) while traversing the clustering tree in depth-first order. For any two nodes p and q in the binary clustering tree and that share the same parent, two orders are possible: (p,q) or (q,p).

Bar-Joseph et al. [BGJ01] describe an exact solution that has a time complexity of $\mathcal{O}(n^4)$ and a memory complexity of $\mathcal{O}(n^2)$.Though this can be improved at the expense of more memory, using *memoization* techniques. Brandes [Bra07] was able to present a solution with time complexity of $\mathcal{O}(n^2 \log(n))$ and memory complexity of $\mathcal{O}(n)$, making it practical for larger matrices. Figure 2.10(c) depicts exemplified matrix reordering results with visually coherent block patterns (*P1*◪) derived from the *RSeriationOLO* algorithm.

Discussion

The quality of Robinsonian approaches directly is influenced by two choices: (i) the measure of distance (or similarity) and (ii) the enumeration approach. The goal of every R-matrix reordering approach is to optimize similarity between neighboring rows and columns. The direct outcome is that blocks patterns (*P1*◪) are made visible. Hence, Robinsonian approaches should be preferred if a dataset partitioning is expected for yet undetermined data subgroups. On the other hand, even if reordering is less strict than clustering, "*analysis via clustering makes several a-priori assumptions that may not be perfectly adequate in all circumstances. First, clustering [...] implicitly directs the analysis to a particular aspect of the system under study (e.g., groups of patients or groups of co-regulated genes). Second, clustering algorithms usually seek a disjoint cover of the set of elements, requiring that no gene or sample belongs to more than one cluster*" [TSS05, adapted, p. 3-4].

Yet, even if the data contains inherent groupings, the similarity function has to be selected with caution, since an inappropriate choice will disturb grouping patterns. Generally, when clusters exists, (hierarchical) clustering approaches show visually promising results, for all linkage functions.
In turn, Robinsonian approaches –in comparison to the other reordering approaches– allow incorporating more domain-specific knowledge into the analysis process, e.g., by applying domain-specific similarity considerations. The Bertifier [PDF14] system uses

extensively this flexibility to allow interactively specified preferences to influence the reordering algorithm.

Spectral Methods

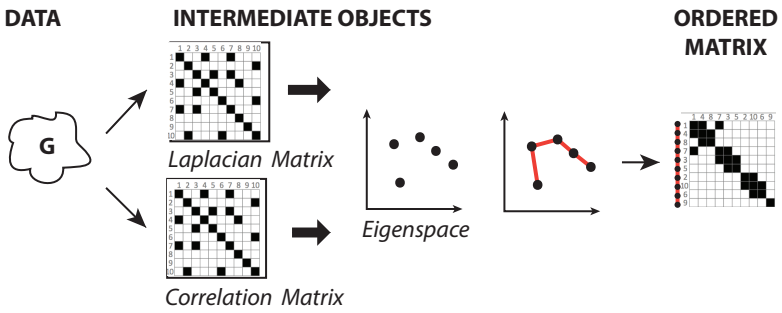


Figure 2.11 Spectral Matrix Reordering

Spectral methods relate to linear algebra and use eigenvalues and eigenvectors to calculate a reordering, i.e. each row (or column) is projected into the eigenspace where distances between eigenvectors are used to calculate a reordering (Figure 2.11). Given a symmetric matrix M of dimension $n \times n$, we say that λ is an *eigenvalue* of M if $Mx = \lambda x$ for some vector $x \neq 0$. The corresponding vector x is an *eigenvector*. An $n \times n$ symmetric matrix has n eigenvectors that can be constructed to be pairwise orthogonal, and its eigenvalues are all real. We refer to the eigenvalues in increasing numbers $\lambda_1 \geq \lambda_2 \ldots \geq \lambda_n$.

Computing the eigendecomposition of a matrix is an expensive operation. However, since the reordering algorithms will use only the few first or last eigenvectors, iterative methods can efficiently be used, in particular *Power-Iteration* [HK02] (see Algorithm 2).

Algorithm 2 Power-Iteration to compute the first k eigenvalues and eigenvectors [HK02].

function POWERITERATIONS(M) ▷ This function computes $x_1, x_2, \ldots, x_k$, the first k eigenvectors of M.
  const $\epsilon \leftarrow 0.001$
  for $i \leftarrow 1, k$ **do**
    $\hat{x}_i \leftarrow$ random
    $\hat{x}_i \leftarrow \frac{\hat{x}_i}{\|\hat{x}_i\|}$
    do
      $x_i \leftarrow \hat{x}_i$ ▷ orthogonalize
      for $j \leftarrow 1, i-1$ **do**
        $x_i \leftarrow x_i - (x_i^T x_j) x_j$
      end for
      $\hat{x}_i \leftarrow M x_i$
      $\hat{x}_i \leftarrow \frac{\hat{x}_i}{\|\hat{x}_i\|}$
    while $\hat{x}_i^T x_i < 1 - \epsilon$
    $x_i \leftarrow \hat{x}_i$
  end for
return $x_1, x_2, \ldots, x_k$
end function

The Power-iteration method is efficient when the largest eigenvalues have different magnitudes. Otherwise, the system is *ill-conditioned* and convergence will be slower or not existent at all. Efficient *preconditioning* methods exist to address this problem, such as the `LOBPCG` method [Kny01], with implementations in many popular languages.

To compute the eigenvectors with the smallest eigenvalues, a simple transformation is required. The matrix $M' = g \cdot I - M$ has the same eigenvectors as a symmetric matrix M but with the eigenvalues in reverse order. The constant g is called the Gershgorin bound [Wat91], which is a theoretical upper bound for (the absolute value of) the largest eigenvalue of a matrix:

$$g = \max_i \left(M_{i,i} + \sum_{j \neq i} |M_{i,}| \right). \tag{2.13}$$

Algorithms and Variations

Spectral methods are used in two ways: a) using the properties of the eigenvectors with the largest eigenvalue(s), or b) using the eigenvector with the smallest non-null eigenvalue due to its structural properties.

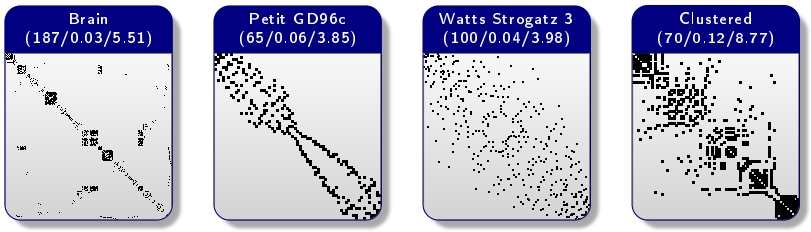
Sternin commented already in 1965 on the useful rank-order properties of the two eigenvectors corresponding two largest eigenvalues λ_1 and λ_2 [Ste65] In his divide-and-conquer approach, he splits the row/column indices into three distinct classes according to their index position and the value of the integral abscissa values of the *second* principal component at that specific index position. Each class is then permuted by the value of integral abscissa values of the *first* principal component at the specific index position.

Friendly [Fri02] developed this concept further in 2002. He was using *correlation matrices* as opposed to the raw data (Figure 2.11) to position similar variables adjacently, facilitating perception. Moreover, rather than sticking to the integral abscissa values of the first two principal components, he arranges the row/columns based in the angular order of the eigenvectors:

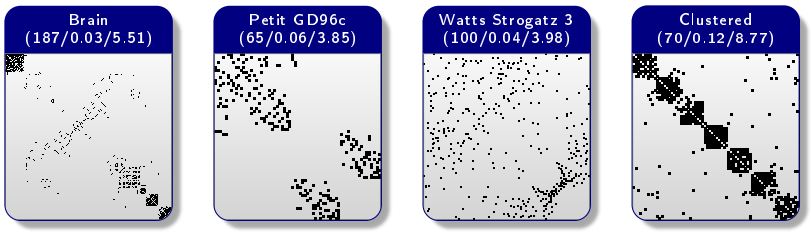
$$\alpha_i = \begin{cases} \tan^{-1}(e_{i2}/e_{i1}) & \text{for } e_{i1} > 0 \\ \tan^{-1}(e_{i2}/e_{i1}) + \pi & \text{otherwise} \end{cases} \tag{2.14}$$

The achieved circular order for the row/column vectors is unfolded into a linear arrangement by splitting at the largest gap between adjacent vectors. Figure 2.12 (a) shows example results for the *RCorrplotSortingAOE* algorithm, depicting visually pleasing global structures, but also bandwidth patterns (*P4*) without recognizable structure within the band.

A related technique was introduced by McQuitty in 1968 [McQ68], who observed the convergence of recursively formed (Pearson) correlation matrices into a matrix with the only elements being -1 and $+1$. Starting from $R^{(1)}$ (the correlation matrix of the original distance matrix) the sequence $(R^{(1)}, R^{(2)}, \ldots)$ is formed by $R^{(i)} = corr(R^{(i-1)})$ and



(a) Angular Order of Eigenvectors [Fri02]



(b) Rank-two ellipse seriation [Che02]

Figure 2.12 Examples for Spectral Matrix Reorderings.

eventually derives $R^{(\text{inf})}$ with the mentioned properties. In 1975 Breiger, Boorman and Arabie [BBA75] found that the resultant block form of the correlation matrix $R^{(\text{inf})}$ (if correctly ordered) represents a valid matrix reordering. Later, in 2002 Chen [Che02] developed these ideas further and explored a rank reduction property with an elliptical structure, even before the convergence, as Algorithm 3 showcases. Figure 2.12(b) shows exemplified results for Chen's rank-two ellipse seriation with noticeable (even though sparse) off-diagonal pattern tendencies (*P2*).

Algorithm 3 Rank-two Ellipse Reordering with recursively built Pearson correlation matrices [Che02].

procedure RANK-TWO ELLIPSE REORDERING
  $D \leftarrow distMatrix(M)$. ▷ Distance Matrix
  $R^{(0)} \leftarrow PearsonCorr(D)$.
  $i \leftarrow 1$.
  repeat
    $R^{(i)} \leftarrow PearsonCorr(R^{(i-1)})$.
    $i \leftarrow i+1$.
  until $rank(R^{(i)}) = 2$. ▷ Recursive Pearson Corr. Matrices
  Get first two Principal Components (PCs) of $R^{(i)}$.
  Project rows/columns onto the 2D plane of these PCs.
  Cut ellipse between the two converged groups.
  $\pi \leftarrow$ 1D rank approximation.
  $applyPermutation(\pi, M)$. ▷ Final Matrix Permutation
end procedure

Solution to the Robinsonian Problem Atkins et al. [ABH98] have solved the Robinsonian problem using the eigenvectors of the *Laplacian matrix*. The Laplacian matrix is defined as $L = D - M$, where D is the degree matrix and M is the adjacency matrix of the graph G:

$$D_{i,j} := \begin{cases} \sum_{k=1}^{n} M_{k,i} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{2.15}$$

A general graph G with c connected components has c eigenvectors with an eigenvalue of 0. If the graph has only one connected component, then $\lambda_n = 0$ and the associated eigenvector is a vector filled with 1 and noted $\mathbf{1}$. The smallest non-null eigenvalue λ_{n-1} is called the *Fiedler value* and the corresponding eigenvector x_{n-1} is called the *Fiedler vector* [ABH98; Fie73]. The fiedler vector can be used to order a graph by ordering the vertices according to the order of the vector values. This order is a solution to the Robinsonian problem for a Pre-R matrix.

Discussion

Spectral approaches using the first eigenvectors build on the assumption that the core matrix structure can be extracted from only a few dominant dimensions. Unfortunately, the eigenvectors are very sensitive to data corrupted with outliers, missing values, and non-normal distributions [Liu+03]. In cases where the underlying correlation-dimension is not uni-dimensional (e.g. multi-dimensional, or cyclic), such as Wilkinsons *circumplex* form [Wil05, c.f. p. 521], these approaches will fail inherently, producing salt-and-pepper visual patterns (*A1*).

Spectral approaches using the Fiedler vector seem robust to noise and tend to generate consistently good results.

Dimension Reduction Techniques

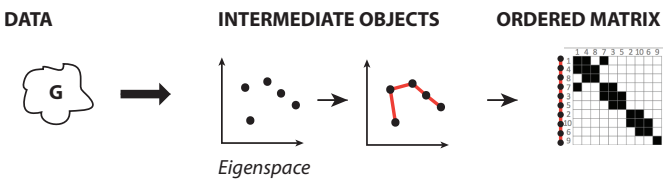
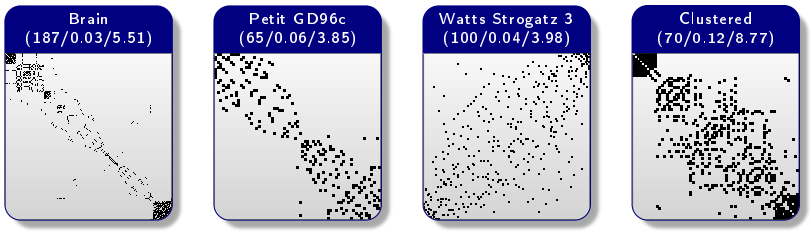
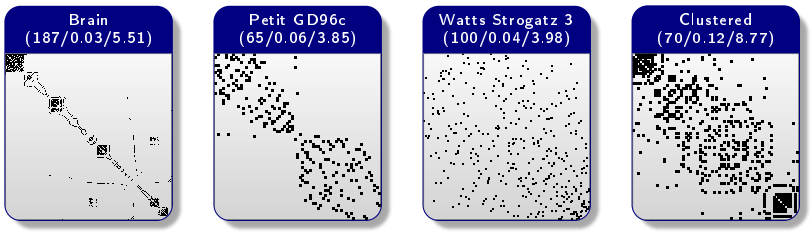


Figure 2.13 Dimension Reduction Matrix Reordering.

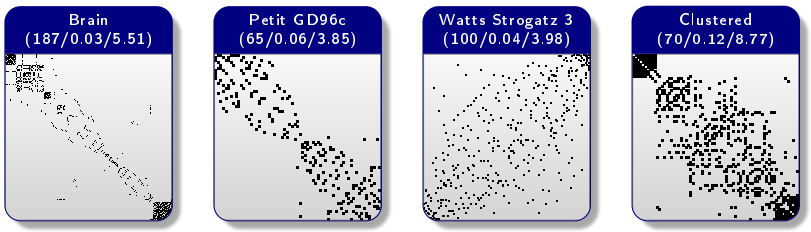
Dimension reduction techniques constitute a rather small stream to the matrix reordering landscape. While the actual mathematical ideas have been developed centuries ago, their applicability to meaningful problem instances in terms of size was hindered by the calculation performance.



(a) Principal Component Analysis [HHB08]



(b) First Principal Component Projection [Fri02]



(c) Multi-Dimensional Scaling [HHB08]

Figure 2.14 Examples for dimension reduction orderings.

The central goal of dimension reduction techniques is to retrieve a one-dimensional layout/ordering of the rows/columns that reflects the *(non-)linear relationships* between them (Figure 2.13).

Algorithms and Variations

The main methods are *Principal Components Analysis* (PCA) and variants, and *Multidimensional Scaling* (MDS).

(a) Principal Component Analysis: One of the most popular techniques for dimension reduction is Principal Component Analysis (PCA). PCA computes a projection of multidimensional data into an n-dimensional space, and that preserves the variance between elements in the data. In the context of matrix reordering, the 1^{st} principal component of a *covariance matrix* must project the data. This 1^{st} principal component represents the most variance and accordingly the most expressiveness of the data. Figure 2.14(a) shows matrix plots resulting from the *RSeriationPCA* matrix reordering.

(b) First Principal Component Projection Alternatively, the 1^{st} principal component can be directly derived from the raw matrix data; without the intermediate step of building a covariance matrix. Figure 2.14(b) shows this approach on exemplified matrix plots.

As one can see, Figure 2.14(a) (PCA) and Figure 2.14(b) (First PCP) resemble each other. However, PCA is able to show off-diagonal patterns, such as *P2*, more clearly and tends to produce patterns along the anti-diagonal of a matrix.

(c) High Dimensional Embedding: Computing the eigenvectors was long time only possible for small matrices. However, iterative approaches (see Algorithm 2) and the ability to parallelize the calculations (even GPU implementations are available [And09; KY05]) allow computing the PCA in real-time for large graphs. A modified PCA method, called "High Dimensional Embedding", is described by Harel and Koren [HK02]. Their method uses the first two/three components of the PCA for laying out node-link diagrams with up to one million nodes in a few seconds. This method was adapted by Elmqvist et al. for reordering matrices for graphs of a million edges [Elm+08]. Elmqvist et al.'s results show that High Dimensional Embedding results in a visually pleasing overview, while the local ordering is poor.

(d) Single Value Decomposition: Liu et al. [Liu+03] follow the idea that data is inherently corrupted with outliers, missing values, and non-normal distributions that cover up the matrix patterns. Thus, a row vector approximation with bilinear forms $x_{ij} = r_i m_j + e_{ij}$ is used, where r_i is a parameter corresponding to the ith row, m_j corresponds to the jth column and e_{ij} is a residual/error value. Rows are ordered iteratively by their regression coefficients r_i, respectively m_j for the columns, with the assumption that similar regression coefficients group visually similar rows/columns. The equation can be solved using *Singular Value Decomposition* (SVD), which decomposes a rectangular matrix D_{mn} into the product of 3 matrices $U_{mm}\Sigma_{mn}V_{nn}^T$ where $U^TU = I$ and $V^TV = I$. Although the method is robust, it is expensive since its complexity is higher than quadratic in time.

(e) Multi-Dimensional Scaling: Similar to PCA, another possibility to discover structure in matrices. is multi-dimensional scaling (MDS)[BL12]—also denoted as Similarity Structure Analysis. In 1974, Spence and Graef recognized this interrelation and applied MDS to the matrix reordering problem [SG74]. MDS assigns rows/columns to a specific index in a conceptual one-dimensional space, such that the distances between the respective vectors in the space match the given dissimilarities as closely as possible. The cost function to be minimized is an overall distortion of the positions. With this approach MDS can derive *non-linear relationships* among the matrix rows/columns.

MDS techniques can be distinguished into two types: (i) non-metric MDS involves data which is not necessarily all numeric was applied by Rodgers and Thompson [RT92] for matrix reordering, and (ii) classical MDS which involves numeric data (preferably variables in the same scale) was applied by Spence and Graef in [SG74]. Classical MDS algorithm is based on the fact the permutation indices X— or one dimensional coordinate matrices—can be derived by eigenvalue decomposition from the scalar product matrix $M = XX'$. To achieve this, each value in the distance matrix must be squared and "double centered", such that the columns and rows both have a zero mean. Subsequently the SVD of this (normalized) matrix is calculated and the index positions are retrieved from the factors returned by the SVD. The steps in Algorithm 4 summarize the algorithm of classical MDS. Figure 2.14(c) shows matrix plots for the same *RSeriationMDS* algorithm, resulting in almost identical plots than when using PCA.

Algorithm 4 Double Centering and Singular Value Decomposition in the MDS Matrix Reordering.

procedure MDS MATRIX REORDERING
  $D^2(i,j) \leftarrow -\frac{1}{2} d(x_i, x_j)^2$. ▷ Squared Distance Matrix
  $rowMean = mean(M)$.
  $colMean = mean(transpose(D^2))$.
  $totalMean = mean(rowMeans)$.
  for $i = 0, |D^2|$ **do**
    for $j = 0, |D^2|$ **do**
      $D^2(i,j) += totalMean - rowMean_i - colMean_j$;
    end for
  end for ▷ Double Centering
  $U\Sigma V^\top = \text{SVD}(D^2)$ ▷ Singular Value Decomposition
  $eigenValues \leftarrow \sqrt{\Sigma}$
  $U' \leftarrow \sqrt{\Sigma}$
  for all $row \in U$ **do**
    $row \times eigenValues$.
  end for ▷ Eigenvector Normalization
  $\pi \leftarrow U_1$.
  $M \leftarrow applyPermutation(\pi, M)$. ▷ Final Matrix Permutation
end procedure

Alike PCA methods, classical MDS can be heuristically adapted to allow for larger problem instances. Brandes and Pich [BP07; Pic09] propose *PivotMDS*, a sampling-based approximation technique for classical MDS, which is able to determine a layout of node-link diagrams in linear calculation time and with linear memory consumption.

Discussion

The central idea of dimension reduction techniques is to take advantage of the inherent and potentially hidden (non-)linear structures in the data. This has direct consequences

on the matrix plot to be expected: Normally these approaches favor high-level/coarse-grained structures (*P1*) over fine matrix patterns (e.g., lines (*P3*)). While PCA is only able to retrieve linear structures, MDS also allows determining non-linear data relationships. On the other hand, there are only rare cases where a non-linear data structure should be examined in a matrix form. Other visualizations, i.e., the raw two-dimensional MDS projection, is better suited for these purposes. In general, Wilkinson notes that SVD and MDS methods are performing best in terms of the Spearman correlation between the known row indices (after constructing the matrix) and the permuted indices [Wil05, p.532].

Heuristic Approaches

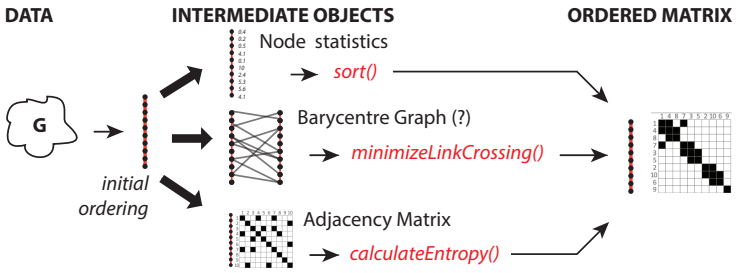


Figure 2.15 Heuristic Approaches for Matrix Reordering.

Heuristics are methods that transform the problem of finding an ordering into another problem space that abstracts the problem appropriately and allows for computationally efficient problem solving. Heuristic approaches can be separated into *problem simplification* and *problem space transformation* approaches. Problem simplification methods try to use row/column approximations to iterate through the permutation possibilities (or a subset thereof), while problem space transformation methods are transforming rows and/or columns into other meaningful representations (e.g. the nodes of a bi-graph).

Algorithms and Variations

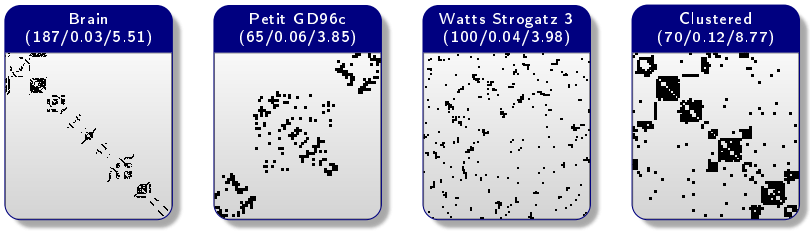
(a) Numerical Abstractions: One classical instance for simplification methods is to neglect the row dimensionality and use numerical abstractions for each row and/or column instead. Deutsch and Martin [DM71] proposed to use the *mean row moments* to find the principal axis and thus a single dominant relationship of the data. Mean row/column moments are defined as follows:

$$x_i = \frac{\sum_{j=1}^{N} j m_{ij}}{\sum_{j=1}^{N} m_{ij}} \tag{2.16}$$

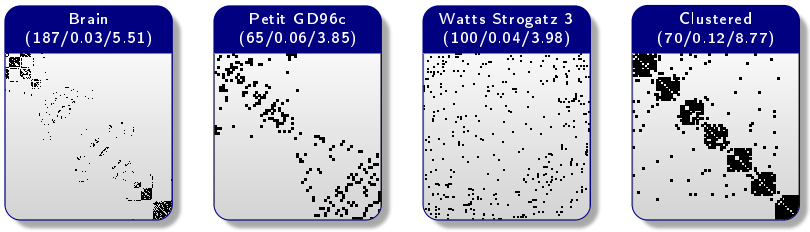
This heuristic approximation is iteratively applied *separately* on the rows and columns until the simple vector mean quality measure stabilizes, as Algorithm 5 depicts.

Algorithm 5 Separately applied row/column ordering with the Mean Row Moments quality criterion [DM71].

```
1: procedure MEAN ROW MOMENTS HEURISTIC REORDERING
2:     repeat
3:         for all row_i ∈ M do
4:             x_i ← meanRowMoment(row_i).
5:         end for
6:         π ← sort(x).
7:         M ← permute(π, M).                          ▷ Row Reordering
8:         for all col_i ∈ M do
9:             y_i ← meanColMoment(col_i).
10:        end for
11:        π ← sort(y).
12:        M ← applyPermutation(π, M).                 ▷ Column Reordering
13:    until Row and column reordering is stable.
14: end procedure
```



(a) Bond Energy Algorithm [McC+69]



(b) Anti-Robinson Simulated Annealing [BKS08]

Figure 2.16 Example for Heuristic Matrix Reordering.

McCormick et al. contrast in [McC+69; MSW72] three different heuristics for establishing a matrix reordering: (i) Moment Ordering; by means of the mean row moments, (ii) Moment Compression Ordering; by means of the sum of second moments and (iii) the *Bond Energy Algorithm*, short BEA. BEA uses a "measure of effectiveness" as a quality criterion, and tries to maximize the so called *bond energy* over all row- and column permutations. Figure 2.16(a) shows matrix plots for the *RSeriationBEA* algorithm, which shows tendencies to produce Block-Diagonal matrix forms (*P1*) in combination with off-diagonal groupings (*P2*) and star patterns (*P3*). As seen from the results in Figure 2.16(a), BEA tries to maximize contiguous chunks, forming more or less concise *groupings (P1, P2)* in the matrix plot.

Three further heuristics are suggested by Hubert and Golledge: “Different Counting Rule”, “Gradient Within Rows”, “Szczotka’s Criterion” [HG81, p. 436-439]. Mäkinen and Siirtola [MS00] propose to reorder the rows/columns iteratively and separately (such as in Algorithm 5) by their weighted row/column sum. Further heuristics with varying goals are described in [Maf14, p. 199].

(b) Barycenter Heuristic: Mäkinen and Siirtola propose to use the *barycenter heuristic* [MS05] to layout a bipartite graph, in which the matrix rows correspond to the first graph partitioning and the columns to the other partitioning. An adaption of the *Sugiyama* algorithm is applied to minimize the edge crossings (Figure 2.17).

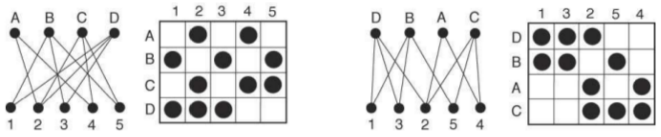


Figure 2.17 Reordering using the Barycenter Heuristic [MS05].

Figure 2.17 operates on the two “layers” (the two partitions). It is repeated until the number of crossings does not decrease any more. Two small changes improve the algorithm results substantially: replacing the barycenter by the *median* [EW94], and applying a post-processing, repeatedly swapping consecutive vertices on the two layers as long as it lowers the number of crossings [Gan+93]. This algorithm has the tendency to arrange matrix plots, such that *clusters* stick out in the top left and bottom right corners.

(c) Entropy Optimization and Genetic Algorithms: Niermann [Nie05] uses a genetic algorithm to assesses the “fitness” of a matrix permutation by means of an entropy-based objective function over all matrix plot pixels. The intuitive idea is that a well-ordered matrix plot—containing locally coherent regions of similar values (e.g. black or white)—requires a minimum number of bits for encoding. In other words, the better a matrix plot can be compressed, the better is its reordering (under the assumption the data contains clusters, groupings, or other partitionings). For the Niermann’s genetic algorithm he models permutations, the individuals of the algorithm, as arrays of ordering indices; child individuals are created from consistently rearranging permutation subsequences in the parents (crossover); mutations are implemented by reversing arbitrary subsequences in the permutation. After each round, the fitness of every offspring is evaluated. Less fit individuals are discarded, more fit offsprings (permutations) survive and can reproduce.

Brusco et al. [BKS08] and Wilkinson [Wil05] propose to use simulated annealing (depicted in Figure 2.16(b)), a technique similar to genetic algorithms and used for finding (local) optima in large search spaces. In each annealing step two rows or two columns are

exchanged and the performance is measured in terms of anti-robinsion events, respectively residual variance.

Discussion

Heuristic approaches transform the matrix reordering problem, such that specific assumptions are met. While problem simplification algorithms are usually fast, they suffer inherently from this restriction. If a dataset is not of the expected form, the results will be inappropriate for an analysis (*A1*). One other problem seems to be specific for problem space transformation: The algorithms are reported to converge slowly and are sensitive to parameter settings. Also it is questionable whether general settings can be derived or inherently depend the structure and size of the data sets. Particularity in these cases, it might be beneficial to (pre-)assess the matrix in terms of density, clusteredness, etc.

Graph-Theoretic Approaches

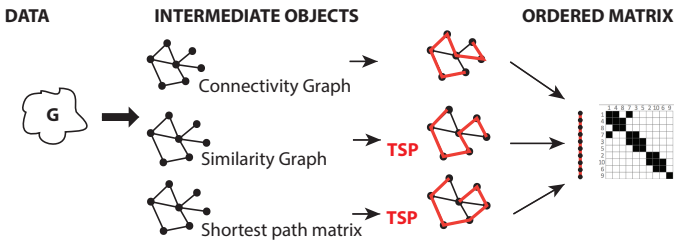
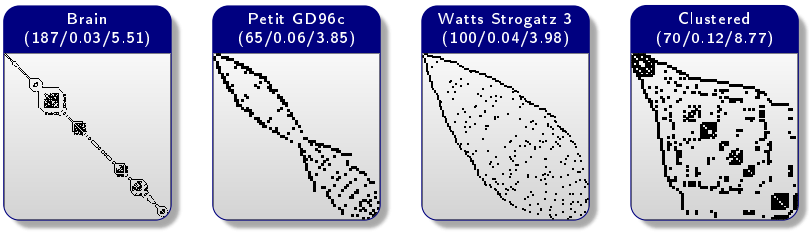


Figure 2.18 Graph-Theoretic Approaches for Matrix Reordering.

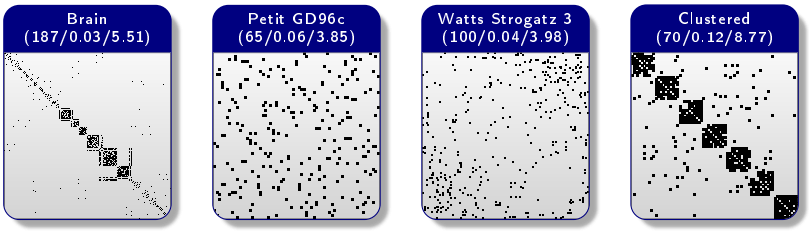
Graph-based approaches share a commonality with heuristic methods: They transform the permutation problem into a related problem space, in this case graph enumeration. The central idea of graph-theoretic approaches is to exploit the graph structure for computing a linear order that optimizes a graph-theoretic layout cost function. Díaz et al. [DPS02] compiled a list of nine layout cost functions from which three objectives have been applied in the context of matrix reordering. We detailed these layout cost functions, along with their visual interpretation in Section 2.2.1.

Algorithms and Variations

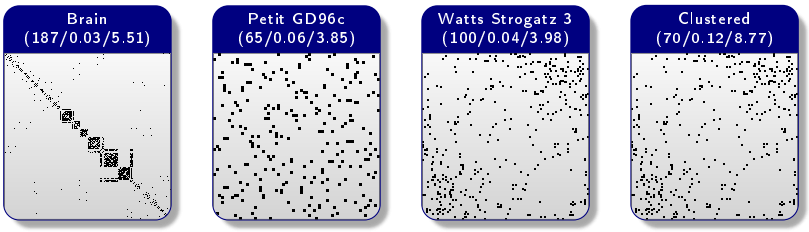
(a) Bandwidth Minimization: In an early approach (1968) Rosen presented an *iterative* method to reduce the bandwidth in sparse matrices [Ros68]. The same central objective is shared by the well-established *(Reverse) Cuthill-McKee* matrix reordering algorithms [CM69; Geo71]. Cuthill and McKee exploit a direct correspondence between the structure of the coefficient matrix and the structure of the adjacency matrix to be ordered. Algorithm 6 shows the pseudo code for the Reverse Cuthill-McKee algorithm. Starting from a graph vertex with low degree, it enumerates, in a *breadth-first search* manner, all



(a) Cuthill-McKee [CM69]



(b) Reverse Cuthill-McKee [Geo71]



(c) Sloan [Slo86]

Figure 2.19 Example for graph-based approaches (a-c).

neighboring vertices sorted by their neighborhood degree (number of common neighbors with the initial vertex). Figure 2.19(a) shows exemplified matrix plots, which depict that this inherently fast approach tends to produce strong bandwidth anti-patterns (*A2*◪). However, it can also lead to good results (*P1*◪) within the bandwidth if the graph structure allows for it and—more crucial—the initial input permutation is appropriate.

An improved version of the Cuthill-McKee algorithm, known as Reverse Cuthill-McKee algorithm, is proposed by George [Geo71]. It reverses the degree ordering of the neighboring vertices in the breadth-first search. A comparative analysis of the two variants shows that this –marginal– change leads to better reordering results [LS76, p. 207]. Figure 2.19(a)(b) show exemplified matrix plots for both algorithm variants However, in our implementations the algorithms produce visually dissimilar results. In [CG80] a linear time implementation of the Reverse Cuthill-McKee algorithm is given.

The memory consumption for the Cuthill-McKee algorithm was improved by King [Kin70]. King uses a local priority queue to select the next vertex to visit, based on the amount of vertices that will be added to the neighborhood list in the subsequent iteration. Later in 1976, Gibbs, Poole and Stockmeyer focused on runtime improvements in their

Algorithm 6 Bandwidth Minimization with Breadth-First Search in the Cuthill-McKee Matrix Reordering Algorithm [CM69].

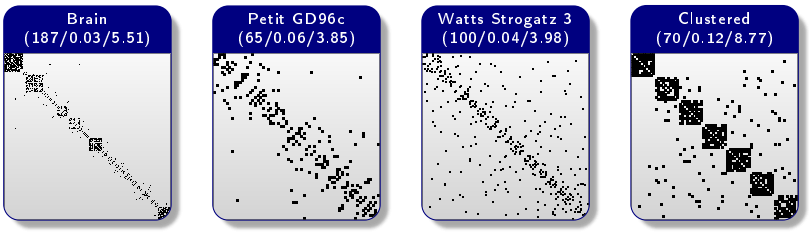
```
1: procedure CUTHILL-MCKEE MATRIX REORDERING
2:     G(V,E) ← adjacencyMatrix(M).
3:     v_start ← minDegree(V).
4:     π ← ∅ ∪ v_start.
5:     i ← 1.                                          ▷ Initialization
6:     repeat
7:         neighbors ← adjacent(v_i).
8:         sortByDegree(neighbors).
9:         for all v_n ∈ neighbors do
10:            π ← π ∪ v_n.
11:        end for
12:        i ← i + 1.
13:    until i = |V|.                                  ▷ Breadth-first enumeration
14:    M ← applyPermutation(π, M).                     ▷ Final Matrix Permutation
15: end procedure
```

popular GPS algorithm [GPS76]. GPS decreases the search space by starting with a vertex that has a maximal distance to another vertex (pseudo-diameter path) and a level minimization step to reduce the number of vertex enumeration cycles. The GPS algorithm is reported to work up to eight times faster than the Reverse Cuthill-McKee algorithm.

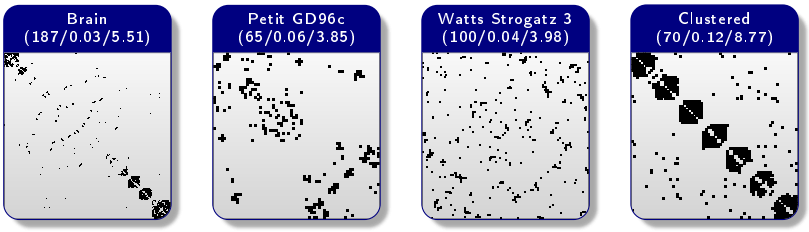
(b) Anti-bandwidth Maximization: A related and visually interesting adaption of bandwidth minimization was introduced by Leung in 1984: the matrix *anti-bandwidth maximization* problem [LVW84]. It says that after a matrix's row/column permutation all nonzero entries should be located as far as possible to the main diagonal. Figure 2.21 shows the exemplified result of a matrix reordering with respect to anti-bandwidth optimization.

Anti-bandwidth maximization is able to show off-diagonal line patterns (a sub-form of the bands pattern, *P4*ℕ, describing paths) spreading over the matrix plot. Similar line patterns can be found in the analysis of high performance computing clusters [Rüd+15a].

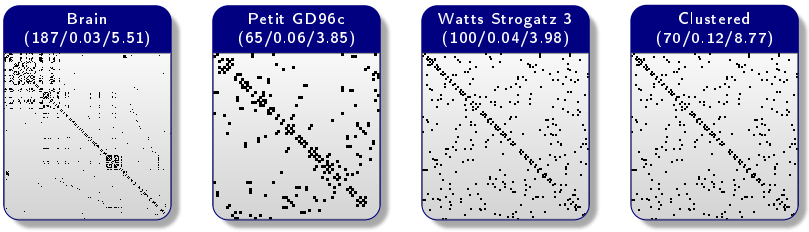
Alike bandwidth minimization, also anti-bandwidth maximization, is in the class of $\mathcal{NP}$-complete problems [LVW84]. Accordingly, heuristic approaches were developed to solve the problem. Lozano et al. [Loz+12] propose a heuristic algorithm based on variable neighborhood search. Also Lozano et al. [Loz+13] proposed a hybrid approach combining the artificial bee colony methodology with tabu search to obtain appropriate results in short computational times. A genetic algorithm for bandwidth reduction, anti-bandwidth maximization and linear ordering is proposed in [PM14], where an exchangeable cost function guides the to be expected result. Further discussion on anti-bandwidth maximization is given by Raspaud et al. in [Ras+09].



(d) Multi-Scale [KH02a]



(e) Traveling Sales Man [LK75]



(f) Multiple Fragment [Ben92]

Figure 2.20 Example for graph-based approaches (d-f).

(c) Profile Minimization: Sloan's algorithm [Slo86; Slo89] has the goal to reduce the profile and the wavefront of a graph by reordering the indices assigned to each vertex. Similarly, to GPS algorithm pseudo-peripherial nodes are chosen as a start and end vertices. All other vertices are prioritized by a weighted sum of the distance of the vector to the end vertex (global criterion). Additionally, a local criterion is incorporated with the current vertex degree. It reflects the status of the renumbering in the neighborhood of a vertex. Therefore, the Sloan algorithm not only takes into account global criterions, but also incorporates local criteria for the reordering process. Figure 2.19(c) shows exemplified matrix plots for the *Sloan* reordering algorithm.

(d) Minimum Linear Arrangement: Koren and Harel propose a multi-scale approach to deal with the MinLA problem [KH02b]. In their multi-level algorithm (depicted in Figure 2.19(d)) the entire graph is progressively divided into lower dimensional problems (reordering of a segment graph). This process is referred to as the coarsening. The coarsening of the graph is based on restricting the consecutive vertex pairs of the current arrangement. In the coarsest level exact solutions for the problem can be calculated. These sub-solutions are projected back to the higher level problem in the subsequent

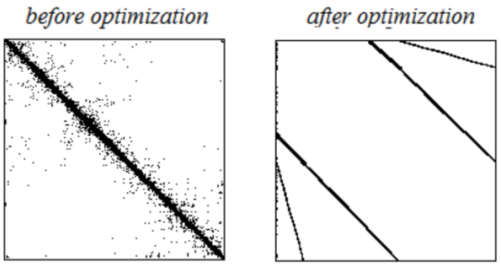


Figure 2.21 An example for the antibandwidth optimization [Maf14, image courtesy].

refinement process until the initial problem is reconstructed. This refinement step iterates over all local permutations and selects the one that minimizes the MinLA (in a dynamic programming fashion). Multi-level approaches in general have one significant advantage: They allow fast exploration of properties related to the global structure, while the local structures can be refined iteratively if necessary. Further multi-scale graph coarsening approaches are described in [OLS15].

(e) Traveling Salesman Problem: In a technical note from 1974 Lenstra pointed out that the Bond Energy Algorithm (c.f. Section 2.3.3) could be modeled as two subsequent Traveling Salesman Problems (TSP) for the rows and the columns: "*It performs well for this type of problem*" [Len74, p. 414]. Shortly after, Lenstra and Kan [LK75, p. 724] showed that the bond energy heuristic is a simple suboptimal TSP variant and compared it to the optimal TSP solution.

TSP matrix reordering approaches model each row, respectively column, as a city and translate the row/column-wise similarity in a virtual distances. Computing an optimal TSP path (a minimum distance TSP tour with a virtual city that is used for breaking the cycle) corresponds then in finding an optimal matrix permutation, such that the pairwise similarity is maximized (wrt. the chosen similarity function).

Figure 2.19(e) shows matrix plots for the *RSeriationTSP* algorithm. Internally the *Concorde* TSP solver [ACR03] and the 2-OPT edge exchange improvement procedure [Cro58] is used to minimize the Hamiltonian path length. A different approach is pursued by the *Multiple Fragment* algorithm, such as described by Bentley [Ben92] or Steiglitz and Weiner [SW68]. It starts by considering every node in the graph as one independent fragment. Repeatedly any edge that will not make it impossible to complete a tour is added to the closes fragment. Fragments can be merged by a connecting edge.

More recently, Henry-Riche and Fekete [HF06] incorporated in their MatrixExplorer system the consideration that vertices with *similar connection patterns* should be positioned next to each other. This has the advantage that not only the coarse matrix plot structure (groups/cluster patterns) is focused, but also that the local density of the occurring clusters is optimized. In their approach the authors use—instead of the adjacency

matrix—the shortest path matrix for each connected component of the graph and reorder each component with a TSP solver; alternatively a hierarchical clustering can be applied (see also Section 2.3.3).

Discussion

The idea to explore the graph structure for computing a linear ordering is self-evident and obvious. But, in analogy to our question *What is a good matrix reordering?* the graph community posing the question *What is a good 2D graph layout?*. These questions are yet unanswered in both domains. However, they share the common ground that a good result allows perceiving visual patterns.

Related to this challenge, several of the mentioned approaches, such as the Multi-Scale or TSP, have the interesting characteristic that a consistent and *intermediate* reordering result can be shown to the analyst on-the-fly, while the actual matrix reordering takes place. This idea follows the "*results-first-refine-on-demand*" mantra. On the other hand, when it comes to the optimization of graph-theoretic layout functions, such as bandwidth or profile, these algorithms are solely governed by the assumption that the data can be brought into this form. This assumption implicitly neglects all other potential patterns in a matrix plot. Though, a matrix plot that represents patterns along the diagonal well (e.g. clusters) will be perceived as interpretable and effective; a goal also expressed by Bertin [Ber73].

The efficiency aspect has to be regarded, as well. Sloan notes, that bandwidth and profile reduction "schemes may be inefficient for sparse matrices which contain a significant number of zeros inside the bandwidth envelope" [Slo86, p. 240]. Recently, Wong also noted that "algorithms such as Fiedler and Sloan consistently require more time to compute than the others when the graphs grow to tens of thousands of nodes" [Won+13, p. 95]. While this is certainly true, a heuristic implementation for most approaches can be found in the vast literature of the graph-theoric domain.

Biclustering Approaches

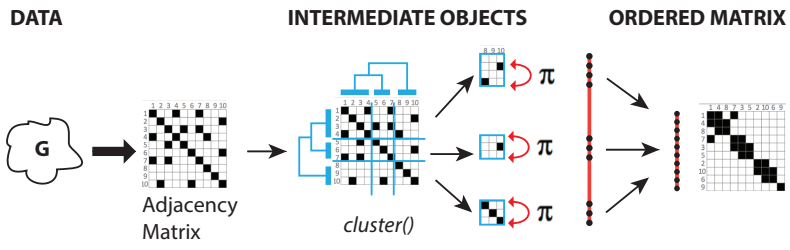


Figure 2.22 Biclustering Approaches for Matrix Reordering.

Recently, the concept of *Biclustering*, also called co- or two-mode clustering or generalized blockmodeling, gained importance for matrix reordering. Biclustering is related to clustering, but comprises a central difference: While clustering

can be applied separately to either the rows or columns of the matrix, biclustering performs clustering in these two dimensions *simultaneously* (Figure 2.22). In other words, clustering derives *global data models* and biclustering algorithms allow identifying subsets of rows and columns that reveal a similar activity pattern, i.e., *local data models*.

Biclustering approaches can be subdivided by the clustering structure they are able to reveal: (a) Single bicluster, (b) exclusive row and column biclusters, (c) checkerboard structure, (d) exclusive rows biclusters, (e) exclusive columns biclusters, (f) nonoverlapping biclusters with tree structure, (g) nonoverlapping nonexclusive biclusters, (h) overlapping biclusters with hierarchical structure, and (i) arbitrarily positioned overlapping biclusters [MO04, p. 34]. For a matrix reordering task the most general subgroup with arbitrarily positioned overlapping biclusters is of the highest interest, since it enables the user to see overlapping block patterns in the matrix plot. A repetitive execution of algorithms that reveal a single bicluster allows finding arbitrarily positioned submatrices, too.

Algorithms and Variations

(a) Single Bicluster Approaches: Cheng and Church [CC00a] present a greedy iterative search algorithm that models the biclustering problem as an optimization problem. The algorithm finds one $\delta-$bicluster (or submatrix) at a time with a potentially local optimal mean squared residue score not higher than a user-specified parameter δ.

The pseudo code for the Cheng and Church algorithm is given in Algorithm 7, depicting two greedy row/column removal/addition steps that are executed subsequently to find one $\delta-$bicluster: Starting from the full matrix, 1) rows and columns with a score higher than the mean squared residue score are deleted. 2) removed rows or columns are added if they do not increase the actual mean squared residue score of the bicluster. This approach converges with low mean residue and locally maximal size for one $\delta-$cluster. In order to find distinct biclusters, an already identified bicluster can be artificially masked with random noise so that a subsequent invocation of the algorithm finds a different $\delta-$bicluster.

(b) Arbitrarily positioned overlapping biclusters Several algorithms to retrieve arbitrarily positioned overlapping biclusters exist. For example, the *Plaid* model biclustering algorithm of Lazzeroni and Owen [L+02] assumes that a matrix can be described as a linear function of possibly overlapping constant layers that do not necessarily have to cover the whole matrix. Iteratively new layers are added to the model, so that the new layer minimizes the sum of squared errors. The Plaid model biclustering algorithm was improved by Turner et al. [TBK05], using a binary least squares algorithm to update the cluster membership parameters. This takes advantage of the binary constraints on these parameters and allows to simplify the other parameter updates.

Algorithm 7 Cheng-and-Church Biclustering algorithm [CC00a].

function CHENG AND CHURCH BICLUSTERING
  $A \leftarrow rows(M)$. ▷ Definitions.
  $B \leftarrow columns(M)$.
  $e_{Aj} \leftarrow \frac{\sum_{i \in A} m_{ij}}{|A|}$.
  $e_{iB} \leftarrow \frac{\sum_{j \in B} m_{ij}}{|B|}$.
  $e_{AB} \leftarrow \frac{\sum_{i \in A, j \in B} m_{ij}}{|A||B|}$.
  $RS_{AB}(i,j) \leftarrow m_{ij} - e_{Aj} - e_{iB} + e_{AB}$.
  $H(I,J) \leftarrow \sum_{i \in A, j \in B} \frac{RS_{ij}^2}{|A||B|}$.
  Initialize *bicluster*(I,J) with $I = A, J = B$. ▷ Algorithm Start.
  while $H(I,J) > \delta$ **do** ▷ Deletion Phase.
    $d(i) \leftarrow \frac{1}{|J|} \sum_{j \in J} RS_{IJ}(i,j)$ for all $i \in I$.
    $e(j) \leftarrow \frac{1}{|I|} \sum_{i \in I} RS_{IJ}(i,j)$ for all $j \in J$.
    if $max_{i \in I} d(i) > max_{j \in J} e(j)$ **then**
      $I \leftarrow I \setminus \{\text{argmax}_i\, d(i)\}$.
    else
      $J \leftarrow J \setminus \{\text{argmax}_j\, e(j)\}$.
    end if
  end while
  $I' \leftarrow I, J' \leftarrow J$.
  while $H(I',J') < \delta$ **do** ▷ Addition Phase.
    $I \leftarrow I', J \leftarrow J'$
    $d(i) \leftarrow \frac{1}{|J|} \sum_{j \in J} RS_{IJ}(i,j)$ for all $i \in A \setminus I$.
    $e(j) \leftarrow \frac{1}{|I|} \sum_{i \in I} RS_{IJ}(i,j)$ for all $j \in B \setminus J$.
    if $max_{i \in I} d(i) > max_{j \in J} e(j)$ **then**
      $I' \leftarrow I \cup \{\text{argmax}_i\, d(i)\}$.
    else
      $J' \leftarrow J \cup \{\text{argmax}_j\, e(j)\}$.
    end if
  end while ▷ Initialization
  return *bicluster* I, J.
end function

(c) Biclusters with similar visual patterns: Another approach, called *xMotifs*, is presented by Murali and Kasif [MK03] for the biological gene expression analysis domain. The authors state that "*a conserved gene expression motif or xMotif is a subset of genes whose expression is simultaneously conserved for a subset of samples.* [...] *If we map each gene to a dimension, each sample to a point, and each expression value to a coordinate value, an xMotif is identical to a multi-dimensional hyperrectangle that is bounded in the dimensions corresponding to the conserved genes in the motif and unbounded in the other dimensions.*" [MK03, p. 2]. In other words, the xMotif algorithm searches for biclusters that share a common visual pattern or motif. To achieve this goal, the data is discretized into statistically significant intervals—sometimes even binarized. In a probabilistic approach, randomly chosen "seed" columns are sz iteratively compared against growing sets "discriminant" columns. For these discriminant columns, all rows are incorporated into

the bicluster if they share a common state with the seed column at the specific position. Motifs with a low overlap coefficient are discarded. The algorithm is adapted for ordinal and nominal scales in the algorithms *Quest* and *Questmet* [Kai11]. A similar approach, called *BiMax* swas presented by Prelić et al. [Pre+06].

(d) A priori submatrices: An interesting biclustering variant is presented by Jin et al. [Jin+08]. Based on the assumption that a set of submatrices of interest is known a priori, the authors try to find a row/column permutation that allows showing these relationships best. For this purpose, the authors generalize the minimum linear arrangement problem into the *hypergraph vertex ordering* problem and propose to solve the submatrix pattern visualization problem in this problem domain. In their suggested algorithm existing graph ordering algorithms are incorporated to solve the optimization problem.

Several other biclustering algorithms exist and are discussed in [MO04; Pre+06; TSS05]

Discussion

Biclustering focuses on finding subsets of rows and columns that allow perceiving coherent activity patterns which cannot be seen with a global scope. Biclustering approaches are therefore operating on *local models*. By definition the retrieved biclusters, or submatrices, should form nearly uniformly colored coherent visual block patterns (*P1*, *P2*) that stand out from the neutral background color. This ideal corresponds to the existence of k mutually exclusive and exhaustive clusters, and a corresponding k-way data partitioning [L+02, p. 62].

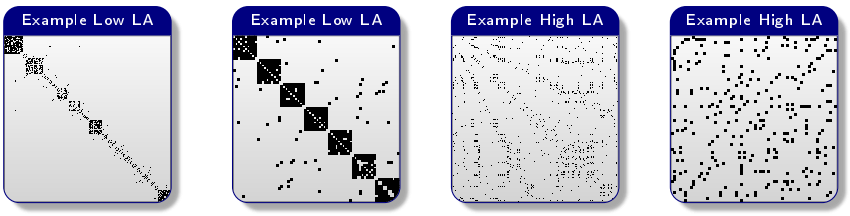
Unlike standard cluster matrix reordering approaches (see also Figure 2.3.3), biclustering approaches are not necessarily depending on a similarity model. In contrast, these approaches even doubt the rationale of an equal weighting of rows and/or columns. Cheng and Church state that any such similarity formula leads to the discovery of some similarity groups at the expense of obscuring some other similarity groups [CC00a, p. 1].

Another central difference to the other reordering approaches is the notion of data partitioning. While standard approaches mostly facilitate a data separation into exclusive groups, biclustering approaches generalize this assumption: data items may be contained in several overlapping clusters. This understanding is based on the circumstances of the gene expression data analysis domain, where co-regulatory genes patterns are represented by (multiple) subsets of conditions.

Generally, it has to be noted that the approaches in this field are not restricted to a-priori constraints on the organization of biclusters, which allows for more freedom, but consequently leads to a higher vulnerability to overfitting. This is especially obvious in the *high parameter sensitivity*: In many cases only slight modifications of the input parameters lead to empty biclustering result sets and even erroneous matrix permutations. However, this problem is mitigated by the fact that most algorithms are greedy implementations, which allows a fast but potentially locally optimal result.

Performance Comparison

The previous section grouped algorithms into 6 categories, described the underlying methods, and showed examples of resulting matrices. In this section we look at two measures to quantify "performance" of an algorithm: (i) algorithm runtime and (ii) Linear Arrangement score (LA), a measure for the compactness of blocks in matrices. A matrix, ordered by an algorithm, results in a *low* LA score if it reveals coherent blocks in the matrix (Figure 2.23 (left, center left)). In the opposite case, a matrix results in a *high* LA score if it is noisy (Figure 2.23 (center right, right)). Comparing both measures can inform the tradeoff between *fast* algorithms and *visually pleasing* reordering results.



Quality Measure *Linear Arrangement*

Figure 2.23 Examples of low/high scores for the Linear Arrangement quality criterion.

For our analysis, we obtained implementations of 35 algorithms, representative for the groups in Section 2.3.3. We used these algorithms to reorder matrices for 150 graphs, resulting in 4348 of 5250 total reordered matrices (35 ×150). The missing ones are erroneous results (e.g., not all row/column indices were returned or indexes occur multiple times), which we attribute to issues with parameters, especially problematic for Biclustering approaches (c.f. Section 2.3.3).

For each trial (i.e., reordered matrix) we measured runtime and LA score, as well as captured the visual matrix in a picture. We provide online the browsable collection of all matrices and associated measures at http://matrixreordering.dbvis.de.

Design and Setup

Algorithms We selected 35 algorithms satisfying the following criteria: (i) well-known and used in practice, (ii) available and accessible implementation, and (iii) runtime complexity is reasonable given the tested datasets.

For example, we tested R implementations for the Branch-and-Bound algorithms (`BBURCG` and `BBWRCG`, *seriation* package [BS05]), but opted to remove them due to their long runtime. Our experimentations reveal them impractical for graphs larger than 30 nodes. Table 2.1 gives an overview of the selected algorithms, the group they belong to (Section 2.3.3), and the source of their implementation.

Family	Name	Implementation
Robinsonian (R)	Hierarchical Cluster	Java [Eis+98]
	Bipolarization	Java [Hub74]
	RSeriationGW	R *seriation* [HBH14]
	RSeriationBEA	R *seriation* [HBH14]
	RSeriationOLO	R *seriation* [HBH14]
	RSeriationHC	R *seriation* [HBH14]
Spectral (S)	RCorrplotSortingAOE	R *corrplot* [Wei13]
	RCorrplotSortingFPC	R *corrplot* [Wei13]
	RSeriationCHEN	R *seriation* [HBH14]
Heuristic (H)	Row Sum (Asc)	
	Row Sum (Desc)	
	Median Iteration	
	Mean Iteration	
	V-Cycle	
Dimension Reduction (D)	RSeriationMDS	R *seriation* [HBH14]
	RSeriationPCA	R *seriation* [HBH14]
Graph (G)	Multiple-Fragment	Java [Ben92]
	Multi Heuristic	
	Multi-Scale	Java [KH02a]
	Cuthill-McKee	Java [CM69]
	Reverse Cuthill-McKee	Java [CM69]
	Degree (Ascending)	Java
	Local Refine	Java
	RSeriationTSP	R *seriation* [HBH14]
	RSeriationBEATSP	R *seriation* [HBH14]
	Sloan	C++ Boost []
	King	C++ Boost []
Bi-Clustering (B)	RBiclusteringBCPlaid	R *biclust* [KL08]
	RBiclusteringBCBimax	R *biclust* [KL08]
	RBiclusteringBCQuest	R *biclust* [KL08]
	RBiclusteringBCQuestmet	R *biclust* [KL08]
	RBiclusteringBCQuestord	R *biclust* [KL08]
	RBiclusteringBCBimax	R *biclust* [KL08]
	RBiclusteringBCrepBimax	R *biclust* [KL08]
	RBiclusteringBCSpectral	R *biclust* [KL08]

Table 2.1 Overview of tested matrix reordering implementations. The table shows (i) the algorithm group according to our taxonomy, (ii) the internal identifier and (iii) the implementation source or respective publication for our Java implementations.

Graphs To obtain a large and heterogeneous collection for graphs, we selected 150 graphs from 3 different sources and with varying characteristics (e.g. size, density). We present them below:

- 20 **real-world graphs** from the Pajek graph collection [BM98].

- 23 **test graphs** from the *Petit Testsuite* [Pet03], one of the primary benchmark suites used in the literature for comparing matrix reordering algorithms.
- 107 **random graphs** generated to control for graph characteristics such as size, density, and number of clusters. We generated graphs using the random graph generators in NetworkX [Net]. We tested the following types of graphs: bi-partite graphs, clustered graphs, graphs with small-world graphs (Watts-Strogatz), and graphs with power-law degree distribution (Albert-Barabasi).

For this analysis, we categorized graphs according to two measures: *size*: (i) *small* (25-100 nodes), (ii) *large* (100-1500 nodes) and *density*: (i) *sparse* (density of 0.05 - 0.28), (ii) *dense* (density of 0.28-0.6).

Setup We generated all trials on an Intel Core i7-2600 Quadcore (3.4 GHz) PC with 16 GB RAM (DDR3-PC3-10600) and a 120 GB SSD. The PC is operated by Windows 7 Enterprise Edition and runs R in the version 3.1.2, Java SE 7. We excluded transfer and preprocessing times from the measured execution time.

We conducted the computation with a Java program. We included 19 implementations from the R packages *'corrplot'* [Wei13], *'biclust'* [KL08] and *'seriation'* [HHB08]. For these R packages we used the R-Proxy implementation from Nuiton[2] to send and receive information from Java to and from an R server. We invoked two algorithms, taken from the C++ Boost library [SLL01], via the Java Native Interface (JNA). We implemented in java several algorithms for which we could not find any reference implementation.

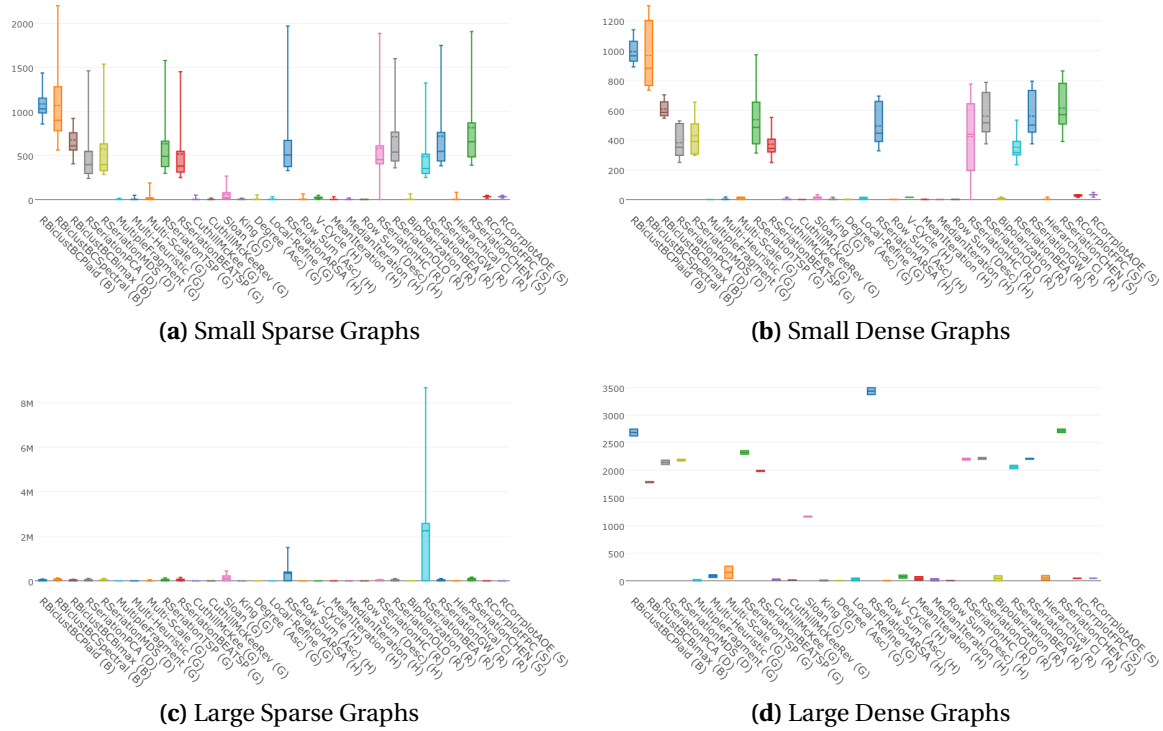
Results and Findings

Runtime Figure 2.24 shows runtime statistics in milliseconds for each of the four graph groups: *small-sparse, small-dense, large-sparse*, and *large-dense*.

Graph-theoretic algorithms (e.g., *Cuthill-McKee*, *King*, *Sloan*, *Multi-Scale*) and some Robinsonian algorithms (e.g., *Bipolarization*, *Multi-Heuristic*, *Hierarchical Clustering*) deliver results fast with $< 1000\ msec$. More interestingly, these algorithms are nearly independent from the graph topology. For example, it appears that the runtime is not influenced by variation in the degree of clustering of the graph.

R Seriation and *Biclustering* algorithms, independent from their algorithm family, tend to perform slower than graph-theoretic algorithms. This could be due to (i) particulary sophisticated implementations, and/or (ii) the data structures used. However, the R *corrplot* package is as fast as the fastest algorithms, which makes it unlikely that the used data structure (access times on the row-/column vectors and cell access) has a significant influence on the calculation time.

[2] https://nuiton.org/projects/nuiton-j2r/



(a) Small Sparse Graphs

(b) Small Dense Graphs

(c) Large Sparse Graphs

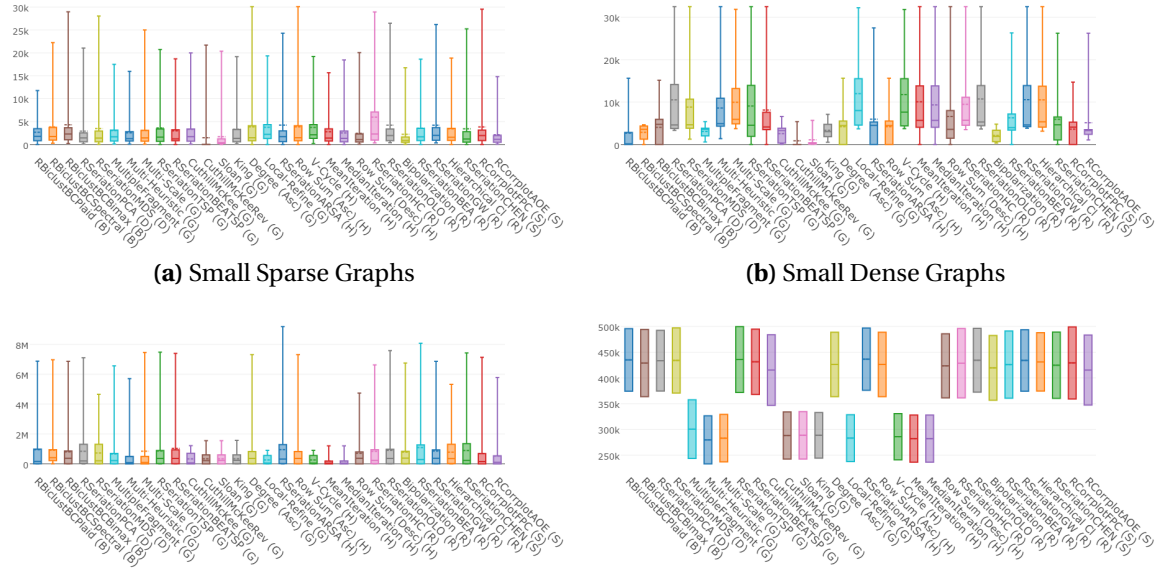
(d) Large Dense Graphs

Figure 2.24 Calculation time (in msec) for each graph category (small/large versus sparse/dense).

Runtime for large graphs (*large*) are still < 3000 $msec$. An exception are *RSeriation-BEA* and *RSeriationARSA* with a runtime about 144,000 $msec$ and 24,000 $msec$ respectively, on the "c4y" real-world Integrated Circuit (IC) network with 1366 nodes and 2915 edges.

Linear Arrangement Linear Arrangement (LA) is a loss function that refers to the minimum linear arrangement problem (e.g., described in [KH02b]). As Figure 2.23 depicts, it allows assessing the visual quality of a matrix plot: The exemplified low LA scores refer to the block pattern (*P1* and *P2*), while high scores prove to be valid indicators for noisy plots (*A1*).

Figure 2.25 depicts boxplots for our Linear Arrangement experiments under varying topology aspects: We can see that sparse graphs lead to a consistent high median LA score; however the mean scores (dotted line) and the prominent Whisker lines indicate the strong variance within the data. Noteworthy algorithms are the Reverse Cuthill-McKee and the Sloan algorithm (both graph-related algorithms), which tend to produce consistently either low scores or end up with noisy visual matrices. In the taxonomy group of large and dense graphs (Figure Figure 2.25d) we can derive similar tendencies: graph-related measures often outperform Robinsonian, Spectral and BiClustering methods.



(a) Small Sparse Graphs

(b) Small Dense Graphs

(c) Large Sparse Graphs

(d) Large Dense Graphs

Figure 2.25 Linear arrangement Scores for each graph category (small/large versus sparse/dense).

Discussion and Research Directions

The question: "What is a good matrix reordering?" does not have a unique answer. However, we consider a "bad" ordering, one that fails to reveal patters such as those desribed in Section 2.1.3, when they actually are present in the data (e.g. clusters, hubs, bigraphs).

Our empirical experience tends to indicate that a higher visual quality requires more sophisticated approaches and thus, more time to calculate. This may prove problematic in scenarios where providing rapid feedback is more crucial than displaying the highest quality result. In fact, there are many tradeoffs pertaining to the selection of a reordering algorithm. In this section, we discuss strategies to select an algorithm, explain how several of them can be parameterized and briefly discuss interactive approaches to introduce the human in the loop. We conclude by discussing the limitations of our survey and outline directions for future work.

Guidelines for Selecting an appropriate Matrix Reordering Algorithm

While ideally one would provide specific guidance to which algorithm to select and which parameter to use with respect to data and tasks, there are too many open research questions remaining to provide formal and robust guidelines at this point. Instead, we provide several insights on how to matrix reordering algorithm selection and parameter gained from our observations in Section 2.3.3, the analysis in Section 2.3.3 and our own empirical knowledge gathered by applying reordering algorithms in domain-specific applications.

Fast algorithms first—Fast reordering approaches can produce results in sub-second runtime when the data structure matches characteristics that this algorithm is optimizing for. Others are robust to certain properties of the data, making them practical to at least try first. For example, *Cuthill-McKee* (see: Section 2.3.3) or *RCorrplotSortingAOE* (see: Section 2.3.3) algorithms produce results at near-interactive processing rates, almost independently of the matrix density. However, results are tailored for a specific data structure and, when not present, these algorithms often produce anti-patterns (*A1*▨) or depict calculation artifacts (*A2*▧), such as described in Section 2.1.3. From our empirical experimenations, we note that if a fast algorithm reveals desired patterns, a more sophisticated is unlikely to improve on its quality significantly.

Heuristic algorithms offer tradeoffs—Heuristic approaches offer a good tradeoff between runtime and quality. They often produce more patterns than very fast algorithms, and improve them via multiple iterations. The freedom to interrupting them after a certain number of iterations enable to strike the balance between runtime and (potentially measurable) quality. For example, the *Barycenter* or *RSeriationARSA* algorithm (c.f. Algorithm 2.3.3) can be stop in early iterations, but generally require more to produce higher quality results. These algorithms tend to first reveal data partitioning (e.g. connected components, sub-networks), but patterns within these partitions require more iterations.

Optimizing for clusters identification—For scenarios where identifying clusters is essential, we recommend to select (hierarchical) clustering approaches, since they explicitly detect clusters and order each one individually, placing them at the matrix diagonal. A good example is *Hierarchical Clustering* (c.f. Section 2.3.3). Alternatively spectral algorithms also highlight clusters in the data set since similarly connected nodes are located close in Eigenspace. A good example is *RSeriationChen* (c.f. Section 2.3.3).

Quality takes time—If the previous algorithms fail to reveal patterns, or if the patterns produced are not matching the tasks and scenario, one may need to compromise on runtime and opt instead for remaining algorithms. Through our experimentations, we observed that Optimal-Leaf-Ordering (c.f. Section 2.3.3) tends to produce visually coherent and well organized block-diagonal forms. Note that we consider higher visual quality when local (sub-)structures and patterns are discernable in the matrix plot. As a direct consequence, algorithms attempting to optimize functions on the entire data may not be able to grasp these phenomena. For example algorithms such as BiClustering approaches often produce visually interpretable (sub-)structures if, and only if, appropriate parameters pertaining to the clusters are set. Another example are the Traveling Salesmen algorithms (c.f. Section 2.3.3), considering distances between

each pair of vertices, which often reveal local patterns (e.g. cliques or hubs) but may fail to optimize the general matrix bandwidth.

Opportunities and Future Directions

While this document describes existing solutions to reorder matrices, there are still many opportunities to improve these solutions and provide better ones. We list here some possible future work and pitfalls of our approach.

Global vs. Local Algorithms vary on their strategy to explore the search space: top-down or bottom-up. Top-down approaches focus on optimizing a global structure metric (e.g. *Multi-Scale*, Section 2.3.3), while bottom-up approches may retrieve local graph structures (e.g. *Bipolarization*, Section 2.3.3). The strategy has a direct impact on the visual results. The majority of algorithms proposed in this article are bottom-up approaches.

Hybrid approaches are an interesting future direction, where retrieving global structures can be first done in a first iteration, and other (possibly different) algorithms applied to sub-networks in a later iterations. Another interesting research direction relates to multi-scale approaches which could allow a user to retrieve results at different scales of interest (e.g. entire data, connected component, sub-network).

Similarity/Distance Calculation An important parameter for reordering algorithms, especially crucial for Robinsonian algorithms (c.f. Figure 2.3.3), is the choice of the distance metric between nodes (rows and columns). However, there is no simple method to choose a good measure given a specific graph or task. Gelfand [Gel71] notes the importance of these considerations and describes three exemplary similarity functions, which should be applied with respect to the different data domains $[0,1]$, $[true, false]$, $(-\infty,\infty)$.

Alternatively, domain-specific considerations can be included into the distance calculations, such as in Eisen et al. [Eis+98]: gene offset shifts over a log-scaled correlation coefficient are applied to analyze gene similarity. Behrisch et al. [Beh+12a] calculate text similarity on news articles and present the pair-wise comparisons in a matrix format.

More sophisticated techniques, such as the earth movers distance, or statistically inspired distance considerations (i.e., Jenson-Shannon Divergence or $\mathcal{X}^2$) cannot be found in the current literature for matrix reordering. Gregor et al.[Gre+15] recently made an attempt at empirically deriving the impact of the respective distance functions for visual tasks. He found that for feature retrieval tasks, the Manhattan Distance is a robust choice and outperforms Jensen-Shannon Divergence and even the Euclidean Distance.

Visual Patterns Assessment Several approaches are tailored to produce block-diagonal patterns (*P1*). Unfortunately, if the data does not contain such patterns, these algorithms mostly fail to reveal any pattern. More work is required to design algorithms that focus on different other patterns. A crucial research direction is to develop quantitative measure to evaluate the quality of these patterns and thus craft objective functions to optimize or assess algorithms' performance.

Limitations

This survey presents a categorization of matrix reordering approaches into reordering families, sharing similar reordering concepts. Our goal is to provide a central document where concepts from multiple disciplines are defined and related, algorithms grouped in categories and discussed in contrast to each other, and, finally, examples of visual results provided systematically.

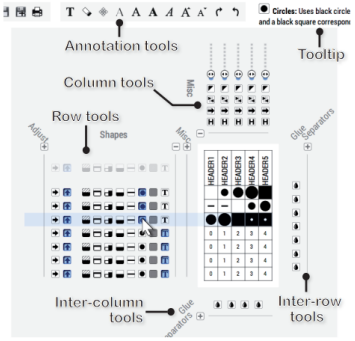
While we discussed at length several alternatives to our present taxonomy, possibly able to better capture nuances between different approaches, we finally opted to provide a categorization of reordering algorithms as simple as possible. We believe matrix reordering algorithms are a fundamental barrier for the use of matrices in practice today. By providing a straightforward classification and formulating mechanisms and approaches in simple terms, we hope to help a wide audience better understand these algorithms and integrate them in future systems and libraries.

While we gave insights in discussion on how to select algorithms for certain data characteristics or specific tasks (e.g. identifying cluster), matching systematically algorithms and tasks (for example tasks described in [Lee+06]) is extremely challenging. We decided against attempting to describe this matching formally as there are still many unknowns and doing so would require a substantial amount of future work. In particular, we do not think this is possible without developing measures to assess and quantify visual patterns produced reliably.

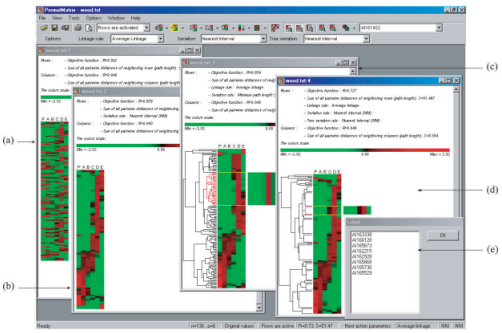
2.3.4 | Interactive Pattern Generation in Matrix-based Representations

While automatic reordering algorithms ideally take the burden off the user while producing high quality results, it may not happen often in practice. To address shortcomings of certain algorithms and enable the user to steer algorithms by making decisions at critical points, interactive reordering techniques started to appear. We point to several examples in this section.

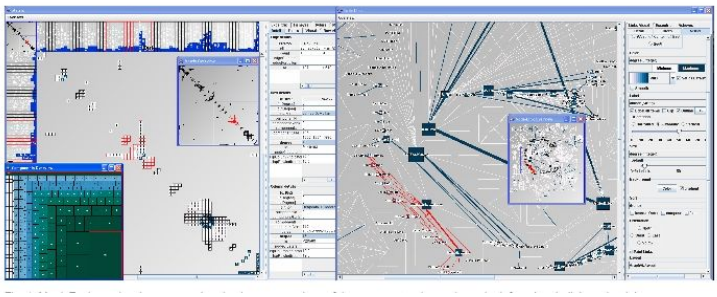
We can broadly categorize interactive reordering techniques in two categories: interactive and semi-assisted (steering). Bertifier [PDF14], TableLens [RC94] and InfoZoom [SBB96] are examples of interactive techniques, providing a user interface in which



(a) Interactive Reordering in Bertifier [PDF14]



(b) Interactive Reordering in PermutMatrix [CP05]



(c) Interactive Reordering in MatrixExplorer [HF06]

Figure 2.26 Examples of Interactive Reordering approaches.

users can manually reorder rows and columns. Compare to Bertin's initial physical device that enable users to move a single row or column at a time [Ber81], strategies used in these pieces of software provide grouping and aggregation mechanisms that enable to move sets or rows and columns at a time, decreasing the labor to reorder a matrix. In addition, Bertifier (depicted in Figure 2.26a) provides primitives to let users change the visual style of the resulting visual matrices, an important step towards communication of the results to an audience.

On the other hand, MatrixExplorer [HF06], Select Objective Measures [BW13], or PermutMatrix [CP05] provide semi-automatic approaches and enable the user to steer or guide algorithms when reordering a matrix. For example, MatrixExplorer (depicted in Figure 2.26c) enable the user to select subsections of the matrix to reorder in addition to interactive reordering of individual rows or columns. PermutMatrix provides a suite of features to enable users to apply different reordering algorithms and a set of primitives to identify, select and manipulate substructures of the matrix (e.g., clusters, cliques or sub-trees). Also, Elmqvist et al. present in [Elm+08] a scatterplot matrix for navigating in multidimensional data sets, where the order of dimensions can be modified manually or (semi-)automatically: While the user may freely change the row-/column order, also the

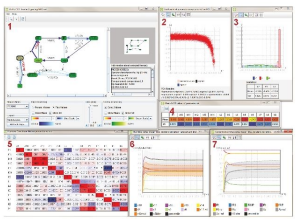
system can give impulses by informing and guiding the user with additional information about the effect of the forthcoming modification.

While these techniques can address shortcomings of certain algorithms and leverage human and computer skills, these systems are still in their infancy and rather rare in practice. We believe some of the most exciting advances for matrix reordering will occur in this space, as our research community crafts visual analytics systems that leverage user interaction and automatic algorithms. In the course of this thesis we will present our approach towards user-guided matrix reordering in Section 5.4.

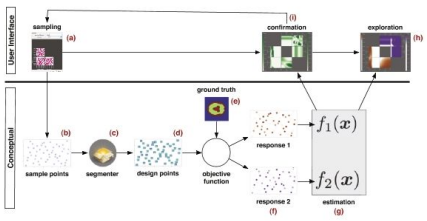
2.3.5 | The Role of Matrix-based Representations in Data Analysis Systems

In this section, we focus on the analytic purpose of matrix-based representations. Especially, we are investigating, which role matrices play in data analysis systems. Generally, we found that matrices are used (i) as powerful "helper visualizations" guiding the user in the analysis process and (ii) for the purpose of communicating insights and findings.

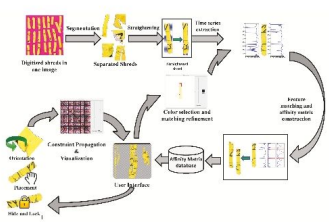
Matrix-based Representations as Auxiliary Views to Guide the Analysis Matrix-based visualizations have an *assisting character* and are used as secondary or auxiliary views to convey additional information. Figure 2.27 shows different representatives of this subcategory. The *Pearson* correlation matrix in Figure 2.27a enables the user to compare dimension correlations and helps to select interesting dimensions for further analysis in the attached/linked views [Veh+12]. Torsney-Weir et al. use in [Tor+11] matrix views represent pre-processed the data, which will be used in the subsequent analysis steps. *Deshredder* [BCR12] uses matrices as a helper visualization for finding best matches in text data sets (depicted in Figure 2.27c). Other examples for this category can be found here: [Mac+03; Ren+05; GWR09; LS09; Bez+10b; CCB12; Ung+12; Son+14; Byš+15; GLT15; Ito15; LZM15; Bre+16]



(a) Vehlow et al. [Veh+12]



(b) Torsney-Weir et al. [Tor+11]

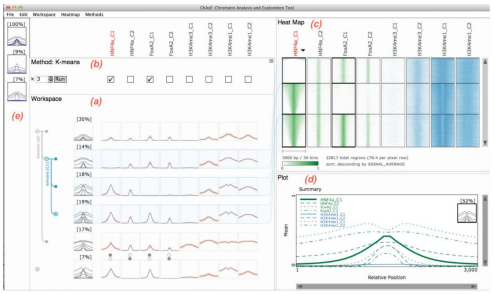


(c) Butler, Chakraborty, and Ramakrishan [BCR12]

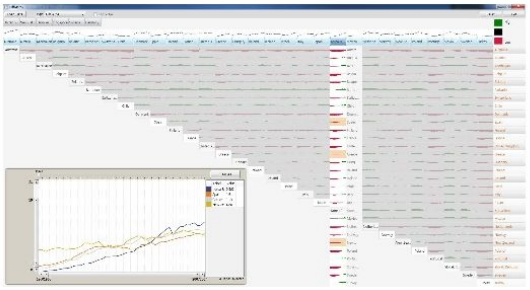
Figure 2.27 Matrices as auxiliary/helper views to guide the analysis process.

Matrix-based Representations for the Representation of Insights The second analytical purpose is to employ matrices to represent analysis *findings and insights*, i.e., this subcategory describes situations in which matrix-based representions are solely used to fully reach an analysis goal. As an example, in Figure 2.28a and Figure 2.28b the user is able to apply modifications, such as filtering, reordering, folding and unfolding on matrices and thereby reveal interesting patterns [You+13; Son+12]. Perin shows in [Per13] an variant of a scatterplot matrix, which is used to facilitate the comparison of data dimensions. The matrix-based visualization in Figure 2.28c helps the user to analyze software decompositions [BD10].

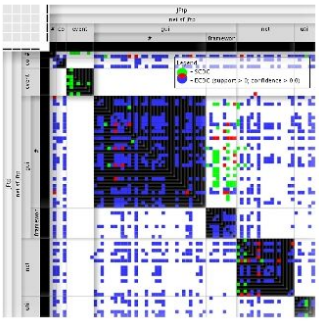
Later examples for this category can be found here: [Bac+15; CT15; DKG15; DRW15; Ko+15; Kol+15; Köt+15; Lei+15; Lub+15; Ma+15; Mor+15; Rai+15; Wu+15; Kle+16; Pap+16]



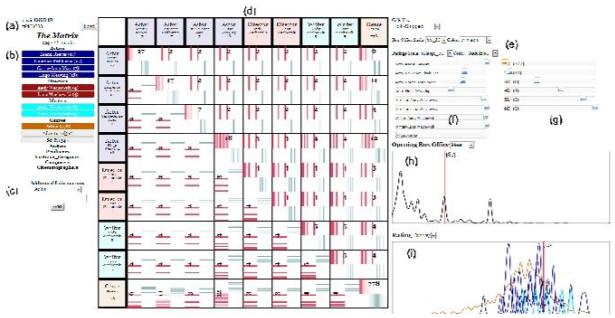
(a) Younesy et al. [You+13]



(b) Song et al. [Son+12]



(c) Beck and Diehl [BD10]

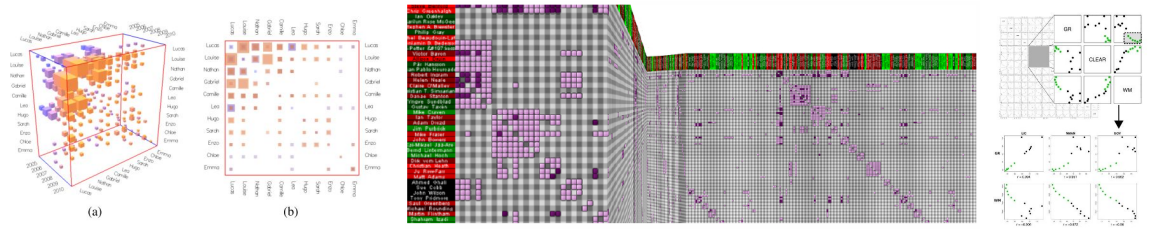


(d) Perin [Per13]

Figure 2.28 Matrix-based representations to convey/represent insights and findings.

2.3.6 | System Integration of Matrix-based Representations

In this category we are distinguishing matrix-based visualizations, which are used along with other visualizations or stands out as the primary visualization and interaction component.

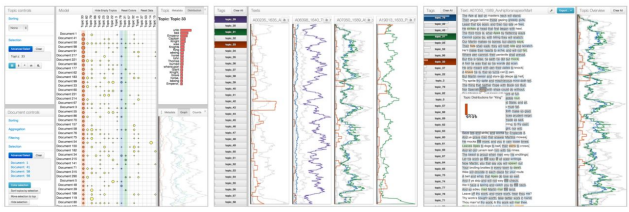


(a) Bach [Bac13] **(b)** Elmqvist, Dragicevic, and Fekete [EDF08c] **(c)** Hao et al. [Hao+07a]

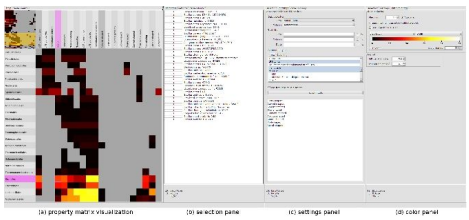
Figure 2.29 Matrix-based Representations as the Primary Interaction Component.

Primary Interaction Component The *Primary Interaction Component* subcategory consists of systems and tools, in which the analyst uses matrices as the primary visualization and interaction component for analytic reasoning. In these cases matrices are often used to convey an overview of the data. Particularly this is interesting, since interaction techniques and the expressiveness of the visualization must be tailored towards the application needs. Figure 2.29 shows examples of these systems. As an example, Bach present in [Bac13] a visual interactive filtering metaphor to navigate within space-time cubes. The cube can be transformed –or unfolded– from its three-dimensional representation to small-multiples of 2D matrices showing, e.g., the respective year or investigated category distribution. Elmqvist, Dragicevic, and Fekete show in Mélange an interaction technique to navigate through large visual spaces, which bases on a space deformation technique that folds 2D space into 3D in order to bring several regions of interest into focus while preserving the awareness of the intermediate context. Hao et al. use extensively the brushing and linking metaphor in [Hao+07a] to depict data distributions in scatter plot matrices. Other examples of this category can be found in [WR04; HFM07; CMP09; BTK11b; DWW12; HSW12; KS12; PGU12; Sil12; Kle+16]

Secondary Interaction Component The *Secondary Interaction Component* subcategory contrast the previous one in that the combination of a matrix and further visualizations is pursued. While matrix visualizations are often used in overview-related tasks, additional linked views allow for a more detailed analysis. In Figure 2.30 some examples of this category are shown, in which matrix visualizations are interlinked with other –potentially primary– views (e.g., text views, geographical maps or small multiples views). For example, Alexander show in [Ale13] a text visualization in which a reorderable and linked matrix view represents the document-topic relationships for one entire text corpus. In this case, the matrix is used to pinpoint the user to interesting analysis questions. In [BW13], Brakel and Westenberg use a color-coded matrix to explore relations between annotation terms in biological data sets. By reordering the matrix patterns become visible, whose selection can be used to construct new “focus” matrices on the fly. Bremm and Hamacher use in

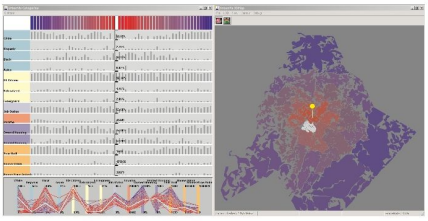


(a) Alexander [Ale13]



(b) Brakel and Westenberg [BW13]

(c) Bremm and Hamacher [BH11]



(d) Chang et al. [Cha+07a]

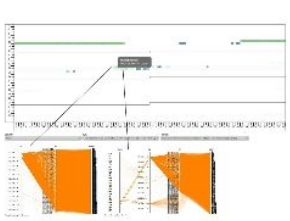
Figure 2.30 Matrix-based Representations as the Secondary Interaction Component.

[BH11] a matrix minimap to depict partial tree similarity and in *WireVis* [Cha+07a], Chang et al. present multi-view system for exploring wire transactions, where a heat map depicts the relationship between keywords and bank accounts in order to gain an overview and enable the comparison of patterns. Other examples of this category can be found here: [AH04; Cha+07b; Elm+08; CDS09; Mey+10; Mig10; Ber+11b; Veh+11; AWD12; Sed+12; Yua+13; CT15; DRW15; LZM15; Ma+15; Mor+15; Yal+16]

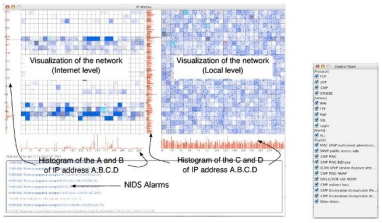
2.3.7 | Result View Integration

The result of interactions can be displayed in different ways. The effects can be shown in the same view, in another existing view, or even in new visualizations which are created in order to reflect the outcomes. In contrast to the *Matrix Appearance* categories which describe the input interface of a visualization, the result view defines how the output interface looks like.

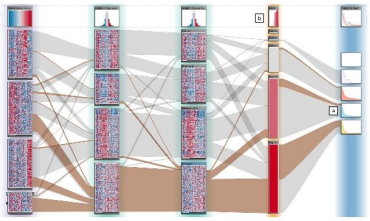
Additional Result Views This subcategory, also depicted in Figure 2.31, shows systems and tools in which the interaction results are presented in new views. For example, in Figure 2.31a, Lamagna use the user's matrix cells selection to invoke the creation of a separate parallel coordinate visualization for further exploration. In [KOK05], Koike, Ohno, and Koizumi allow the user to select dimensions shown in a matrix representation as dependent and independent variables to generate a model space visualization in a new view. Further examples for this subcategory can be found here: [EMR06; GWR09; Fri10; CKX11; CCB12; LZM15; DKG15; Kol+15]



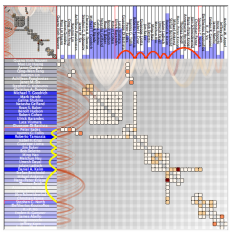
(a) Additional Result Views in [Lam11]



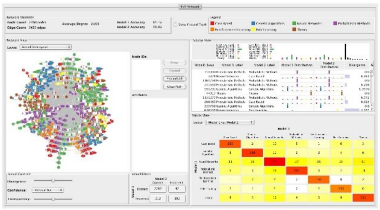
(b) Additional Views in [KOK05]



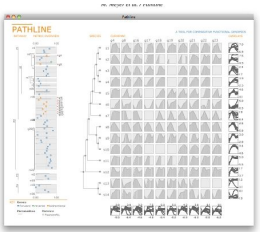
(c) Result Integration in [Lex+12]



(d) Result Integration in [HF07]



(e) Coordinated Result Views in [Sha+11]



(f) Coordinated Result Views in [Mey+10]

Figure 2.31 Examples of the *New View* subcategory.

Result Integration In this subcategory, modifications and visual enhancements are only applied in the matrix visualization itself. The primary approaches in this field are changing the cell representation, such as in [AAB07; May+11; Mac+03], merge and split [HFM07], or filter [Lex+14]. Another important approach is to overlay the analysis result on the visualization itself. In Figure 2.31d and Figure 2.31c, two examples of result overlays are shown. For example, Henry and Fekete use in [HF07] color-coded arcs to depict transition routes through the network. Other examples, for this category are: [KOK05; Cha+07b; HP11; May+11; MGW11; DWW12; Kle+16; AAB07; May+11; Mor+15; Wu+15]

Coordinated Result Views This subcategory describes systems in which user interactions invoke the update in other subparts of the system, i.e., multiple linked views are coordinately used to depict the analysis result. To give an example, in Figure 2.31e, a multiple connected network visualizations –among others a matrix– are used for comparing uncertain graphs. By selecting a cell in the matrix view, the node-link diagram and the network statistics table will be filtered to show the nodes of interest [Sha+11]. Also in [Mey+10], a coordinated view system is presented where the matrix interaction invokes updates on the connected pathline visualization and vice versa. Many other examples for this category state the generalizability and power of this approach: [AH04; AWD12; And+10; Bez+10b; Bre+11; Cha+07b; CDS09; EMR06; Goo+05; HF06; Ing+10a; Ko+12; May+11; Mey+10; Per13; Ren+05; Sed+12; Sil12; Tor+11; Veh+12; Via+10; You+13; Bre+16; Pap+16; Yal+16; CT15; GLT15; Lub+15; RPC15; Tri+15; WL15]

3 | Visual Interactive Support for Exploring Matrix-based Representations

Contents

This chapter of the thesis collects all contributions focusing the visual interactive support for exploring matrix-based representations. After a motivation (Section 3.1) and an excerpt of the related work that relates particularly to the content of this chapter

in Section 3.2, the overview section (Section 3.3) will outline the specific research goals and relates the research contributions, -questions and -vision.

The core contribution of this chapter lies in the visual justification that an enhanced cell design for matrices inherently enhances the analytic expressiveness and usefulness (cf. Section 1.2) of matrix-based representations.

This chapter is based on the following publications:

"Visual Analysis of Sets of Heterogeneous Matrices Using Projection-Based Distance Functions and Semantic Zoom"

Behrisch, Davey, Fischer, Thonnard, Schreck, Keim, and Kohlhammer. Computer Graphics Forum, Eurographics Conference on Visualization (EuroVis 2014), The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd., 2014, 33, 411-420. [Beh+14b]

"Visual Comparison of Orderings and Rankings"

Behrisch, Davey, Simon, Schreck, Keim, and Kohlhammer. Proceedings of the EuroVis Workshop on Visual Analytics, The Eurographics Association, 2013. [Beh+13]

"Matrix-Based Visual Correlation Analysis on Large Timeseries Data"

Behrisch, Davey, Schreck, Kohlhammer, and Keim. Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (Poster Paper), Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on, Institute of Electrical & Electronics Engineers (IEEE), 2012, 209-210. [Beh+12b]

Parts of the Motivation Section Section 3.1 and the Overview Section Section 3.3 are adapted and/or taken from the text/figures I have written/developed for the German Research Foundation (DFG) research proposal "Transregional Collaborative Research Center 161 Quantitative Methods for Visual Computing."

3.1 | Motivation

In recent years not only the *data velocity* was increasing at an extreme pace, but also the *data complexity* [Z+11]. While in the former decade the exploration of a single (relational) data aspect stood in the focus of most analysis scenarios, today the *multifaceted* and *interrelated analysis* of distinct (relational) aspects describes the analysis level.

Research Objectives: This data analysis complexity increase leads to novel research questions for the Information Visualization community, as well as the Visual Analytics domain:

1. How can we support the user in the formation of mental models of the investigated data spaces?
2. How can we visualize and navigate in large (relational) data spaces?
3. How can we make relational visualizations more scalable wrt. the number of simultaneously displayed data aspects?
4. How can we guide the user in the exploration process with novel search and query mechanisms?

Visualization and navigation of large data spaces can support the formation of a *mental model* of the data space and better address user information needs [WR09; Zha08]. To date, many approaches have been proposed to visualize similarity and group relationships using 2D layouts to compare data [Gle+11b]. 2D layouts can show data items e.g., as data points [AS94], using glyphs [War02], or an information landscape [Wis+95]. For matrix representations only very few works have focused on extending the visual scalability and efficiency. We summarize the contributions in this field in Section 2.3.2 and develop our own domain dependent, as well as generalizable, glyph design for matrix representations.

Most visualization techniques typically rely on a given similarity method for data, or support adapting the similarity measures by manual filtering or selecting of attributes. For example, in [EDF08b] an interactive approach for comparing pairs of dimensions in a high-dimensional data space is proposed. The technique supports navigation by extrusion-based transitions between views. It is interactive in nature and does not support guiding users towards interesting views. In general, purely interactive methods for navigation, especially in large data spaces, may not be scalable, as the user may miss relevant views due to a too large navigation space. For certain visual representations such as Scatter Plots [WAG06] or Parallel Coordinate Plots [DK10b], heuristic measures have been proposed which can estimate the potential interestingness of data views and support navigation in large analysis spaces. Our approaches for an automatic pattern analysis of matrix-based representations will be presented in Chapter 4 and Chapter 5.

A further significant challenge is the exploration and querying for patterns, connections, and correlations in large data sets. Patterns are seen as the gold nuggets of the data [FPS96a], but are mostly hidden in the dimensionality. Their discovery process is subject to an exploratory search. One approach, is to have *Overview+Detail* systems that lead the user in an overview to areas of interest and let him/her explore these areas with drill-down mechanisms. Overview+Detail systems are an established and well understood method, but may end up to be expert systems, restricted to specific data set characteristics or user interactions, if the applied data abstraction is not chosen generally.

In recent years many domains have emerged in which the *comparative analysis* of distinct relational data aspects became relevant. For example, to monitor computer network traffic a dynamic set of hosts and their peer-to-peer connections on different ports must be analysed. Another example would be the analysis of social networks in which actor influences and group memberships may change over time.

Existing techniques, see [Lan+11] for an overview, generally support the display of a single, static graph. However, graphs such as computer or social networks may *change* over time. Many analysis tasks are focused on exploring that change; e.g. the comparison of a series of *snapshots* of a network over time. This gives rise to a matrix *comparison problem*. This problem is particularly challenging since both the edge sets and the node sets may change, yielding graphs, and thus matrices, of different sizes.

3.2 | Related Work

Interactive features may improve the usability of matrix visualization when specific aspects, such as groups or connectivity, should be explored in the data [HFM07; HF06]. Relational data, such as computer or social networks, can be modelled as graphs. These graphs can be visualized as matrices by simply using their adjacency matrices. The matrix cells can be coloured to show binary, categorical or continuous attributes for each edge, e.g., the edge weight [Ber81, p. 33]. Matrix visualizations are particularly suitable in cases where the associated graph is dense [GFC05].

Static and One-Dimensional Matrix Data. Matrix visualizations provide a highly scalable visual representation of graphs [Lan+11; GFC05]. They can reveal important aspects of graph structure if they are appropriately sorted and rendered. However, matrix representations are less intuitive than node-link diagrams, thus they need to be supplemented by additional visualizations and interaction techniques to improve understanding. In [HFM07], matrix visualizations and node-link visualizations are combined in an interactive system. The matrix is used to provide an overview representing very dense areas of the graph, and a node-link view shows details for selected parts that are globally sparse. Semantic zoom interaction can help navigate matrices which do not fit into the available

screen space. In [Elm+08; AH04] zooming and dynamic aggregation techniques support the navigation process in large matrices. Matrices, and thus matrix visualizations, occur in many real-world analysis tasks. Matrix visualizations have been used in the analysis of social networks [HF06], and gene regulatory networks [DWW12]. Time series can efficiently be summarized in a matrix visualization [Sip+12]. Visualizations of similarity matrices are frequently used in the analysis of non-numeric attributes, such as in the pairwise comparison of text documents [Beh+12c].

Time-Dependent and Multivariate Matrix Data. Many analysis tasks involve multiple, heterogeneous matrices. For example, a social network is a time-dependent graph whose nodes correspond to entities and edges and/or their attributes corresponds to relations such as friendship or a message (count). Both, nodes and edges may change over time. Each attribute dimension yields a matrix with a single value/dimension encoded in each cell. Each time step may give rise to a different attribute matrix. Many matrix visualizations were developed for one-dimensional and static matrix entries and do not support dynamic and complex matrix data well. One approach to handling time-dependency in graphs is [Bur+11], where graph states are represented as consecutive narrow stripes, in which vertices are arranged vertically on each side. Directed edges connect vertices from left to right to show the graph evolution. In [Bre+10], the interactive visualization of pairs of matrices was addressed. Specifically, one matrix contains weight values and the other contains target values in a correspondence-matrix representation of molecular data, and interaction allows cross-filtering in both matrices. In [Beh+12a] time-series data is presented in a triangular matrix, where the matrix cells are statistical aggregates over all possible subintervals.

Sequential and Non-Sequential Data Comparison. Much work exists that studies visually analyzing and comparing sequential (ordering, ranking) data. One instance of this sequential data is the computed linearization of a matrix, which sequentially aligns all vertices in a graph. The notion of sequential data per se is very broad and comprises many applications. The article of Gleicher et al. [Gle+11a] surveys and structures the solution space for visual comparisons of different data types.

Generally, time series are an important instance of sequential data. Time series visualization is concerned with visual mappings for series of measurements, typically given by quantitative, equally-spaced consecutive values [Aig+11]. The comparison of two or more sequential data sets is a key problem in many applications. In fact, many time series visualization techniques were designed for comparison tasks, such as dense pixel-based approaches for comparing large numbers of time series [KAK95]. The elements of a series or sequence can also be symbolic, as e.g., in DNA sequences. The analysis of sequences

of values may include relationships among them. An example are sequences of email messages sharing reply/forward relationships [Ker03].

Techniques exist which allow comparing data which is inherently non-sequential, by finding a linear mapping of data elements, on which then sequence visualization can be applied. Examples include the TreeJuxtaposer [Mun+03] system, which compares pairs of hierarchies side-by-side by finding correspondences between tree nodes mapped in sequential order (e.g., by a dendrogram). Another example is given in [HW08], where pairs of hierarchies are compared by linear (icicle) mappings with bundled connectors showing element relationships. A further example is the TimeArcTrees [GBD09] approach for comparing sequences of directed graphs. It is based on a linear mapping of nodes, a sequence of which is shown with nodes aligned for comparability.

In order to compare different matrix ordering solustions, we are interested to compare for differences in the positions of elements among sets of sequences. Our approach, presented in Section 3.4.2, is inspired by the Scatter Plot Matrix technique [Cha+83], allowing comparing pairwise combinations of variables in high-dimensional data. Matrix structures have been exploited previously for comparison of relational data, e.g., in [BN11; GHS10; SM07]. Small-multiple views of graphs for comparison based on clustering and projection have been proposed in [LGS09]. In Section 3.4.2 we present an approach that combines a matrix visualization with a custom glyph, based on a radial network layout, to compare the differences among pairs of sequences with permutations of its data elements.

Text Data Comparison. In recent years, visualization of text data has been gaining increased interest by researchers, who are developing techniques for efficient display of document collections, as well as single documents. For example, the visual analytics tool *VISRA* [Oel+10] combines readability feature selection with document visualization techniques based on *Literature Fingerprinting* [KO07], *TileBars* [Hea95] and *Seesoft* [ESS92] to evaluate the readability of the input text. In the domain of news analysis, several tools exist that deal with summarization and visualization of news content. Newsmap [Wes12] is a well-known treemap visualization of data gathered by Google News. Other popular news aggregators include Yahoo! News and Europe Media Monitor [EMM12]. The TextMap website, based on Lydia [LKS05], is an entity search engine, which provides information about people and places extracted from the news sources. These systems have limited visualization capabilities that would allow the user to understand the content differences among different sources that provide news reports on the same real world event. In the area of knowledge discovery and data mining, ongoing research efforts exist that deal with meme-tracking [LBK09] and refining causality [Sno+11]. In this field, the main goal is to find out how the information propagates through networks and how network processes cause a specific behavior in the network, by analyzing appearance of short phrases in document nodes. Researchers working on web indexing and crawling have developed

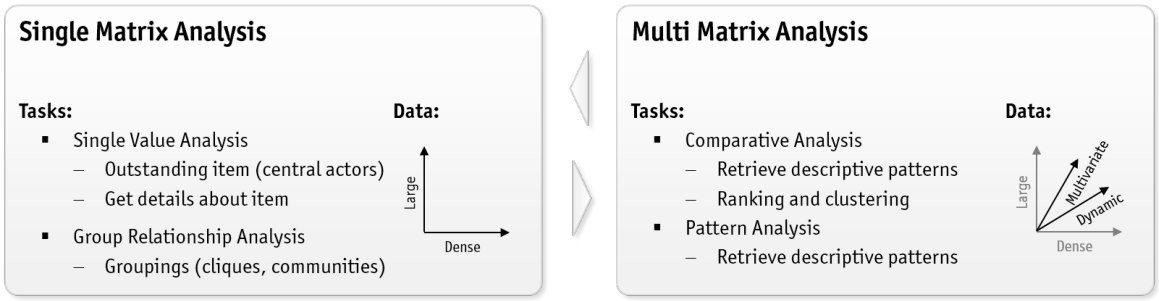


Figure 3.1 Tasks for the Single Matrix Analysis and Multi Matrix Analysis. Both analysis levels are naturally intertwined, since e.g., ranking and clustering questions require the definition of similarity which is inherently dependent on a single matrix's ordering.

methods for identifying near duplicates [MJS07], i.e. redundant web documents that differ only in a small portion.

3.3 | Overview

This chapter presents visual interactive techniques that can be used to explore and navigate in large matrix spaces. As already noted in Section 1.1 we are conducting matrix research from two intertwined perspectives: (a) single matrix analysis and (b) multi matrix analysis.

Figure 3.1 depicts exemplified analysis scenarios and tasks for single, respective multi matrix analysis. Single matrix tasks mostly relate to the analysis of distinct matrix dimensions (indices) for their special characteristics, such as being a central actor in a social network. Single matrix analysis tasks are inherently linked with two research questions: (1) How can we support the user in the formation of mental models of the investigated data spaces? And (2) How can we make relational visualizations more scalable wrt. the number of simultaneously displayed data aspects? We are approaching these research questions in Section 3.4, where we investigate interactive and static glyph-based visualizations. We are proving the usefulness of these approaches in Section 3.6.

One central objective of this Ph.D. thesis is to enlarge the scope of matrix research towards dynamic and heterogeneous matrix analysis scenarios (c.f. Chapter 1). Figure 3.1 (right) depicts the second set of analysis task relating to this comparative analysis aspect. While we are focusing on the purely automatic analysis approaches in the following Chapter 4, we are showing here in Section 3.5 Overview+Detail approaches for the exploration and navigation in large matrix spaces (c.f. Research questions in Section 3.1). For this reason, Section 3.5 presents an overview-first approach to explore large sets of matrix visualizations, accompanied with a semantic zoom technique to show the user details wherever needed. Section 3.6.2 shows the generalizability of this metaphor in another

context and applies the techniques to a visual correlation analysis scenario. Section 3.4.2 and Section 3.4.3 function as application scenarios presenting the developed techniques and their usefulness in their respective domain.

Although most of the developed techniques work for scenarios in which multiple matrices are involved, many of them are still applicable to single matrix analysis scenarios or nicely integrate into multi-matrix analysis workflows.

3.4 | Multivariate Data Analysis with Matrix-based Representations

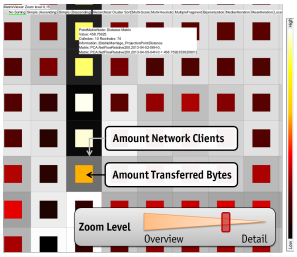
During our studies we found that the visual readability of a glyph matrix decreases if all matrix cells are allocated with (complex) glyphs. While this seams to be a general –and not yet validated– assumption, we tried to mitigate this effect by increasingly augmenting the information content in a glyph matrix depending on the zoom level. Our experiments show that on the one hand the complexity of the visual glyph highly influences the readabiltiy of the matrix, but on the other hand contributes to the expressiveness of matrices.

3.4.1 | Multi-Dimensional Data Glyphs to Support Visual Comparison Tasks

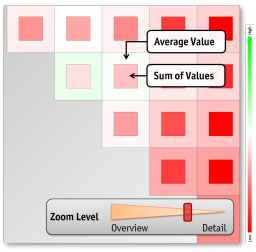
In many data analysis scenarios sets of measures have to be compared against each other. While, in general, data abstractions and aggregations help to fulfill this task, there are several situations in which multi-dimensional attributes can and should be displayed simultaneously, i.e., whenever the depicted values contribute to each other or contain a natural interrelation. We are showing one example for a time-series correlation analysis with varying offsets in Figure 3.2. Another example can be found in Figure 3.7e, where the distance value composition will be focused. In Section 2.3 we collected the related work for matrix visualizations with non-uniform and heterogeneous matrix cell representation.

Inner-/Outer Rectangle Glyph In several investigated application scenarios we are applying an inner-/outer rectangle glyph, as depicted in Figure 3.2, to represent multiple values simultaneously. The glyph allows depicting a two dimensional feature relationship by color-coding an inner and outer rectangle. If the size of the inner rectangle is considered variable, then a third feature dimension can be mapped.

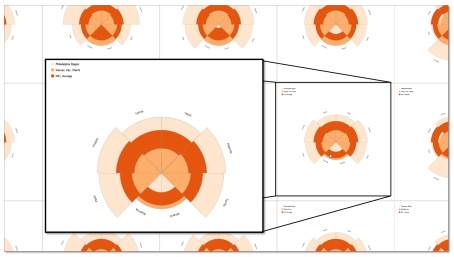
As we can see from Figure 3.2 distinct color mappings can be applied on inner-/outer rectangle glyph to support the exploration process. In Figure 3.2 (b) we used a consistently retained heat color coding for the representation of the distance score, where the color naturally corresponds to the semantic color of fire temperature and contrasted it with a



(a) Inner-/Outer Rectangle Glyph for the VAST 2014 Challenge Data.



(b) Inner-/Outer Rectangle Glyph for Time-Series Correlation Analysis.



(c) Sunburst Glyph for a eight dimensional feature vector comparison.

Figure 3.2 Glyphs Designs for Matrix Cells.

white-to-black color map for depicting a secondary relationship (the impact of the penalty function on the distance calculation; cf. Section 5.6). On the other hand, in Figure 3.2 (c) we are using a bipolar red-white-green color map twice to visually reference the zero correlation point of the investigated measures.

Sunburst Glyph For application scenarios in which more than two (to three) features should be compared we developed a sunburst glyph as visually depicted in Figure 3.2c. For each of the feature dimensions an arc is rendered to show the two comparison dimension values. In the depicted use case we comparing NFL rankings and statistics over time; therefore we additionally encode the season average for the respective feature dimension within the arc.

3.4.2 | Ranking Glyphs to Support the Visual Comparison of Matrix Reorderings

In many data analysis scenarios, sequentially ordered (or ranked) data needs to be understood and compared. Ranking information is essential in applications such as multimedia search where retrieval rankings need to be inspected; alignments of gene sequences in bio-molecular applications; or for a more abstract example, considering the permutations/reorderings of rows and columns for purpose of matrix visualization. In each of these examples, often many different orderings of a given data set are possible. E.g., a search engine may produce, based on different user parameterizations, different rankings. A relevant problem then is to understand the commonalities and differences of a potentially large set of rankings. E.g., finding global or partial orderings in which different ranking or sorting algorithms agree can support the certainty in the respective ranking by the user.

We developed a *ranking glyph*, as depicted in Figure 3.3, for comparing sets of rankings. This glyph can be used effectively in a matrix layout to foster the visual comparison of

sets of rankings, allowing spotting commonalities and differences among each possible pair of rankings. The ranking glyph in turn is defined on a radial node-link representation which allows effective perception of agreements and differences in pairs of rankings. We show how we apply our approach to different use cases in Section 3.6, where we also demonstrate the effectiveness of our approach in spotting patterns of similarity and differences in sets of matrix reordering results.

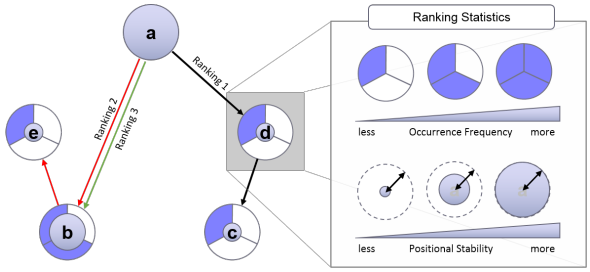


Figure 3.3 The *Ranking Comparison Glyph* allows the user to focus on one specific ranking result comparison (1:1 comparison). *Data:* $[a,d,c]$, $[a,b,e]$, $[a,b]$

Ranking Comparison Glyph (1:1 Comparison)

The ranking comparison glyph serves to identify the consensus, respectively disagreement, between pairs of rankings. Figure 3.3 depicts its design. A clock-wise circular layout of nodes encodes one selected baseline ranking. The nodes are positioned according to their index position in the ranking. Additionally, black arcs represent the base ranking sequence. A second ranking to compare against is then overlaid by inserting red arcs into the base ranking glyph. We assume the element sets of the rankings to largely overlap, but there may be elements present in only one ranking. To ensure comparability, we position all nodes that occur in both ranking sets to the position given by the baseline ranking, whereas additional nodes from the comparison ranking(s) are inserted at the end of the baseline ranking. As a result, the structure of the glyph arcs are a visual indicator for the degree of agreement between the two rankings. More rankings can be displayed on top of the base ranking, each resulting in distinctively colored edge sets (Figure 3.3 exemplifies a third ranking with green arcs). Considering Figure 3.3, the rankings differ (a) in their retrieved result list size (the black-colored ranking comprises three items, the green-colored ranking two) and (b) in the ordering ($a-d-c$ versus $a-b-e$).

We visually encode additional information regarding ranking positions and occurrence frequency in the nodes of the glyph. (1) Most use-cases require to assess the amount of occurrences for one specific result item among all investigated ranking lists (e.g., found in every/none/some of the investigated ranking list) and (2) the user wants to investigate the stability regarding positional changes (e.g., found always on first position). Thus, we encode the agreement on the position for this specific item among all investigated ranking

lists in the glyph. As Figure 3.3 depicts, for example node *d* and *c* are found only in one of the selected experiments. Hence, a pie-chart like metaphor represents this aspect. The more rankings are under investigation the smaller the portioning of the pie-chart. For demonstration purposes, we are adding the ranking result $[a, b]$ to the example above. As Figure 3.3 then depicts, node *d* was found in one of the three selected experiments.

In addition to that, the positional agreement is encoded by the diameter of an overlay on top of the pie-chart. For example, the item *a* is ranked by all three selected rankings on the same position. Hence, the diameter is 100% of the node's size and explicitly hides the double-encoded occurrence information. On the contrary, only two rankings disagree on the position of *b*, thus leading to a smaller (66%) overlay.

Ranking Comparison Matrix (1:N and N:N Comparison)

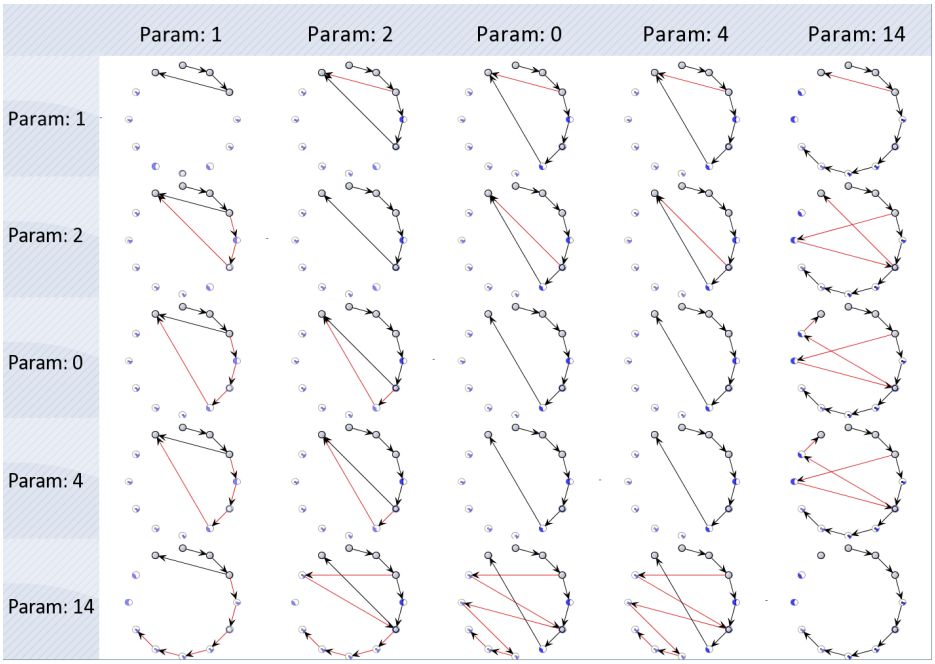


Figure 3.4 The *Ranking Comparison Matrix* allows the user to get an overview about various ranking results (N:N comparison). Furthermore, it allows the comparison of one ranking to several other ranking (1:N comparison).

A matrix of ranking comparison glyphs facilitates the 1:N and N:N comparison tasks, similar to a Scatter Plot Matrix. The vertical axis spans the space of base rankings, over which all other rankings in the data set are overlaid each one along horizontal direction. Along each row, the same baseline ranking is compared against all other rankings, as Figure 3.4 illustrates.

Each cell in the matrix depicts a ranking comparison visualized by a circular alignment of nodes (or result set items). One ranking result is illustrated by an ordered set of directed

edges connecting all result set items in the network. Ranking results are colored distinctively and mapped to directed edge layers. The ranking matrix can be sorted according to specific criteria. The current implementation sorts similar ranking comparisons to the upper left corner by considering the amount of reoccurring items among the two selected rankings. Further sorting approaches, e.g., considering inter-comparison of edge crossings could be useful.

3.4.3 | Text Glyphs to Support the Visual Comparison of Text Clusters

The News Auditor publication [Beh+12c] was the result of my Master Thesis. The glyph design, the system and the use cases were developed during the course of my master thesis and may thus not be considered as a dissertation contribution. We decided to insert the content in this chapter, since it contributes to the overall glyph-matrix storyline/contributions.

In recent years, the quantity of text content –for example– generated by news agencies and blogs is constantly growing, making it difficult for readers to process and understand this overwhelming amount of data. Online news aggregators present clusters of similar stories in a simple, list-based manner, where the most important article is shown first, while all the other similar articles appear below as hyperlinked headlines. This layout makes the user unaware of the content differences between text news articles, thus making it very difficult to get a comprehensive picture. Understanding what was changed, how, when and by whom, would lead to new insights about the content distribution over the Internet and help in dealing with the news overload problem.

We developed a *text thumbnail glyph* that depicts the structural appearance of text documents in a document thumbnails fashion. It enables the analyst to compare text documents by highlighting paragraphs of the text that were copied, modified or repositioned by different sources. As Figure 3.5 depicts we integrated the text thumbnail glyph in a matrix display. After getting an overview of the news cluster from the overview matrix, the user can choose to expand one or more rows to explore the structural features of the documents and their differences. As an example, the rows with high overall similarity scores in a few or all documents can be regarded as suspects for plagiarism. We adapted the fish-eye table exploration technique [RC94] for a seamless integration of details-on-demand in the exploration process. Expanding a row in the overview matrix reveals our text comparison glyph which follows the visual guidelines presented in [ESS92]. It visually encodes the sentence- and paragraph structure, as well as their textual similarity in comparison to the row's pivot document. The thumbnail width is fixed to allow a comparison of the news articles' text length. Likewise, the sentence bar's length corresponds to the amount of characters in this sentence. The paragraph boxes are determined by the amount of space required by all paragraph sentences, thus leading to a bottom-up layout approach of the

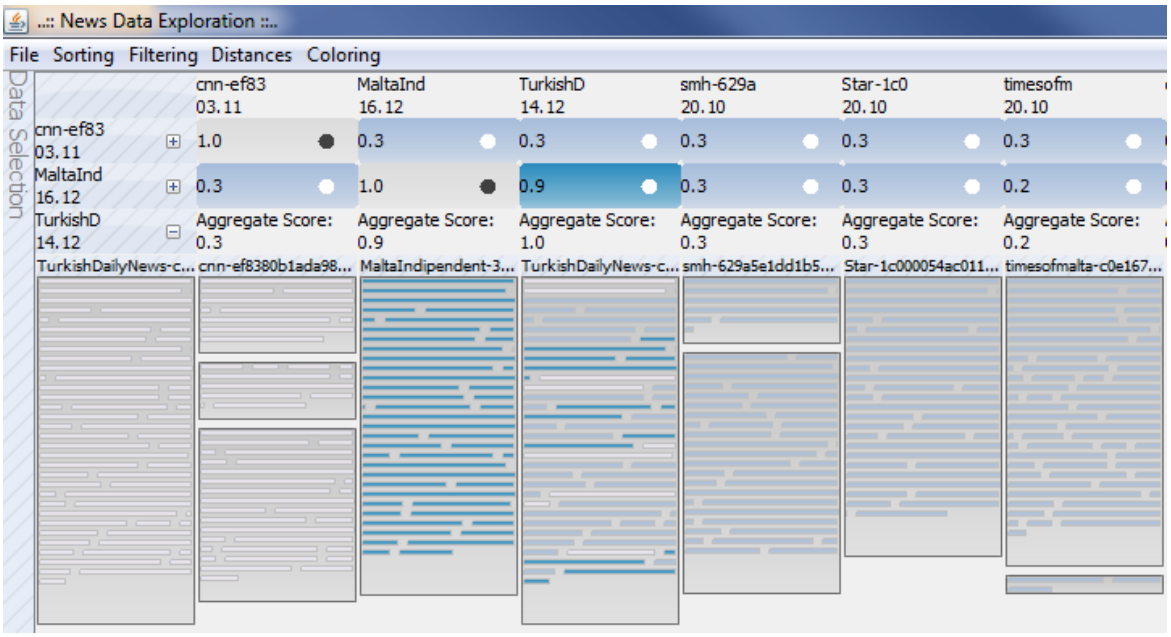


Figure 3.5 A structural comparison of text documents can be facilitated with the text thumbnail glyph. It provides a visual comparison of differences between selected articles on the paragraph level.

document thumbnails. The coloring stays consistent with the overview, thus justifying the overview's aggregate scores.

In the Application Section 3.6.3, we present an integrated visual comparison system, called *News Auditor* , that allows the user to compare text articles that belong to the same story and understand the differences at three levels of detail: (a) overview corpus level, (b) structural document level and (c) detail text insertion/deletion level.

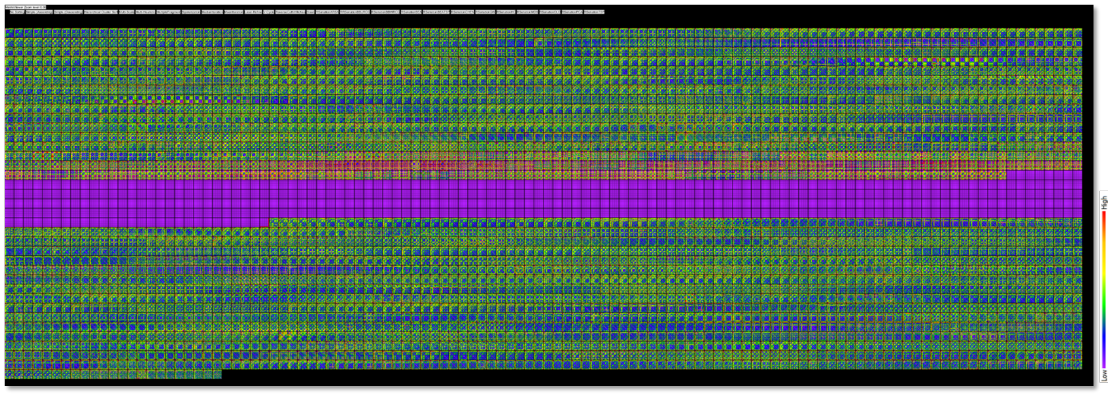
3.5 | Visual Exploration and Navigation in Large and Heterogeneous Matrix Spaces

The Visual Information-Seeking Mantra *Overview first, zoom and filter, then details-on-demand* described by Shneiderman [Shn96] is often adopted in information visualization, whenever multiple instances –in our case matrices– should be investigated.

While our research shows that the Visual Information-Seeking Mantra is applicable and beneficial for matrix exploration, we also applied the alternative bottom-up exploration approach *Search, show context, expand on demand* [VP09; Kai+15], as presented in Chapter 5.

3.5.1 | Small Multiple Displays for Exploring Large Matrix Spaces

The classic overview-first approach in information visualization is to show many views simultaneously. Figure 3.6 adapts this idea on two example use cases: (a) the soccer analysis use case, described in Section 5.8.2 and the VAST Challenge Data set, described in the same Section 5.8.2.

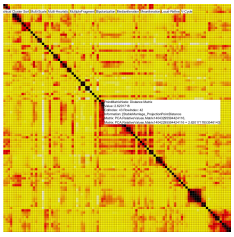


(a) Small-Multiples of Matrices with the same dimensionality.

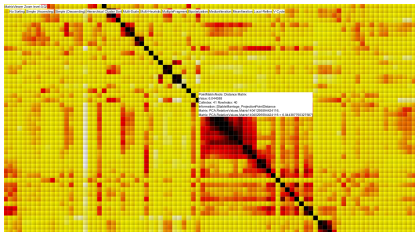
(b) Small-Multiples of Matrices with a differing dimensionality.

Figure 3.6 Small Multiple Displays for Matrix Analysis: The left Figure depicts the soccer matrices use case with 4,127 simultaneously shown matrices and the right Figure shows an overview of the VAST Challenge 2013 data set with 120 simultaneously shown matrices (c.f. Section 5.8.2).

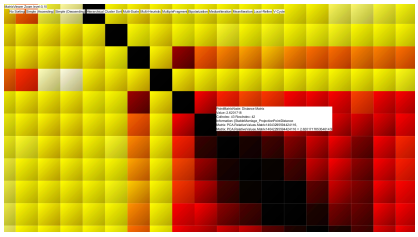
As one can see for matrices with the same dimensionality the color encoding helps to find outstanding –since visually different– matrices. In cases where the matrix dimensionality varies, even structural matrix changes can be easily perceived. Small multiple displays as depicted in Figure 3.6 help to get an overview of the visual characteristics of the dataset, i.e., amount and differences between groups. However, these views tend to be visually cluttered, such that a visual inspection is mostly only possible with more sophisticated analysis, visualization and interaction methods.



(a) Semantic Zoom Overview

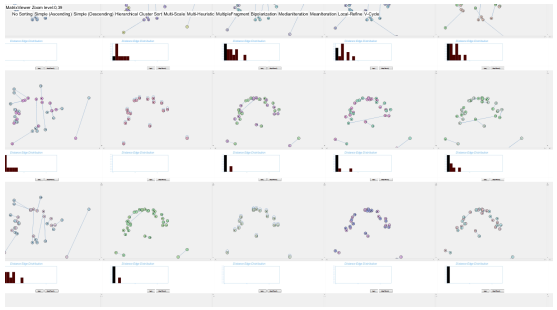


(b) Semantic Zoom Level 1



(c) Semantic Zoom Level 1 (cont.)

(d) Semantic Zoom Level 2



(e) Semantic Zoom Level 3

Figure 3.7 Semantic zoom interface for a comparative matrix analysis task in the soccer analysis scenario: (a) Shows a meta-distance matrix with all pairwise combinations of matrix comparisons for the soccer analysis use case (c.f. Section 5.8.2). (b/c) Shows a successive zooming into the meta-distance matrix. (d) represents the first change in the semantic zoom level. Here a (inner/outer) glyph (c.f. Section 3.4.1) shows the impact of the dimensionality difference on the calculation and the actual distance value. (e) shows the visual representation of the distance score calculation (c.f. Section 3.6.1)

3.5.2 | Semantic Zoom Metaphors to Support Navigation in Large Matrix Spaces

Semantic zoom interaction can help navigate matrices, which do not fit into the available screen space. In [Elm+08; AH04] zooming and dynamic aggregation techniques support the navigation process in large matrices.

We are applying the semantic zoom technique in various contexts and for varying tasks, as Figure 3.7, and Figure 3.8 depict. Our goal is to support visual comparative analysis tasks, as well as visual correlation analysis with this visualization technique. Our core intuition is to give the user details wherever needed, without introducing visual breaks in the exploration process. Thus, semantic zoom interaction helps to keep track of the current analysis focus, while giving incrementally more information to the user. In other words, depending on the granularity of analysis we are adding more and more information and possibilities to interact with the system.

As Figure 3.7 depicts the overview matrix in Figure 3.7a appears visually cluttered. However, a large rectangle area in the middle of the overview stands out representing an abrupt and interesting change in this soccer game analysis use case. In order to find the reason for this break, we are zooming into the meta matrix (see: Figure 3.7b and Figure 3.7c). In Figure 3.7d the inner/outer glyph representation helps us to find out the compared matrices have the same dimensionality (white outer border). Figure 3.7e gives then in the next semantic zoom step and shows details that led to the similarity calculation from the overview. In this case, we can find that the soccer players changed from a classical back-four defense tactic to a close man-marking formation, after a foul was committed.

A different semantic zoom/glyph setup is depicted in Figure 3.8. As Figure 3.8a depicts the overview matrix looks visually homogeneous and is only interrupted with reoccurring low-to-high-to-low gradients patterns. If the analyst is interested in the nature and reason of these gradients, representing a significant increase in solar power production, then he/she can zoom into the area of interest to get more details. Figure 3.8d shows statistical values (here sum and average) of the investigated –or zoomed– time period and Figure 3.8e changes the glyph representation to a time series chart in which the start and end points of the investigation period are highlighted, but the entire time series is given for a contextual view.

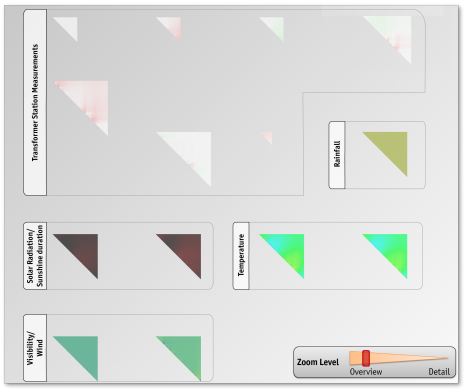
3.6 | Research and Application Context

The case studies in this chapter are centered around several application scenarios in which we show the usefulness of interaction and glyph design for matrix-based representations. As already mentioned in Section 3.3 most of our analysis tasks/scenarios are focused on extracting relationships between one or multiple matrices.

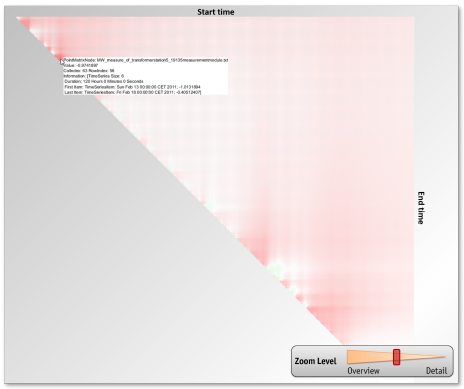
3.6.1 | Visual Comparison of Sets of Heterogeneous Matrices

In recent years many domains have emerged in which the comparative analysis of *sets of matrices* of potentially varying size is relevant. For example, to monitor computer network traffic a dynamic set of hosts and their peer-to-peer connections on different ports must be analyzed. Another example would be the analysis of social networks where actor influences and memberships may change over time.

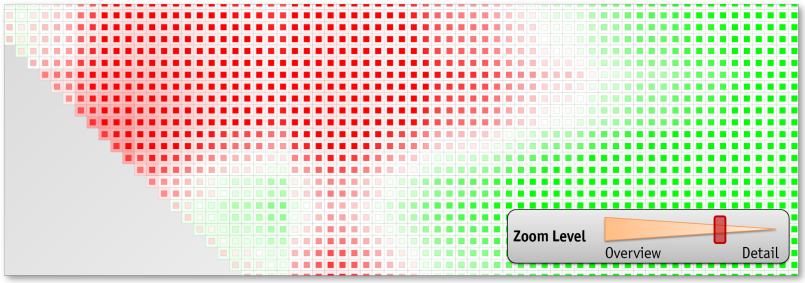
Existing techniques generally support the display of a single, static graph. However, graphs such as computer or social networks may *change* over time. Many analysis tasks are focused on exploring that change; e.g. the comparison of a series of *snapshots* of a network over time. As already detailed in Section 3.3, this gives rise to a matrix *comparison problem*. This problem is particularly challenging since both the edge sets and the node



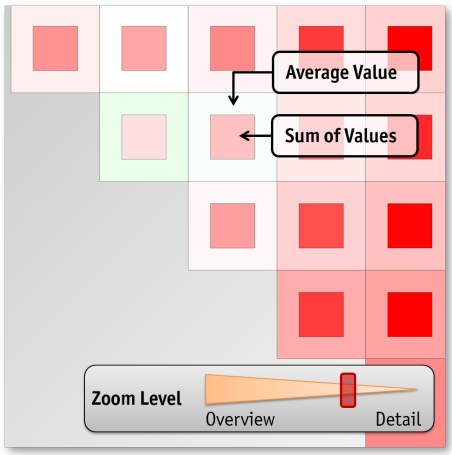
(a) Semantic Zoom Overview



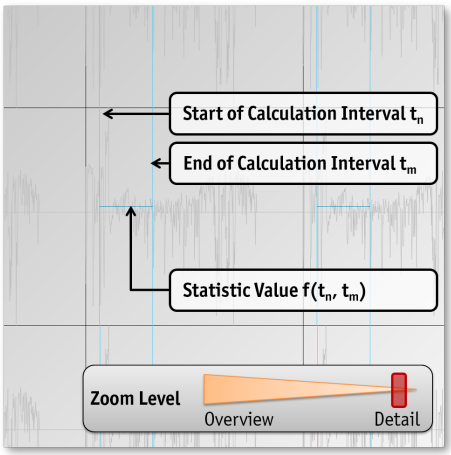
(b) Semantic Zoom Level 1



(c) Semantic Zoom Level 2



(d) Semantic Zoom Level 2 (cont.)



(e) Semantic Zoom Level 3

Figure 3.8 Semantic zoom interface for a visual correlation analysis task: Figure 3.8a shows an overview of all time series matrices for an regenerative energy production use case. Figure 3.8b and Figure 3.8c show the efficiency of substation 56 in the selected time period. Figure 3.8d and Figure 3.8e reveal the impact of the sun's solar radiation and temperature on the substation's efficiency.

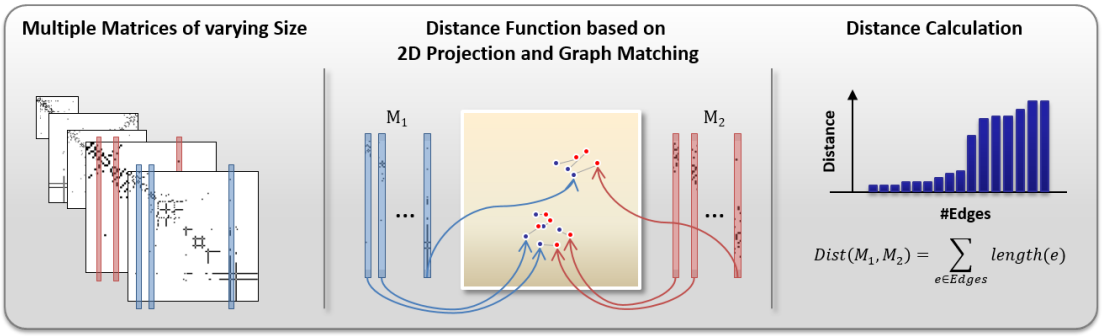


Figure 3.9 Processing pipeline for our projection-based matrix comparison technique: A set of matrices of potentially different size is input (left). Their columns and/or rows are interpreted as high-dimensional vectors and projected to the plane (middle). Solving a bipartite graph-matching problem on the resultant point clouds leads to a set of allocation edges. Aggregating the euclidean lengths of the edges results in a similarity score for each pair of matrices (right).

sets may change, yielding graphs, and thus matrices, of different sizes. Comparison tasks in large datasets can be supported effectively through ranking or clustering operations. These typically require the definition of a *distance function* for sets of matrices. Here too, the variation in size is problematic; the Euclidean distance, for example, can only be applied to matrices of the same size.

In this Section we will predominantly present the interactive system, which allows the users to explore and perceive the matrix dissimilarities and understand potential differences in a set of matrices. A flexible semantic zoom mechanism enables users to navigate through sets of matrices and identify patterns at different levels of detail. Furthermore, we will show a visual representation and interpretation of the distance calculation. Our projection-based distance calculation for heterogeneous matrices will be described in detail in Section 4.5.1, where we also provide a technical evaluation to illustrate strengths and weaknesses.

Visually Interpretable Matrix Distance Comparison

Figure 3.9 illustrates our matrix comparison approach which is used to build a meta distance matrix of distances and incorporated our semantic zoom framework. We regard a matrix as a set $\mathbb{M}$ of high-dimensional row or column vectors. Two sets $\mathbb{M}_1, \mathbb{M}_2$ of vectors can be compared by computing an aggregate vector-based similarity score. If the matrices have the same size then we can easily compare them, e.g. using the Euclidean distance. However, for matrices of different size this is not possible. In the following, we introduce a projection-based distance calculation approach to overcome these challenges.

Our projection-based distance calculation considers the rows and/or columns of a matrix as the basic elements of the analysis. We project these vectors for pairs of matrices

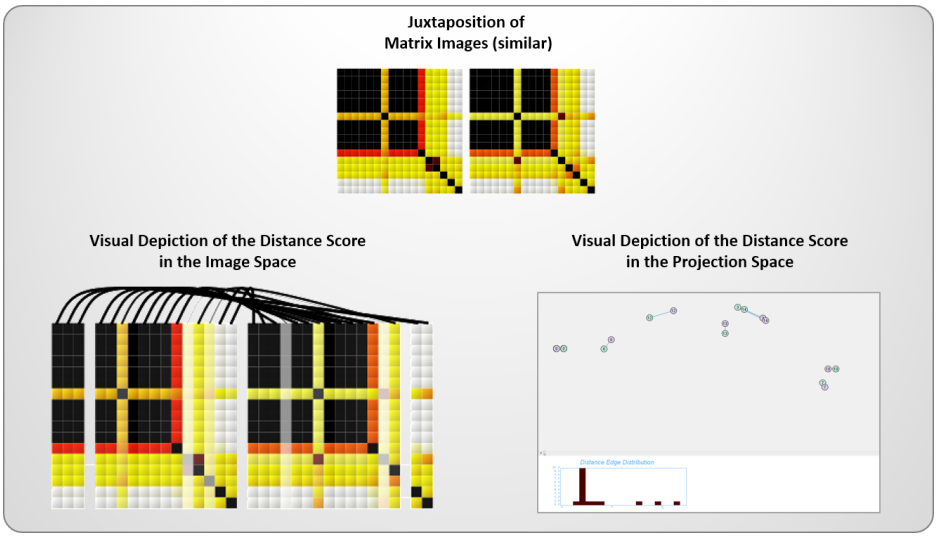


Figure 3.10 A visual interpretation of the projection-based distance calculation: Clicking on a cell of the distance meta-matrix (see Figure 3.7a) shows the compared similar matrices (upper part). A transparency factor for columns indicates their impact on the overall distance score. The matrices' columns are visually connected by edges to represent the bipartite graph matching decisions (lower left). These connections are also shown in the projection view (lower right), which lets the user explore patterns in the projection of columns, i.e., projection points which are close together represent similar columns.

into a low-dimensional space which is used as the reference to compare matrices and identify relationships among them. For each pair of matrices we obtain a pair of point clouds, which can be compared by solving a bipartite graph-matching problem. This bipartite graph matching on the projected elements allows us to compute a measure of distance. Since the matrices are projected with a topologically preserving method we are able to emphasize the essential information of the matrices: i.e. the structural characteristics of a matrix. A key advantage of approach is that it can be interpreted and manipulated interactively as a *visual distance function*, and serves as a comprehensible basis for ranking, clustering and comparison in sets of matrices.

A set of matrices $\mathbb{M}_1, \ldots, \mathbb{M}_n$ which vary in size is shown on the left in Figure 3.9. It is possible to interpret either the rows or the columns as high-dimensional vectors of the length of the respective matrix. Assuming a topology-preserving projection, we project the matrix vectors to the plane (see the center of Figure 3.9). To compare two matrices, we match the projected points of one matrix $\mathbb{M}_1$ to the projected points of the another matrix $\mathbb{M}_2$. Having obtained a matching of the projected point sets, we compute an aggregate distance score over the matching, as shown in Figure 3.9 (right).

Figure 3.10 and Figure 3.11 show examples of the visual distance score depiction. Integrated into the interactive semantic zoom interface (c.f. Figure 3.7a/Figure 5.6; overview-first semantic zoom system) the user is able to inspect the automatic distance calculations.

The visual depiction of a *low distance score* is shown in Figure 3.10. Here all matrix columns could be matched with a low score. This is visually highlighted by the strength of the edges connecting the two comparison columns (visual double encoded by stroke width and -color). Additionally, a transparency factor on the columns intuitively stresses the most similar columns. In the conceptual projection space the circumstance is mirrored by short edges connecting the 2D points (representing the comparison columns). Figure 3.11 shows a counter example with a *high distance score* for the soccer analysis use case presented in [Beh+14b]. Here two matrices are chosen which are not as close in time as the comparison matrices from Figure 3.10. Intuitively, this corresponds to game situations that are less similar than subsequent player positions. This is shown by the visual depiction of the distance score in the conceptual projection space, which allows perceiving many long edges. We are showing ways of steering the distance calculation process to include domain specific modifications (e.g. excluding the goal keeper's position or focusing only on the defensive formation in the soccer analysis scenario) in Chapter 5.

It is also possible to interpret the projected vectors (point clouds) as a complementary form of matrix visualization. Thus, our approach can easily be embedded in visual analytics tasks. In this way, it is possible to visually identify outliers in the two-dimensional space that correspond to outliers in the high-dimensional space. Furthermore, we can use linked interactions between the projected representation and the traditional matrix visualizations to help users interpret and interact with the matrix. Since the approach to interactively steer the distance calculation is inherently incorporated in the visual analytics mantra, we will describe the details of this approach in Chapter 5.

3.6.2 | Visual Correlation Analysis for Time-Dependent Data

In recent years, the quantity of time series data generated in a wide variety of domains has grown consistently. Thus, it is difficult for analysts to process and understand this overwhelming amount of data. In the specific case of time series data another problem arises: time-dependent measurements can be highly interrelated. This problem becomes even more challenging when a set of parameters influences the progression of a time series.

A common analysis task is therefore the comparison of measurements over time, with the aim of discovering meaningful correlations between measurements. An example is the correlation of temperature and power consumption measurements in an energy supply/consumption context. Many existing visual analytics techniques are restricted to the comparison of short time periods (e.g. weeks or days) and do not scale well for longer time series. We present in [Beh+12a] a technique for the visual correlation analysis of numerous, potentially interrelated time series datasets. Our technique centers on a small-multiples representation of several matrices and allows for the simultaneous comparison of multiple, overlapping time series of varying length. In the following we illustrate the

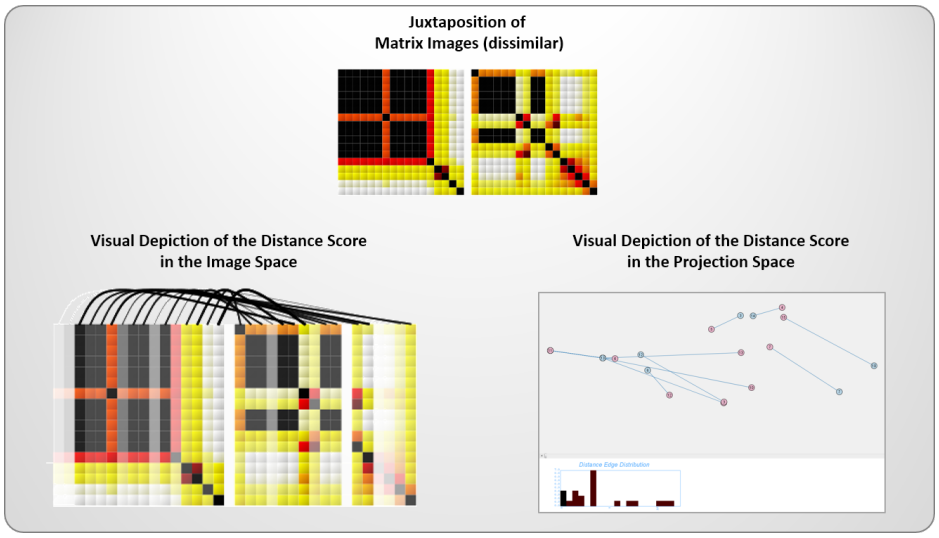


Figure 3.11 A visual interpretation of the projection-based distance calculation: Clicking on a cell of the distance meta-matrix (see Figure 3.7a) shows the compared dissimilar matrices (upper part). The transparency factor for columns indicates dissimiliar matrix columns; only the similar columns are sticking our visually. The long connections in the projection space (bottom right), let the users perceive that the structural differences are significantly higher than in the example Figure 3.10

principle of our technique by applying it to a real-world data set, including measurements of weather conditions and performance parameters of nodes in a power grid .

Power Grid Dataset and Analysis Problem

In collaboration with a German national energy provider, we considered two real world data sets obtained from the town of Freiamt, Germany [Ene]. Freiamt is home to a large number of regenerative power sources, including 160 small, roof-mounted photovoltaic, 3 biomass, and 3 small hydroelectric power plants. Up to 11 photovoltaic plants are connected to one *substation*, acting as a gateway to the power grid. In total, 29 substations and a weather observation station in Freiamt provide the measurement readings. The substation measurements are aggregations of the power generation and -consumption of multiple households and regenerative power plants. The weather station delivers a large number of weather parameters, e.g. rain fall rate, sunshine duration, temperature- and wind measurements, and visibility ranges (fog). In total, we consider nine weather parameters. The measurements span a duration from *2010-12-15* to *2011-12-17*, and are taken at intervals ranging between 10 and 30 minutes.

The goal of the analysis is to understand the interaction (correlation) between the state of the power grid and weather conditions. From a large number of potentially dependent measurement parameters, we want to find the parameter subset which is most useful

for the analysis. While this is a problem in itself, it becomes even more challenging due to the fact that the correlations are inherently local with respect to scale and the time interval. These factors need to be considered in the visual analysis. Accordingly, one can find time-correlating predicate conditions, such as high temperature and long sunshine duration, that could lead to a drop or rise in the efficiency of the power grid.

The following data analysis questions arise: (1) Which parameters lead to a correlation between the reference and other time series datasets? (2) Can we show the parameter's impact on a reference dataset? (3) Which large-scale trends can be determined in long time series (e.g. in one year with more than $365 \times 24 \times 4$ sampling points)?

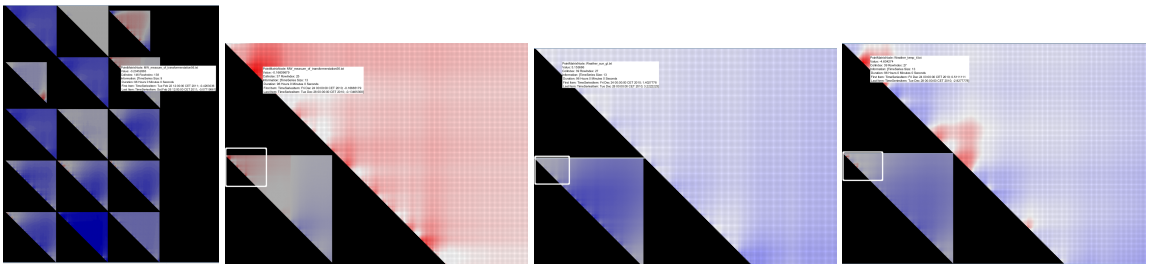
Time Series Matrix Visualization

Our approach centers on a matrix visualization, as it can represent large numbers of time series in a pixel-oriented way, mapping each value to the color of a pixel. In this triangular matrix representation, the horizontal and vertical axes describe the start and endpoints of a specific time interval in the overall time series. Accordingly, each matrix point $x_{(i,j)}, i > j$ refers to a time series interval starting at time $t(i)$ and ending at time $t(j)$. To foster correlation analysis in the matrix we show statistics computed over the respective time series intervals, thus providing a tool for the *screening* of correlations at different intervals and offsets. The statistic values are presented in a mouseover tooltip.

The color of each data point represents a statistical measure $f_{(i,j)}$ computed over the interval $[t(i), t(j)]$. The measure, f, can be set by the user on-demand. Example measures include the trend (slope of the regression line), the standard deviation, average, or the geometric mean. Minimum, maximum, variance, sum and squared sum statistics can also be computed. The time series can be resampled on demand to set hours, days, months, etc. as the base unit of measurement.

The color map is an important design factor for the comparability of the matrices. Since, the transformer stations can have a positive or negative net output to the energy grid, depending on whether power is consumed or produced, we chose a bipolar red to blue color map. As Figure 3.12 (b) depicts, negative net outputs (power production states) are visually outstanding, due to their red color. Other measurement parameters are shown by additional triangular matrices in a small-multiples display. They represent the available weather information (e.g. air temperature 5 cm above ground level in °C, sunshine duration and rainfall rate) over the available time period. A local (per matrix) color map can be applied on demand to reveal the feature's special characteristics. A semantic zoom interface lets the user explore the correlations between the matrices in an overview (c.f. Figure 3.8a) or inspect the data characteristics in an emphasized time span (c.f. Figure 3.8c). For the lowest semantic zoom level a line chart representation of the corresponding time span is used, as depicted in Figure 3.8e.

Application to Power Grid Data Analysis



(a) Matrix Overview **(b)** MW/hour of one Transformer Substation **(c)** Sun Solar Radiation: **(d)** Temperature in °C

Figure 3.12 (a) shows an overview of all time series matrices. (b) shows the efficiency of a specific substation in the selected time period. (c) and (d) reveal the impact of the sun's solar radiation and temperature on the substation's efficiency.

Figure 3.12 showcases one of the findings we made using our approach on the Freiamt dataset. Figure 3.12a shows an overview over six transformer substations and nine weather parameters. Figure 3.12b represents the positive and negative megawatt consumption rate of substation 56 in a larger view. This substation is especially interesting, since eleven photovoltaic power plants are connected to it. The visual task is to find power injection phases represented by a dark red color. Two visually outstanding areas exist in the one year time period. The first ranges from *2010-12-24* to *2010-12-26* and the second from *2011-02-22* to *2011-02-24*. Here the substation fed on average 0.167 and 0.122 megawatts per hour respectively into the power grid.

The weather factors corresponding to this effective power production can be seen in Figure 3.12c. Here, the temperatures were on average −4.60 and −2.11°C respectively. Figure 3.12 reveals that the global solar radiation, measured in Joules/cm^2, averaged 5.154 in December and 4.483 in February. This leads to the hypothesis that photovoltaic power plants work most efficiently in temperature ranges between −5°C and −1°C and lose efficiency in temperature ranges above and below, even if the sun duration and solar radiation is high.

3.6.3 | Visual Comparison of News Text Clusters

Websites of newspapers, magazines, radio and television broadcasters publish stories, which are often provided by major news agencies, such as Associated Press, Reuters, AFP, etc. These news stories and feature articles can be prepared by agencies in a way that requires little modification, but very often the clients edit the text before delivering it to the reader. Alternative information flow is created by independent and local media, who publish news stories that are later picked by other media providers and redistributed through their channels.

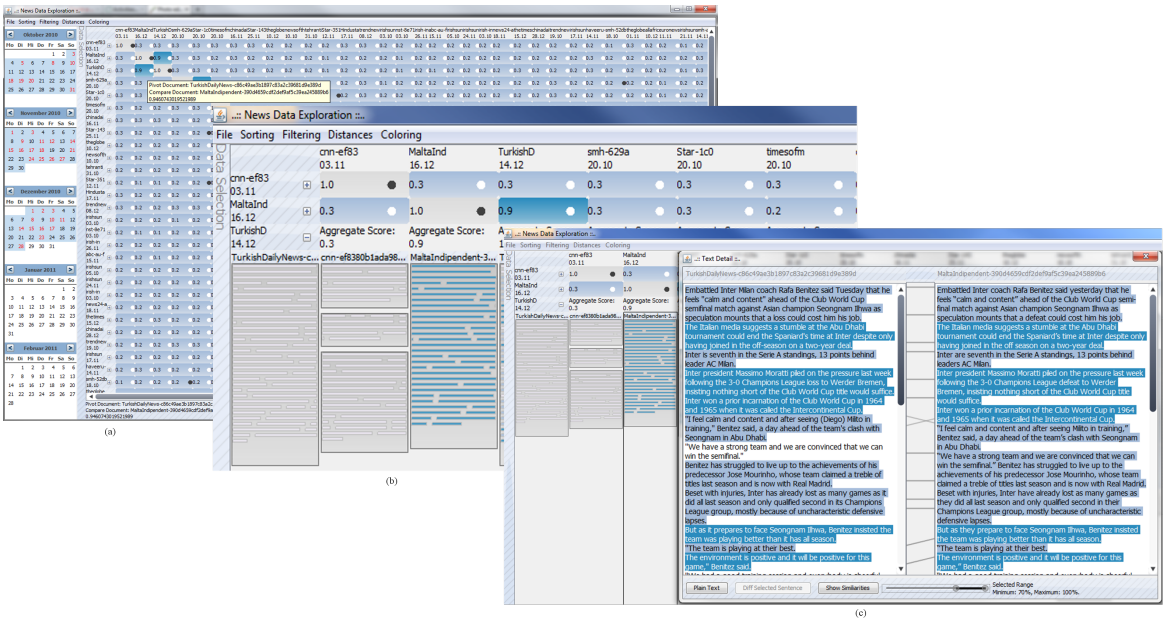


Figure 3.13 News Auditor: The user explores a news story cluster by identifying interesting patterns in the similarity matrix Overview (a); The Structural View (b) provides a visual comparison of differences between selected articles on the paragraph level; The Document View (c) shows direct changes between two articles on the word and sentence level.

News aggregators, such as Google News, Yahoo News, or Europe Media Monitor provide the end users with the latest and the most important story clusters, where news articles are grouped by their similarity. The aggregators present the user a simple list, where the first (and the most important) article is usually presented with a title, short summary and a photo, while the other articles are represented as hyperlinked headlines. Navigating through a story cluster becomes a daunting task, since it gets very hard to understand the differences across different sources and find new information without reading every article. A similar problem exists when important breaking events happen, when immediate response by news providers is required. These developing stories are continuously updated as soon as the new information becomes available and the reader needs a fast and effective solution to differentiate the new from the old.

In the following, we present a visual analytics tool, called *News Auditor* , that helps the reader in the exploration of a news story cluster. Our work presents a proof of concept that identifies *what* is different in similar news items, combining existing automated methods for measuring text similarity and interactive document visualization. The architecture of the tool allows easy integration of more sophisticated natural language processing methods, which would help the reader in understanding *how* the content is different.

Problem Description

The goal of our work is to help the user to understand the content of a large document corpus, while understanding the main themes and the various differences among individual news articles. In a real-world scenario, news clusters can contain hundreds of related news articles, but only rarely more than 100 documents per cluster can be retrieved without taking topic shifts or -drifts into account. In this document corpus user should be able to:

- identify interesting articles in the story cluster
- understand *what* are the differences between news articles
- understand *who* changed the content.

Given a cluster of news stories, we assume that the documents are related and can appear in one of two scenarios: a) the documents are news reports from multiple sources on the same event; b) the documents are updates from a single news source on an ongoing event. In order to help the user to get a better insight from a cluster of news articles, we need to combine automated methods for efficient computation of document similarity and visualization techniques that would show the changes at different levels of detail.

Text Cluster Analysis in the *News Auditor* System

We design our system following the overview and detail concept [Shn96], to allow the user exploration of a document collection on different abstraction levels. On the overview level, inter-document distance scores guide the user to interesting patterns within the text collection. A more detailed comparison on the structural level shows the differences between the documents on the paragraph level. Lastly, a document level view shows two articles side-by-side, to provide direct comparison of the texts. Due to this structured approach, it becomes possible to lead the user to non-obvious patterns in a topic-coherent news cluster.

The overview visualization, depicted in Figure 3.13 (a), represents a heatmap color-coded *similarity matrix* and functions as an inter-document comparison view. In the matrix, each cell represents the similarity between the *pivot document* (row ID), and the *comparison document* (header ID). Due to this compact matrix-based visualization approach we can investigate document corpora with hundreds of documents. In the case of a pixel-based representation of one document comparison, the total number of inter-document comparisons is only limited by the end-user's screen space. The cell's numeric value depicts the user-chosen textual similarity measure and is color-coded with the selected or default option. A logarithmic color-to-distance mapping is implemented to emphasize important distance intervals. To guide the user, each cell contains a small black or white glyph that depicts whether the articles stem from the same news source (black dot) or not (white dot). A binning-based or continuous heatmap color coding is used in all aggregation views. The binning-based color codings differ in the number of bins and the base colors. In Figure 3.13, a light-to-dark-blue color coding with three classes, extracted

from [Bre12], is shown. Furthermore, users can decide to filter out news updates from the similarity matrix.

The matrix view is enhanced by three information filtering and interaction subcomponents, which help in finding patterns of interest. On the left side, a calendar component is used to filter time intervals. The user can control sorting, filtering and coloring settings and choose from three available distance measures, such as Cosine, Google NGD [CV07], or a semantics-driven bag-of-synsets distance. The matrix can be reordered by highest/lowest similarity or according to different usage-driven scenarios that can rely on the article metadata, such as finding copied or reused texts from different news providers, by time of publication, etc. Additionally, the articles can be grouped visually by the news source, showing the update processes happening during the news evolution. To give more information to the user, a context-dependent information status bar on the bottom shows the article id, numerical scores or other meta-data.

After getting an overview of the news cluster, the user can choose to expand one or more rows to explore the structural features of the documents and their differences. As an example, the rows with very high overall similarity scores in a few or all documents can be regarded as suspects for plagiarism. Structural View is shown in Figure 3.13 (b). Here, *document thumbnail* glyphs, as described in Section 3.4.3 visually encode the sentence- and paragraph structure, as well as their textual similarity in comparison to the row's pivot document.

For in-depth investigation, the users can switch from the structural view to the textual representation. This component is shown in Figure 3.13 (c). The text detail view shows the pivot and comparison text in the left and right text panel, respectively. Besides comparing the text by reading both articles, the user is supported by the color coding. The range slider on the bottom of the screen helps the user to highlight sentences within user-selected similarity intervals. Thus, it is possible to filter out all sentences above 80%, see the minimum or the maximum similarity boundaries. Highly similar sentences (above 70% similarity) are visually connected by reference-lines that appear in the space between the documents. By clicking on a sentence, the most similar sentence in the other document is highlighted, showing the word-based similarities with the help of the *Diff* algorithm [HM76], which visually marks insertions/deletions.

Case Study: Reuse of Text by different News Agencies

One primary question, which can be answered with *News Auditor*, refers to the reuse and copying of text. In Figure 3.13 (a), one can see an example for the copying of news from an earlier news source. These are Champions League soccer news articles, which appeared in the period from October, 1^{st} of 2010 until December, 31^{st} of 2010.

The overview is configured with the distance-aggregate sorting option, filtered updates, and the cosine similarity as a text similarity measure. With *News Auditor* most

uninteresting documents can be discarded immediately in the Overview matrix due to its low inter-document similarity score (rendered in light blue). Every document with a high similarity score, depicted by a dark blue color, and a later publishing date appears interesting. These characteristics occur, e.g., at the second column and third row. The copying hypothesis is even more obvious if it is not the same source that published the article. In Figure 3.13, the initial article was published on October, 14^{th} of 2010 from the *Turkish Daily News* agency and modified on October, 16^{th} of 2010 from the *Malta Independent Press*.

The structural comparison in Figure 3.13 (b) shows that most sentences are in high similarity classes. The structure appears to be stable, yet the length has changed marginally. In fact, the textual investigation, shown in Figure 3.13 (c), reveals that 21 of 31 sentences are in the similarity interval of 90% to 100% with insignificant changes, such as inserting/deletion of hyphens, quotation marks, or punctuations. Eight sentences have minor modifications, such as plural/singular changes, with a similarity score between 80% and 89%, two sentences are in the 70% to 79% range with word (-suffix, -prefix) exchanges or additions, and only one sentence is in remaining range of 0% to 69%, which has been deleted in the latter news text.

Case Study: Updating of News from the News Producer

Figure 3.14 depicts a different use case. Here, the task is to find updates, which stem from the same news source, and compare them with regards to their content. Thus, the similarity matrix is sorted according to the same-sources-first option, without filtering updates, and the Cosine similarity measure. For this specific task, a user needs to find cells that are labeled with a black dot (depicting the same source) and a high inter-document comparison score.
As Figure 3.14(a) shows, a news article by *CNN* can be found in a news cluster that deals with the Wikileaks founder *Julian Assange*. It has been published and modified on Dec., 7^{th} 2010.

Figure 3.14 depicts in (b) that various modifications have been made to the news article, both in the structure and the text. Despite the case that the majority of sentence are the same, it can be seen that sentences like, e.g., *"English socialite Jemima Khan had offered to pay bail of 20,000 pounds ($31,500) and journalist John Pilger also offered a sum of money."* have been deleted. Figure 3.14 (c) shows one of the minor textual modifications. Here, *"[...] he wrote a location [...]"* has been modified to *"[...] he then wrote it [...]"*. Marginal changes, such as exchanging currencies, insertions/deletions of abbreviations, etc. can be found throughout the news samples and lead to the hypothesis that either a full sentence text is copied or none of it.

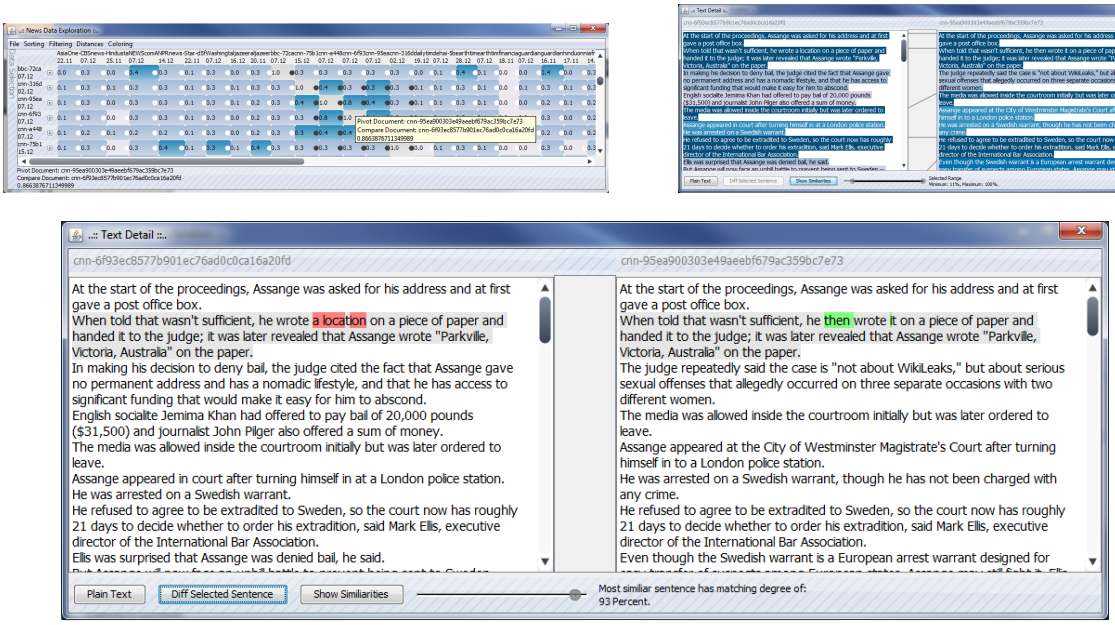


Figure 3.14 Exploration of the same-source content differences. The sorting in the Overview (a) reveals several similar articles published by CNN. The sentence- and word-level differences are shown in (b) and (c), respectively.

3.6.4 | Visual Comparison of Matrix Reorderings and Retrieval Rankings

Retrieval systems are omnipresent and indispensable components for information-centered work. However, different retrieval systems may provide deviating rankings, the joint consideration of which may be important. Further, in analysis domains such as bioinformatics or network security, decision making processes are based on sets of rankings. A central problem is that users are often not able to assess the quality and/or stability of a given ranking, since alternative rankings are often neither presented nor computed as a reference. This problem is inherently prominent whenever rankings are influenced by user-/system-determined parameter settings, such as the used similarity function, feature vector representation, or the underlying retrieval algorithm. A similar problem arises in the case of matrix reordering: Since a matrix can be reordered with many different algorithms an obvious question is “Where do matrix reorderings differ or overlap?” This information can be used for interactive or user-guided matrix reorderings, such as presented in Section 5.4.

We are considering the problem of comparing large sets of rankings. We devise a solution to this problem inspired by Shneiderman’s Visual Information Seeking Mantra [Shn96]. Specifically, we define three comparison levels of interest and corresponding visualization support as follows:

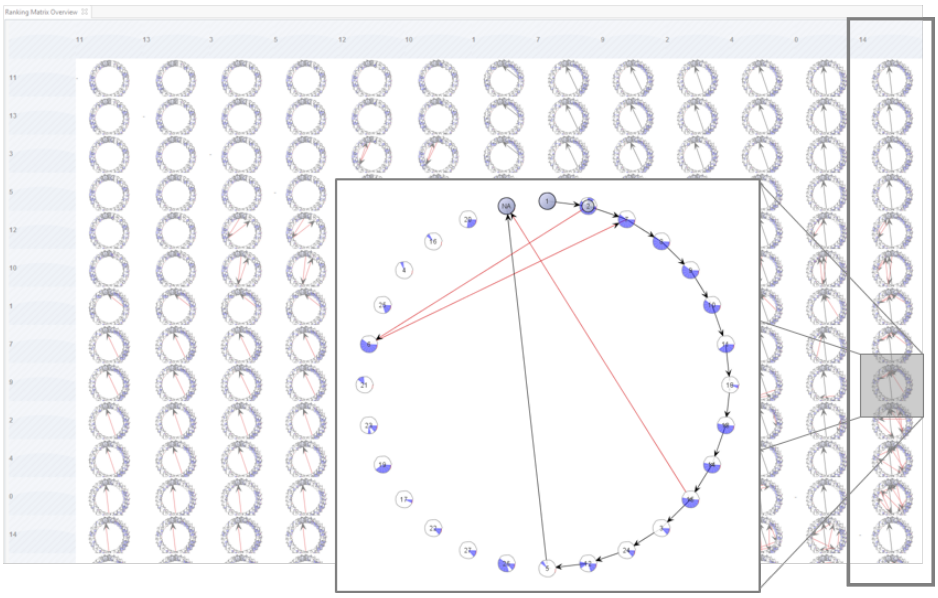


Figure 3.15 Visual comparison of gene sequence data in a biological data use case (1:N comparision).

1. The first comparison level refers to *overviewing of rankings*. In our case, this corresponds to all possible combinations of rankings. In this N:N comparison task, a goal is to identify consistent from contradictory results by visual means. A matrix representation is a straightforward tool. Correlating structures among the matrix cells can be identified, similar like in the Scatter Plot Matrix approach [Cha+83] for high-dimensional data.
2. A row-wise or column-wise analysis can take place in the comparison matrix, corresponding to a *more detailed* comparison level (1:N comparison task). The goal is to comprehend, which comparison ranking has the most consensus (or disagreement) with respect to the default ranking.
3. When the user is able to identify one interesting comparison view the task changes to a *detail-on-demand* view. Specifically, 1:1 comparison views among rankings can be selected by users

We will next discuss three different use cases to illustrate the applicability of our ranking comparison approach.

Application to Biological Gene Sequence Data

In bio-molecular research, the BLAST [Alt+97] algorithm is a well-known approach to perform a similarity search against a database of genes with a given query sequence. The result is a list of genes ordered by similarity and limited by a significance threshold. A high similarity of a retrieved gene to a query suggests that the gene and the query have a

common ancestor, and probably serve the same biologic function. BLAST requires a set of parameters to be provided. Typically, it is used with default settings, yet it is known to be sensitive to parameter changes. It is therefore of interest to compare result lists from different runs with varying parameter settings to the default settings. Result lists from different parameter settings can either include more or less genes, furthermore the order can be different.

Figure 3.15 shows the comparison of the result ranking for the default settings compared (used as base) against rankings obtained by twelve alternative parameter settings. It is clearly recognizable that nearly half of the genes are not found with the default parameter settings and that the order of the found genes differs between parameter settings. However, some settings result in no positional changes and others show similar changes among each other, which can be explained by only small parameter variations between settings. By means of the pie-chart node representation, the single ranking view also allows to assess that genes which have not been found with the default settings are only found with a small number of parameter settings. Moreover, the positional accuracy encoding of node "1" depicts that this gene was always the most similar gene settings. Therefore, this result item can be considered the most stable.

Application to Image Retrieval Ranks

The search for similar images is a prominent task in multimedia retrieval. It typically relies on image descriptors and according similarity functions, of which many different alternatives exist. We consider an example of comparing rankings of TreeMap views. Alternative rankings are given by using different descriptors and similarity functions. We consider a set of standard descriptors (including Global and Local Color Histogram, Local Edge Histogram, and Hough Transformation descriptor) and similarity functions (Euclidean, Cosine Distance, Dice coefficient). We consider a set of 100 artificially created TreeMap views ranked against a given query view and using different combinations of descriptors and similarity functions. Our question is, which combinations result in similar rankings.

Figure 3.16 shows illustrative results for the comparison of rankings along different descriptors and similarity functions. One can visually depict that despite the very different definition of the rankings, some combinations provide similar rankings. E.g., in the intra-descriptor analysis (varying the similarity function, but not the descriptor) the Edge Histogram descriptor results in a similar ranking result when comparing Cosine- and Euclidean distance. In an inter-descriptor comparison (varying the feature descriptor, but not the similarity function) it becomes obvious that the Global Color Histogram descriptor delivers significantly diverging ranking results as the Hough Transformation.

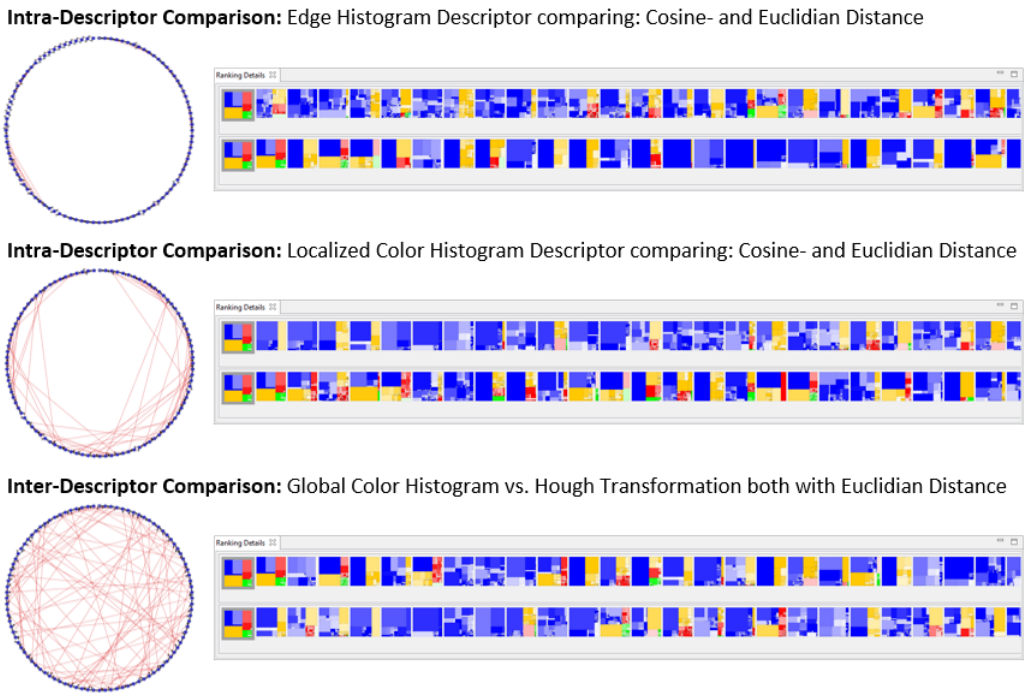


Figure 3.16 Visual comparison of image retrieval results obtained using different image descriptors and similarity functions (1:1 comparision).

Application to Matrix Reordering Comparisons

Finally, we consider a more abstract use case. Effective matrix visualization typically requires appropriate matrix sorting. To date, many matrix sorting algorithms have been proposed, and we can apply our tool also to compare such sorting algorithms. We use matrix data from the Jordi Petit test suite [Pet03], and a set of eight matrix sorting algorithms for illustrative purposes. In the N:N comparison of the matrix sortings depicted in Figure 3.17 we can assess the matrices' sorting conformity and conduct a visual pattern search for correlations. It stands out that a larger amount of edge crossings identifies the *Multi-Fragment* sorting algorithm as the most disagreed ranking result (also depicted by the matrix image on the bottom of the column). In a more detailed view, the *gray areas* become of interest: Here some of the algorithms disagree on a part of the ranking list. However, this is contrasted by the *green areas*, representing algorithms with a large consensus among the ranking results.

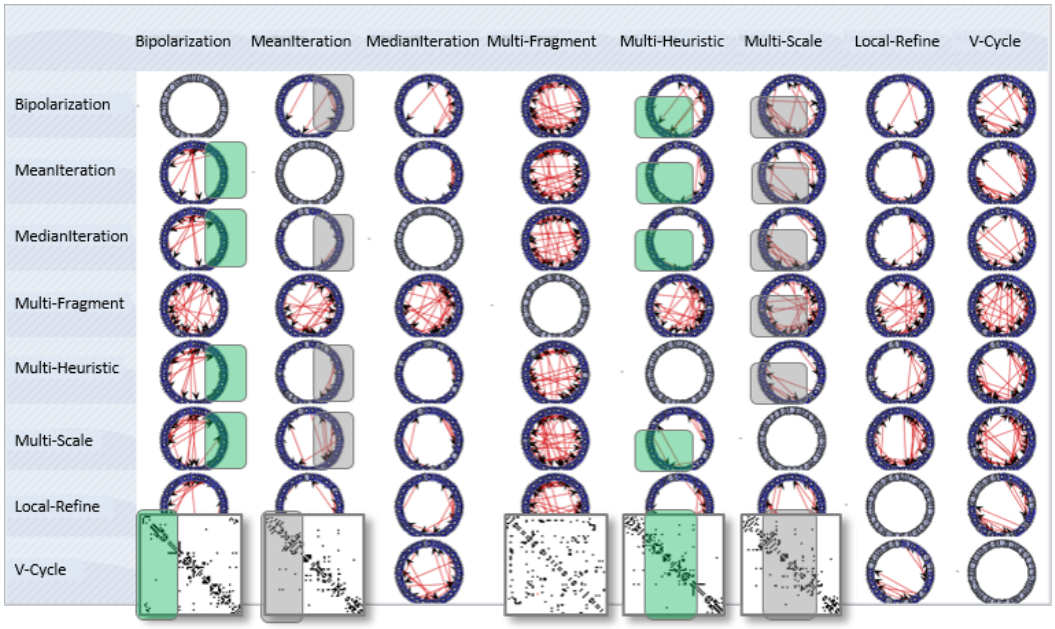


Figure 3.17 Visual comparison of 2D sortings in a matrix sorting use case (N:N comparison)

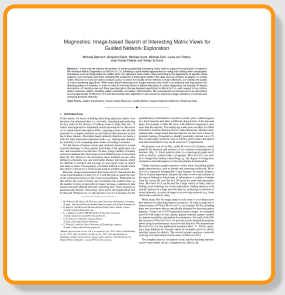
4 Automatic Support for Pattern Retrieval in Matrix-based Representations

Contents

This chapter of the thesis collects all contributions focusing the automatic analysis of matrix patterns. We will motivate our work in Section 4.1 and highlight related work.
In Section 4.4 and Section 4.6 we will present several novel, established, adapted feature extraction approaches for a pattern retrieval in matrix-based representations.
In Section 4.7 we will introduce a quantitative evaluation scheme to assess the effectiveness and efficiency of the proposed image-space pattern descriptors for retrieval and analysis tasks.
In the last Section 4.8 we will demonstrate the applicability and usefulness of our approaches in several application contexts and show how data mining can help to leverage retrieval and exploration processes.

The core contribution of this chapter lies in the quantitative performance evaluation of feature extraction methods for the suitability to detect matrix patterns.

This chapter is based on the following publications:



"Magnostics: Image-based Search of Interesting Matrix Views for Guided Network Exploration"

Behrisch, Bach, Hund, Delz, Rüden, Fekete, and Scheck.
Under Revision and conditionally accepted at Visual Analytics Science and Technology (VAST), 2016 IEEE Conference on, IEEE CS Press, 2016, 43-52. [Beh+16a]

"Quality Metrics Driven Approach to Visualize Multidimensional Data in Scatterplot Matrix"

Behrisch, Shao, Buchmüller, and Schreck.
Eurographics Conference on Visualization (Poster Paper), 2015. [Beh+15]

Parts of the Motivation Section 4.1 and the Overview Section Section 4.3 are adapted and/or taken from the text/figures I have written/developed for the German Research Foundation (DFG) research proposal "Transregional Collaborative Research Center 161 Quantitative Methods for Visual Computing."

4.1 | Motivation

Searching and analyzing are key tasks for making use of large data sets. For structured –such as relational– data much progress has been made to date in database query processing. Indispensable to search and analysis in structured data are appropriate and meaningful feature descriptors on which a distance function can measure the degree of similarity (or distance) between data objects. Similarity functions are the basis for any ranking method (for search tasks), or for clustering data by similarity (for analysis tasks).

For relational and high-dimensional data, typically, custom feature extraction methods are required to implement similarity computation as a basis for search and analysis purposes. To date, an abundance of feature extraction methods have been proposed for different types of structured data. However, they are often defined in a heuristic way, and yield rather abstract representations, which are difficult to understand and leverage by non-expert users in search and analysis applications. Consequently, it remains difficult to decide which descriptor is best suited to solve a retrieval and analysis problem at hand. In other words, if a content-based analysis focuses on pattern-related retrieval tasks (e.g., Which pattern occurs more often than another? How much noise can be discarded before starting a detailed analysis?) then patterns must be modeled sufficiently by its feature descriptors. One main scientific challenge is that the definition of feature vectors, as well as similarity functions is complex, and often requires knowledge of the user or application context.

Research Objectives: A range of research questions can be derived whose answer forms a basis for pattern-driven exploration approaches:

1. How can we describe the visual characteristics of matrix-based representations, such that visual patterns are in the focus?
2. How can we validate and quantify the performance of these automatic feature detection approaches?
3. How can we develop automatic analysis approaches and visual depictions thereof that help to assess a feature vector's performance wrt. the human's intuition?
4. Which data transformation, i.e. matrix reordering, has an impact on the similarity and relevance computation for retrieval and analysis tasks? Specifically, the dependency between data transformation and feature extraction methods and their impact on the effectiveness of retrieval and analysis applications will be in the focus.

Many (image-based) feature descriptors are available that are specifically tailored to quantify a distinctive aspect of the data. For matrix-based representations, our hypothesis is that some of these feature descriptors can be used to derive an insight about their

"patternness". In other words, we assume that feature descriptors allow us to quantify the *interestingness* of matrices with respect to the visual patterns they contain. For the evaluation of this hypothesis a practical problem arises: in many cases large amounts of matrices are available, but textual annotations about their patterns are lacking or are only partially available. To overcome these problems we present in this chapter several (automatic) evaluation approaches for pattern analysis in matrices.

4.2 | Related Work

Our work relates to interactive and automatic approaches for view selection, relevance-driven information retrieval, and systems which capture user feedback to guide the analysis process.

Quality-Driven Relevance Analysis in Large Data Spaces Visual data analysis methods need to be able to handle increasingly large data sets. However, not only the data size grows, but also the possible visualization space for this data. This problem gets even worse when the amount of view parameters is taken into consideration. In the case of the analysis of an n-dimensional data set with scatter plots, $(n \times (n-1))/2$ two dimensional projections can be produced [Tat+11a]. If the same data set is visualized with a Parallel Coordinates Plot even $n!$ possible column orderings exist [DK10a]. Similar problems arise in visualization of adjacency matrices, which comprise $(n! \times n!)$ valid row/column orderings. However, all of these visualization approaches have in common that only few view configurations lead to relevant or non-redundant information. Hence, intelligent methods for compressing and filtering data for potential patterns of interest are researched.

General approaches to support the identification of relevant views in large view spaces include *clutter reduction* [ED07b; PWR04b] and *dimensionality reduction* [Ing+10b; Tat+12a]. Besides fully-automated approaches, others explore interactivity, empowering the user. For example, in [Beh+14a] an interactive scatter plot exploration approach using a classifier to learn the notion of interestingness from user feedback is proposed. A visual query interface for multivariate data using regressional features is presented in [SBS11]. Alternatively, sketching can be used to express patterns of interest in a large scatter plot space [Sha+14].

Methods Based on View Quality Quantifying the interestingness of visualizations typically requires heuristic feature-based approaches that respond to the (potentially) interesting structural characteristics of a visualization. These methods try to mimic human perception in that they distinguish one or more visual patterns from noise. Several previous works exist, tailored towards specific patterns for certain visualization techniques.

For scatter plots, Wilkinson et al. [WAG05c] introduce *Scagnostics* (scatter plot diagnostics), using graph-theoretic measures to describe point distributions. Their feature vector consists of nine interpretable characteristics which are important in the analysis of scatter plots. By using one of these measures, an analyst can make assumptions about inherent information of the described scatter plot. Scagnostics are global features, describing a whole scatter plot at once. Recently, Shao et al. [Sha+15] proposed usage of local features to rank scatter plots. The approach first applies a density-based segmentation of local scatter plot patterns, and then identifies relevant views by an interest measure defined over local patterns.

Similarly, Dasgupta et al. [DK10a] propose *Pargnostics* for Parallel Coordinate Plots with the goal to optimize the axis layout so that user's preferences are met. Pargnostics introduces several statistical and image-space measures to quantify e.g., the number of line crossings, crossing angles, convergence, or overplotting measures, all being candidates to rank relevant or informative views.

For dense pixel displays, Schneidewind et al. proposed *Pixnostics* [SSK06], a set of statistical measures in pixel-oriented visualizations. The entropy of an image is measured and shows to be useful to distinguish structured views from noisy ones, reducing the interactive search time for pattern retrieval tasks. In line of this work, Albuquerque et al. present the Noise-Dissimilarity measure for Jigsaw Maps [Alb+10], which we also adapted and tested for our matrix pattern analysis scenario. For high-dimensional data analysis, Bertini et al. [BTK11c] proposed a conceptual model for assessing the quality in image spaces and to integrate view quality into the visual exploration process. Finally, Sao et al.'s [SS04] rank-by-feature framework makes use of correlation and entropy measures to find an appropriate order within histograms, boxplots and scatter plot views.

Dimension Reduction and Fuzzy Graph Matching. One of our data-space feature extraction approaches presented in this chapter (see: Section 4.5) uses data projection to compare matrices based on their row/column elements, and defines a distance function based on bipartite graph matching. We therefore pinpoint here to the relevant literature.

Many techniques exist to reduce the dimensionality of data [POM07] and support the exploration process in analysis tasks [Yan+07a]. Matching algorithms, on the other hand, can compare data by finding correspondences of local data properties. An example is the matching of regions in images based on the correspondence of local SIFT features [Low04]. Comparison of graphs by edit distances has been proposed in [ZWS96], however this is an expensive process. Inexact or fuzzy graph matching approaches try to cope with the computational effort by applying tree search/indexing algorithms [SF83; Cor+96; Pel98], transforming the graph matching problem into a continuous, non-linear optimization problem [FE73; WH97; WW02] or exploring spectral characteristics of the graph [Ume88; CK04; KC02]. Similar to our approach, in [KC02] a vector space is defined using the

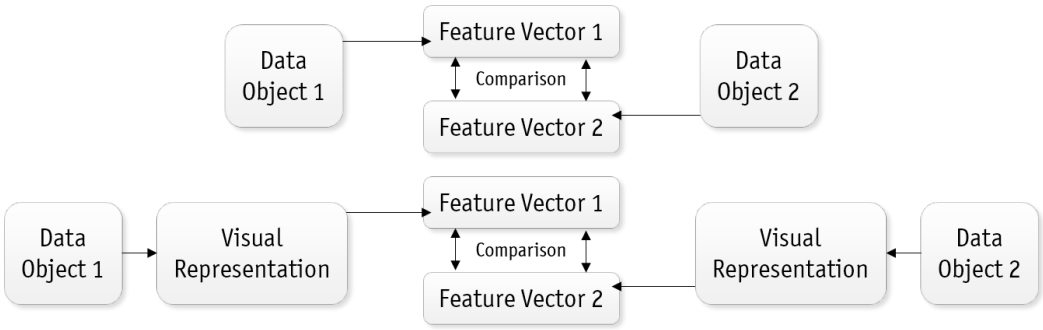


Figure 4.1 Top: The standard approach to feature extraction operates on the raw data and is typically defined in a static and heuristic way. Bottom: Our approach extracts features from a visual representation of the raw data. The approach is able to visually represent why objects are similar, and provides a starting point for user interaction and navigation.

Eigenvectors of the adjacency matrices. The graph nodes are projected onto points in this space. Then, a clustering algorithm is used to retrieve common local relational structures between different graphs [KC02]. The authors state that this method is robust with respect to graph distortions; corresponding nodes are always close to each other in their graph Eigenspace. The comparative analysis of sets of graphs has been considered in [LGS09] by means of clustering of statistical graph features.

Content-Based Image-Based Retrieval Many image retrieval systems so far rely on low-level image features, such as color histograms, edge histograms, or texture measures [DKN08], which are heuristically combined to form distance functions. A problem in content-based search is often how to define a query if no example search object is available. Relevance feedback methods for image retrieval may operate typically on low-level feature representations in various ways. One option is to construct a new query vector by averaging the feature vectors of all image examples marked by the user as relevant. Another option is to train a classifier (e.g., SVM or Decision Tree [HKP11a]) from the set of relevance information provided by the user.

4.3 | Overview

In this chapter we present *automatic feature extraction* techniques that can be used to describe the visual patterns in large matrix spaces. As already noted in Section 2.1 we are striving for automatic detection methods for visual features and patterns, which form the basis for more sophisticated analysis scenarios, such as retrieval, clustering or classification of visual patterns in matrix plots.

From an abstract view point, we are exploring a new class of similarity calculation approaches that are based on visual representations of relational data. The basic idea

is to regard visual representations as a proxy to the data of interest, and base similarity and relevance computation tasks on the visual data representation, instead of the original (raw) data. Our aim in doing so, is to provide user-friendly, interpretable and interactive assessment functions as a basis for search and analysis tasks. Novel visual descriptors try to mimic human perception aspect thus helping the user to improve the interactive query specification and analysis result interpretation stages of the visual data analysis process. In the following we will focus on the definition, usage and interpretation of visual features for relational data.

As discussed in Section 4.1, searching and analyzing are key tasks for retrieving, relating and reusing of complex data sets. Similarity functions are required to implement search and analysis applications and are especially important for relevance/quality assessment tasks. Existing approaches typically compute the similarity between complex data objects based on static feature vector representations which are extracted from the raw data. These representations are typically defined heuristically and applied in a black-box manner. As a consequence, 1) the similarity concept (feature encoding and similarity function) are fixed and cannot adapt to the context of the user task, and 2) it is hidden from the user why two data instances are considered similar or not. These problems reduce the effectiveness of search and analysis applications, as the similarity notion cannot adapt to user context and results may not be easily interpreted by the users. While we are dealing with the extraction and quantification of patterns the following sections, we will show in Section 5.6 several approaches to let the user change and adapt the notion of similarity with respect to the current contexts and questions.

As Figure 4.1 depicts the standard approach to *similarity computation* extracts feature representations from the raw data (see Figure 4.1 (Top)). We propose to extract feature representations from visual transformation of input data (see Figure 4.1 (Bottom)). This approach may provide several advantages. First, a visual transformation of data is naturally linked with the user interface: Visualization of the data is used in many applications, and it can be intuitively shown why two data representations are considered similar – namely, by showing corresponding visual features in the visual data representation. A second main advantage is that one can easily implement visual query interfaces which allow the user to sketch or mark data or patterns that they are interested in, and compute similarity between these visual queries and the target data. We present a query-by-sketch interface for matrix-based representations in Section 5.5. Finally, the similarity notion can flexibly adapt to user needs: different visual abstractions give rise to different similarity notions and allow users to choose the similarity notion simply by selecting one visual transformation from the possibilities. Specifically, this relationship is given in our research context, since distinct matrix reordering algorithms may produce different visual patterns, hence making some algorithms more suitable for a specific task at hand. An appropriate,

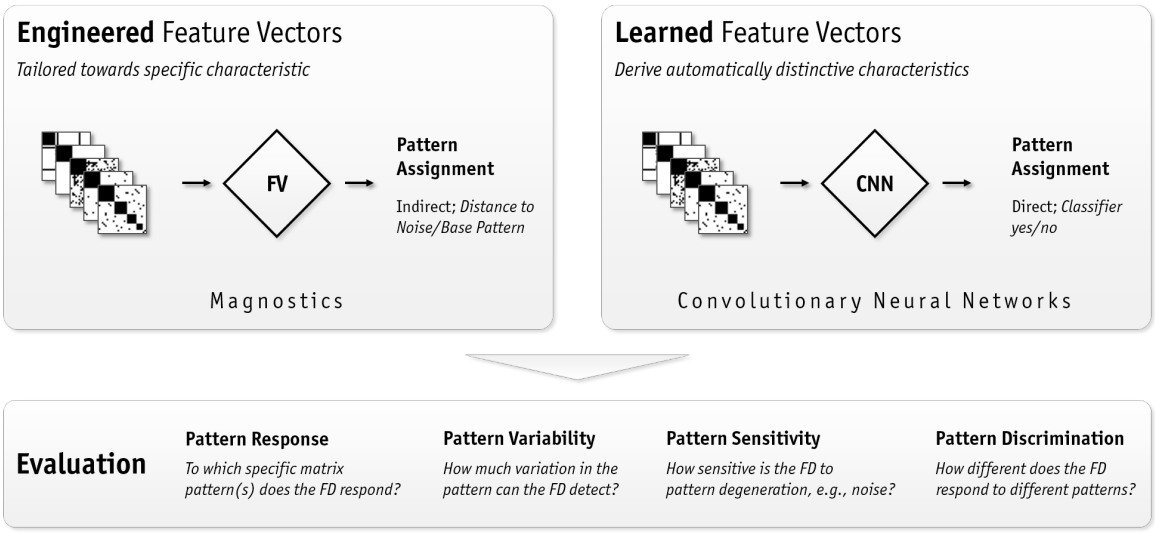


Figure 4.2 Automatic Matrix Pattern analysis approaches are investigated in two subgroups: (1) Engineered feature vectors are extracting specific, designated visual characteristics, (2) Learned feature vector approaches are deriving the potentially interesting characteristics from a given training set. Both approaches are evaluated for their applicability in the matrix-pattern analysis scenario.

user-adaptive classification can help further to answer the question which algorithm is appropriate for the task at hand.

Fundamentally, we differentiate two feature extraction approaches: (1) Engineered Features Vectors as described and developed in Section 4.4 and (2) Learned Feature Vectors as described in Section 4.6. While engineered feature vectors target a specific characteristic to be explored, learned features are obtained by training classifiers for target concepts on the input data. To evaluate the effectiveness of both approaches we describe in Section 4.7 the results of a series of experiments. In these experiments we evaluate our developed and applied feature descriptor approaches with the focus how well they are suited to retrieve and distinguish between specific visual patterns and anti-patterns.

4.4 | Image Feature-Driven Analysis of Matrix Patterns

In this section we focus on finding visually interesting adjacency matrix visualizations for relational data, i.e., networks. A widely used approach to implement search and analysis for relational data relies on so-called image feature descriptors (FDs), capturing certain relevant data properties to compute similarity scores between data elements according to these features. Descriptor-based similarity functions are hence a basis for many important exploration tasks, e.g., ranking data elements by similarity or for computing data clusters according to features.

Yet, the choice of feature vectors and similarity functions is a main research challenge; it often requires knowledge of the application context, and sometimes even the user. To date, a large number of feature extraction methods have been proposed for different types of structured data [RHC99; Sme+00]. However, the descriptors these methods use are often defined in a heuristic way, and yield rather abstract information, which are difficult to interpret and leverage by non-expert users in search and analysis tasks. Thus, it remains difficult to decide which descriptor to choose for a retrieval/ analysis problem at hand.

Recently, image-based features have been used to characterize the visual representation of data [DK10a; Leh+15] with the goal to guide the user in the exploration based on the visual representation. The idea is, for example, to search for an improved view, or to find views of potential interest to the user. Doing so describes the data on a space that is different from the data itself and that instead is based on the characteristic of visual patterns [WAG05c; SSK06; DK10a; Leh+15]. This has the advantage that image representations are closely related to what the user visually inspects, namely, the visualization of the considered data.

Influential for this field is the work of Tukey who formulates the problem that –as the number of plots to interactively inspect increase– exploratory data analysis becomes difficult and time consuming [WAG05c]. Tukey proposes to automatically find the "interesting" plots and to investigate those first. To that end, Wilkinson et. al. [WAG05c] present a set of 14 measures for the quantification of distribution of points in scatter plots, called Scagnostics. Each measure describes a different characteristic of the data and helps, for example, to filter the views with different Scagnostics measures than the majority. The underlying scatter plots are likely to exhibit informative relations between the two data dimensions. Besides static ranking tasks, image-based data descriptions can also form a basis for dynamic training of classifiers to identify potentially relevant views [Beh+14a]. This is particularly useful for cases in which a given (static) description and selection heuristic may not fit some user's requirements.

We propose a set of six FDs, called MAGNOSTICS features, which quantify the presence and salience of six common visual patterns in matrices (presented in Section 2.1.3). Each patterns refers to a topological graph motif, such as clusters, central nodes, or bigraphs. MAGNOSTICS are similar to Scagnostics features describing e.g., the degree of stringyness, clumpiness and outlyingness as relevant patterns in Scatterplots.

Unlike statistical graph measures, which allow describing global graph characteristics, such as density and clustering coefficient, MAGNOSTICS represent interpretable visual features for matrix displays. This is of great importance, because the order or rows and columns in the matrix influences which type of information is visible or hidden from the viewer [Beh+16b], just like in a 2D layout for node-link representations. MAGNOSTICS can be used for a large variety of tasks, such as finding *good* orderings for visual exploration, finding matrices with specific patterns in a large network data set, analyzing a collection

of varied networks, or series of stages in an evolving network (e.g. brain functional connectivity data).

While many FDs for image analysis exist, there is no evidence how they perform for detecting patterns in matrices. In order to make an informed choice of FDs for MAGNOSTICS, we evaluate 30 FDs, including three new descriptors that we specifically designed for detecting matrix patterns. Using a set of 5570 generated matrix images, we evaluated each FD with respect to four criteria: pattern response, pattern variability, pattern sensibility, and pattern discrimination. For each of the FDs that are part of MAGNOSTICS, we provide a more detailed description, showcasing its performance on real-world data sets. We demonstrate MAGNOSTICS on two application scenarios (Section 4.8.1). Firstly, querying a large database by example (query-by-example) and via a sketch interface (query-by-sketch). The second scenario analyses a network evolving over time based on time-series of MAGNOSTICS.

The complete data set, our analysis result, and the sketching interface can be found online: http://magnostics.dbvis.de.

4.4.1 | Feature Descriptor Analysis Methodology

The literature on image analysis provides an abundance of image descriptors, including such based on color, texture, shape, structure, among other properties. These descriptors are traditionally developed and used for processing of real-world images. Our goal is to make an informed selection of candidate FDs appropriate for responding to patterns in matrix visualizations shown in Section 2.1.3. To that end, we started with an initial set of 27 existing, well-known image measures described in the literature. We also include three additional FDs that we designed specifically to respond to patterns in matrix views which by their nature show different properties than real-world images.

Evaluation Criteria

To inform a selection of FDs, we evaluate each FD according to the following four criteria. The individual results are reported in Section 4.4.2:

- **C1: Pattern Response**—*To which specific matrix pattern(s) does the FD respond?* An appropriate FD must distinguish patterns from noise, i.e., a matrix with random distribution of black and white cells. For every pattern in Section 2.1.3, we can generate a set of prototype patterns and measure the performance by precision and recall for every FD on our entire benchmark data set.
- **C2: Pattern Variability**—*How much variation in the pattern can the FD detect?* Patterns in matrices vary, mainly in size or number; for example, there can be one or more blocks, and each of the blocks can have a different size. To measure variability,

we can generate variations for every pattern in Section 2.1.3 and calculate how much the FD response varies/discriminates, using Euclidean distance.

- **C3: Pattern Sensitivity**—*How sensitive is the FD to pattern degeneration, e.g., noise?* Patterns are barely encountered in a pure form. For example, blocks may show holes (less dense clusters), or less sharp boundaries (e.g., overlapping clusters). For every pattern in Section 2.1.3 and its variations, we gradually degrade the pattern until eventually returning a noise matrix (randomly distributed black cells). We develop and derive a pattern sensitivity measure to quantify how well a FD is able to cope with the degeneration of a pattern.
- **C4: Pattern Discrimination**—*How different does the FD respond to different patterns?* An effective FD should yield discriminative results for different patterns. Otherwise, it does not allow to correctly interpret the FD's response. We measure the pattern discrimination by analyzing the differences between vectors returned by the FD.

To inform our selection of MAGNOSTICS FDs, we consider C1 a decisive criteria, meaning that we do not want to include FDs into MAGNOSTICS if they do not respond properly to any patterns. Results for C2 and C3 are descriptive in that depending on the final use case, a more variable (C2) and/or sensitive (C3) FD may be preferred. Alike C1, C4 is considered a decisive criteria as we want FDs to discriminate different patterns. In the following, we describe which initial FDs are included in our analysis, how we generated the benchmark data, and how we analyze the returned feature values for each defined experiment.

Selecting an Initial Set of Feature Descriptors

Table 4.1 summarizes the list of FDs we considered for our analysis. Main selection criteria were the suitability to respond to our set of visual patterns in matrices, availability, and stability of the respective implementations. We also made sure to include two to five FDs from every different image analysis subdomain [RHC99; Sme+00]: texture descriptors, (localized) color-, edge- and line descriptors, shape-, structure- and contour descriptors, interest point descriptors, and noise descriptors.

In addition, we developed three novel FDs specifically for matrix diagnostics. For example, our BLOCK FD not only responds to blocks, but is able to return their number and density (Block Pattern *P1*). NOISE STATISTICAL SLIDINGWINDOW quantifies the (local) amount of noise in a matrix (Noise Anti-Pattern *A1*), while the PROFILE FD is designed to respond to matrices with many lines (Star/Line Pattern *P3*). Details are given in Section 4.4.3.

Creating a Matrix Pattern Benchmark Data Set

We tested each FD against the same data set, i.e., visual matrices. An appropriate data set must contain patterns (C1) and variations thereof (C2), and different degrees of pattern quality (C3). Moreover, we need multiple samples for the same pattern and its variations in order to account for the variability in the individual data samples. We decided to create an artificial controlled benchmark data set to control for the presence, variation, and quality of patterns, as well as to create as many data samples as necessary. Figure 4.3 shows example matrices from the data set. The complete benchmark data set is available online.

Patterns and Variations For every of the five patterns in Section 2.1.3, we generated *prototypes* of 30 × 30 matrices and *variations* thereof (Figure 4.3 first line). A prototype is an image with only the pattern in an otherwise empty matrix. Any purposeful FD must respond to this pattern (C1). A variation of a prototype is a variation of the general characteristics of the pattern, mainly number, size, and position. The goal is to allow conclusions regarding which type of variations a FD is able to differentiate (C2).

For example, a variation of the block pattern varies the number of blocks; a variation of the line patterns changes the line width or the amount of lines, and so forth.

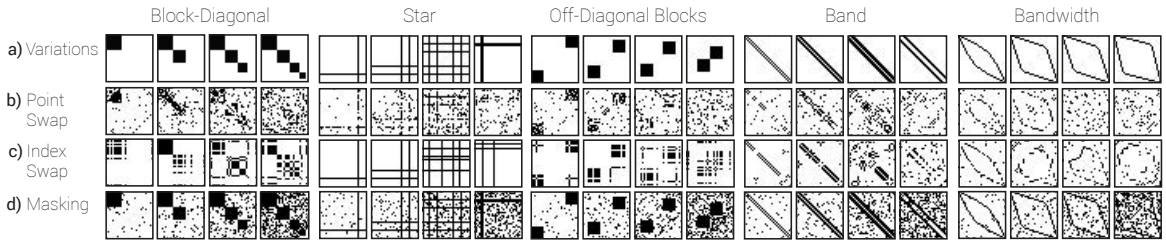


Figure 4.3 Examples of matrices as used in benchmark. Benchmark data set can be downloaded at our MAGNOSTICS website.

Gradual Pattern Degeneration To measure a FD's ability to respond to unclear and noisy patterns (C3), we generated a gradual degeneration schema for each pattern and its variation, i.e., we gradually decreased the quality of the pattern by introducing "noise" into the matrix. We generated matrices with the following *degeneration* functions, making sure that the last steps of every degradation resulted in completely random matrix.

a) Point-Swap—A first type of noise function introduces structural noise into the graph, by randomly swapping cells in the matrix (Figure 4.3b)). Any cell-swap randomly exchanges one black and one white cell in the matrix (symmetry is preserved by a second corresponding cell-swap in each degeneration step). We can consider the number of cell-swaps as a quality measure how salient a pattern is expressed in the data. For example,

Feature Descriptor	Group	Reference
GLOBAL_COLOR_HISTOGRAM	Color	[RHC99]
AUTO_COLOR_CORRELOGRAM		[Hua+97]
FUZZY_HISTOGRAM		[HM02]
FUZZY_OPPONENT_HISTOGRAM		[SGS10]
COLOR_OPPONENT_HISTOGRAM		[SGS10]
THUMBNAIL		[Gra15]
MPEG7_COLOR_LAYOUT	Color Layout	[KY01]
LUMINANCE_LAYOUT		[LC08]
CEDD		[CB08b]
FCTH		[CB08a]
JCD		[Gra15; CB08b]
EDGEHIST	Edge	Java
MPEG7_EDGE_HISTOGRAM		[PJW00]
HOUGH		[Hou62]
SURF	Point of Interest	[Bay+08]
FAST		[RPD10]
BLOCKS	Shape	Java
COMPACTNESS		[Mor]
ECCENTRICITY		[YWB74]
ADAPTIVE_GRID_RESOLUTION		[YKR08]
JPEG_COEFFICIENT_HISTOGRAM	Structure	[LC08; LPM10]
PROFILES		Java
FRACTAL_BOX_COUNTER		[SLM96]
PHOG		[BZM07]
HARALICK	Texture	[HSD73]
GABOR		[LC08]
TAMURA		[TMY78]
LOCAL_BINARY_PATTERN		[HP06]
NOISE_STATISTICAL_SLIDINGWINDOW		Java
NOISE_DISSIMILARITY		[Alb+10]
GRADIENT		[Rüd15]

Table 4.1 Overview over all tested feature descriptors (FDs). FD names are hyperlinks to access an interactive FD profile page with amongst others a distance-to-noise and a distance-to-base ranking.

a complete block without holes represents a clique in the graph, while the presence of holes indicates a less dense cluster. For our data set, we generated pattern degenerations with the following numbers of cell-swaps: 0, 1, 2, 4, 8, 16, 32 percent of the data, with the last step (32%) showing no evidence of any pattern.

b) Index-Swap—With a second noise function, we iteratively swap rows and columns in the matrix (Figure 4.3c)). An index swap preserves the topology of the graph, by randomly exchanging two rows (and columns). Degrading a pattern with an index-swap is similar to using a matrix reordering algorithm that fails to show a topological structure, even though it is present in the data. For our data set, we generated data samples from 0 to 10 index-swaps. We can use the number of index-swaps as quality measure of the respective pattern.

c) Masking—Lastly, we gradually add additional black points (noise) to the matrix to mask the pattern (Figure 4.3d). Thereby, we simulate situations in which the visual pattern is overlapping and intervenes with other patterns, e.g., overlapping clusters, closely connected nodes. The applied noise is exponentially increasing (0% to 16%).

For each of the 23 pattern variations × 24 degeneration types (4 *P1*+ 4 *P2*+ 6 *P3*+ 5 *P4*+ 4 *A2*× 7 point-swap + 11 index-swaps + 6 maskings) we created 10 samples, resulting in a set of 5520 benchmark matrix images with patterns. On top of that, we added 50 pure random noise images with a varying noise density between 1% and 16%, leaving us with a total number of 5570 MAGNOSTICS benchmark matrix images. The full data set can be downloaded from our website: http://magnostics.dbvis.de/#/benchmark.

Setup

For each one of our 5570 matrix images m_i we run each of our 30 FDs. This resulted in 167100 trials $FD_j(m_i)$, each one returning an n-dimensional feature vector $v = FD_j(m_i)$. For most FD (except, BLOCKS, TAMURA and some others), vector dimensions may not have a specific human interpretation. In order to analyze and compare all feature vectors we have to rely on *distances* between individual feature vectors $d(v_i, v_j)$. Our distance function is the Euclidean distance between n-dimensional vectors.

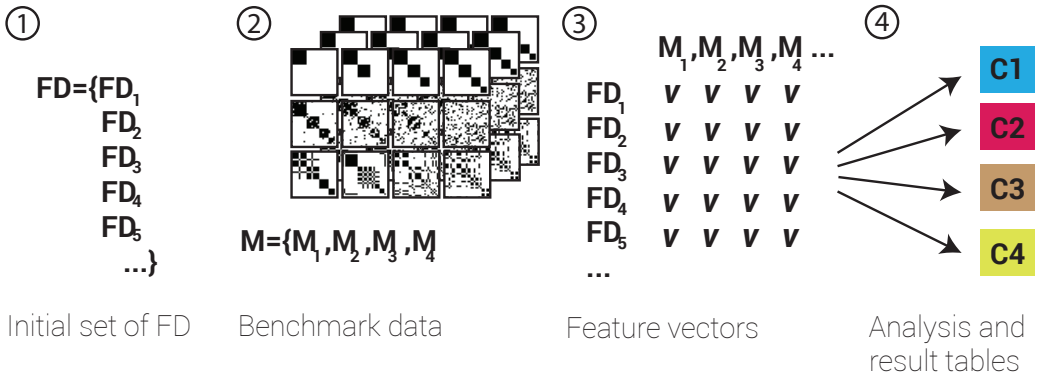


Figure 4.4 FD evaluation methodology and main concepts.

Two FDs (ADAPTIVEGRIDRESOLUTION and HOUGH) had to be excluded from the analysis process, since their implementations did not reliably return comparable feature vectors.

4.4.2 | Analysis and Result Overview

Each criteria (C1-C4) requires an individual analysis of the feature vectors. We report on each criteria individually.

Pattern Response (C1)

Our pattern response criteria refers to a FD's ability to respond to a specific pattern in the data. As a measure of effectiveness, we use precision and recall measures [BR11a]. High precision reflects a FD's ability to rank a larger number of correct answers at the early ranking positions. High recall means that the FD is able to retrieve a large fraction of all correct answers (matrices) from the target data set. The weighted harmonic mean is an aggregate that combines both measures, but prefers recall over precision (F_2 score, with $\beta = 2$). It is given in Equation 4.1 and will be used to assess the capability of the FDs to identify the sought matrices.

$$F_\beta = (1+\beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}} \tag{4.1}$$

Separate precision, recall and F_1 tables can be accessed online[1].

Our experiment setup looks as follows and is repeated for all 30 feature descriptors: First, we derive an appropriate sample data set for our experiment, which consists of all slightly degenerated pattern images ($\leq 4\%$ for point-swap and noise and ≤ 6 index swaps). Second, we conduct a 10-fold cross validation, where each repetition involves the random partitioning of the data into complementary subsets: the training set (70% of the data), which is used to train a classifier, and the validation set (30% of the data) to derive our performance measures. The validation results are averaged over the rounds to reduce the variability. To mitigate the impact of the classifier algorithm we run all experiments with three distinct classifiers (Random Forest, Naïve Bayes, Support Vector Machine). Since the results are comparable we decided just report on the Random Forest Classifier[2].

We present the results of our pattern response experiments in Figure 4.8(a). The F_2 scores are ranging from 0 (low; white) to 1 (high; saturated). As shown in the table, most of the FDs perform with an F_2 score of 0.9 or higher, meaning that these FD are able to differentiate patterns from noise. Generally, texture descriptors yield the highest overall F-scores. None of the patterns were specifically a problem for any of our selected FDs, which means that altogether, our preselected set of FDs is able to respond to each of our six base patterns. Some measures yield higher scores for certain patterns while scoring less for others. For example, MPEG7_EDGE_HISTOGRAM yields lower scores for both block patterns. These results confirm the purpose of an algorithm designed to respond to edges in images. In contrast, BLOCKS performs well for block patterns, but less for lines.

We will report on the choice of each feature descriptor for each pattern separately in Figure 4.4.2.

[1] http://magnostics.dbvis.de/#/evaluation/patternresponse
[2] All other experiment result tables can be found in the Appendix.

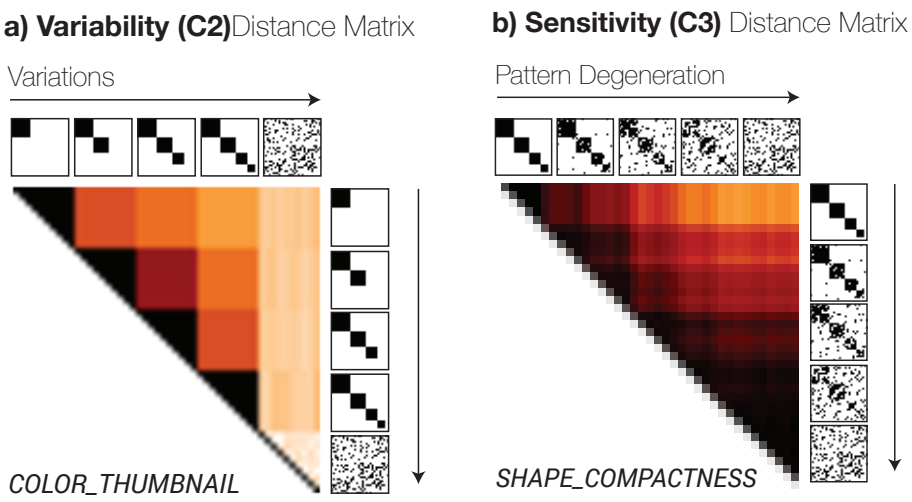


Figure 4.5 Distances between two selected feature vectors (COLOR_THUMBNAIL and SHAPE_COMPACTNESS) of different pattern variations.

Pattern Variability (C2)

Pattern variability corresponds to how sensitive a FD is with respect to variations in the pattern. Therefore, we decided to take only prototypical patterns without any additional degeneration (e.g., noise) into consideration for these experiments. Since no scalar measure exists to describe the pattern variability, i.e., there is so far no vector dimension on the number of blocks or lines, we need to look at pair-wise distances between feature vectors and analyze if they allow us to discriminate the pattern variations.

Figure 4.5 illustrates all pairwise distances of the test patterns between two selected feature vectors in a distance matrix (Descriptor Distance Matrix, DDM). One row and column represents each pattern variation. The last row contains the noise matrix as point of reference for a distance normalization. Black values mean low distances, red represents medium distances, and white reflects high distances. The example in Figure 4.5 shows a desired distribution of distances for the selected FDs (COLOR_THUMBNAIL and SHAPE_COMPACTNESS), i.e., pairwise distances become gradually higher for more variation of the pattern. Hence, we consider these FDs as good candidates to measure gradually the variability of matrix patterns by distance. For better quantification and comparison, we also report on the standard deviation of the normalized distance scores, as shown below.

$$\sigma_{\mathrm{FD}_i}(\mathrm{PV}) = \sqrt{\frac{1}{|\mathrm{PV}|} \sum_{1 \le i,j \le |\mathrm{PV}|} \left(ndist(\mathrm{PV_i}, \mathrm{PV_j}) - \bar{x} \right)^2} \tag{4.2}$$

Where, $\mathrm{PV_i}$ corresponds to the i^{th} pattern variation from a base pattern variation set (e.g., Block1-4), $ndist()$ represents a normalized Euclidean distance and $\bar{x}$ corresponds to the average of all distance combinations. We excluded the reference noise column from this calculation, since it has only illustration purposes.

Other than for precision and recall, variability is no strict criteria for us to exclude a FD from the selection. A FD with low variability supports application cases where *any* expression of a pattern is of importance, e.g., finding matrices with block patterns. A FD with high variability supports applications where more detail is necessary, such as finding matrices most similar to a given one.

We report the results of our pattern variability experiments in Figure 4.8(b). The standard deviation scores are ranging from 0 (low = white) to 0.5 (high = dark red). From the table we see that the only FD with no variability at all is HARALICK. Other FDs score consistently high for our variability measure meaning that they are able to discriminate between the pattern variations. We also found that variations for the off-diagonal and block pattern are lower for most FDs, which might be due to the fact these coherent rectangles are not recognized independent of their location in the matrix plot (i.e., translation invariance). Line patterns seem to yield generally high variability, suggesting that our FDs are sensitive to changes in the expression of lines.

Pattern Sensitivity (C3)

Pattern sensitivity refers to how sensitive a FD is to pattern degenerations (e.g., noise), implying a visually less salient pattern. High sensitivity means that only visually salient patterns can be detected, while a low sensitivity allows even less salient patterns to be detected.

Our analysis approach here is similar to the one for C2; we analyze the Euclidean distances between feature vectors of different degeneration levels (Section 4.4.1). The assumption is that with an increase in degeneration, the distances between feature vectors and the feature vectors for the non-degenerated base pattern follow a *monotonic increase*.

Figure 4.5(b) shows a DDM for a variation of the Block-Diagonal Pattern (4 blocks) and examples of increasing degeneration with the PointSwap modification function. The matrix shows a monotonic increase in distances from the proto-pattern to the degenerated matrix. For our analysis of sensitivity, we only report a one pattern variation for every base pattern. Tests with the other pattern variations showed only small difference. Eventually, we obtained 150 DDMs, one for every combination of the five base patterns and the 30 FDs.

In analyzing the monotony, we (1) assess if there is any monotonic increase or not. If the increase in not monotonic, the respective FD is very sensitive to any degeneration, i.e., the FD cannot accurately rank matrices according to their amount of noise. (2) For monotonic plots, we can then quantify the sensitivity of the FD: highly sensitive (fast increase of distances) or tolerant (slow increase of distances).

Figure 4.6 shows examples of monotonic increases of distances. The x-axis shows the degeneration level, the y-axis shows the distance to the average of one base-pattern vector. Each point in the plot represents the distance of a sample. The red points and line

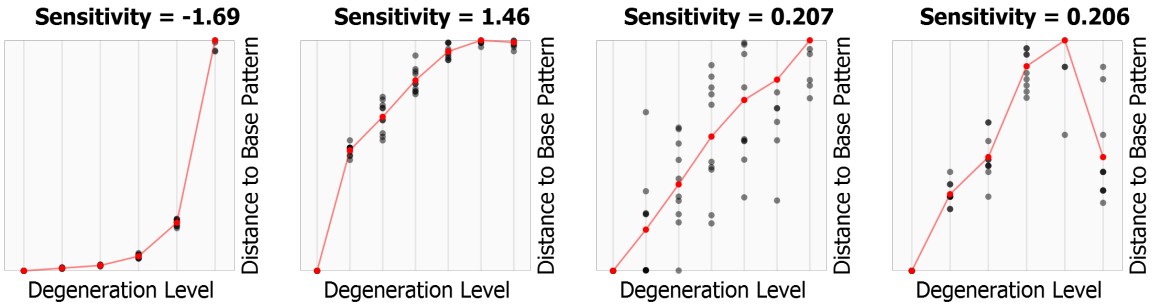


Figure 4.6 Relations between degeneration levels and mean distance (red) to base patterns.

represent average distances. Figure 4.6 (a) shows low sensitivity, (b) high sensitivity, (c) medium sensitivity but with a high variation in the distances, and (d) non-monotonically increasing distances.

To quantify monotony of the 150 DDMs and to better inform our final choice, we calculated the monotony for the feature descriptor FD_i and the base pattern P_j as follows

$$monotony(FD_i, P_j) = \begin{cases} 1, & (mean_{l+1} - mean_l + 0.05) > 0; \\ & \forall l \in noise\_levels. \\ 0, & otherwise \end{cases}$$

where $mean_l$ indicates the average of feature vectors retrieved for the noise level k. For monotonic increases, we then calculated the sensitivity as the *signed difference* between the averaged vector for each degeneration level, and the normalized equality function $x = y$. $sensitivity(FD_i, P_j)$ is high for $sensitivity > 0$ (fast increase), and low (slow increase) for $sensitivity < 0$.

$$sensitivity(FD_i, P_j) = \sum_{l=0}^{|noise\_levels|-1} \left((mean_k - (\frac{l}{|noise\_levels|}) \right)$$

We show the results of our pattern sensitivity analysis for the point-swap modification function in Figure 4.8(c). Cross signs mean that the respective FD did not yield a monotonic increase of distances. The entire overview of our results for all degeneration functions and base patterns, as well as every distance distribution, can be explored on our interactive website[3].

In summary, most feature vectors show low sensitivity (i.e., are tolerant) to our masking function. However, for structural degeneration (point-swap and index-swap), there are significant differences in the sensitivity of all FDs. In most cases, a FD is tolerant for all

[3] magnostics.dbvis.de/#/evaluation/patternsensitivity

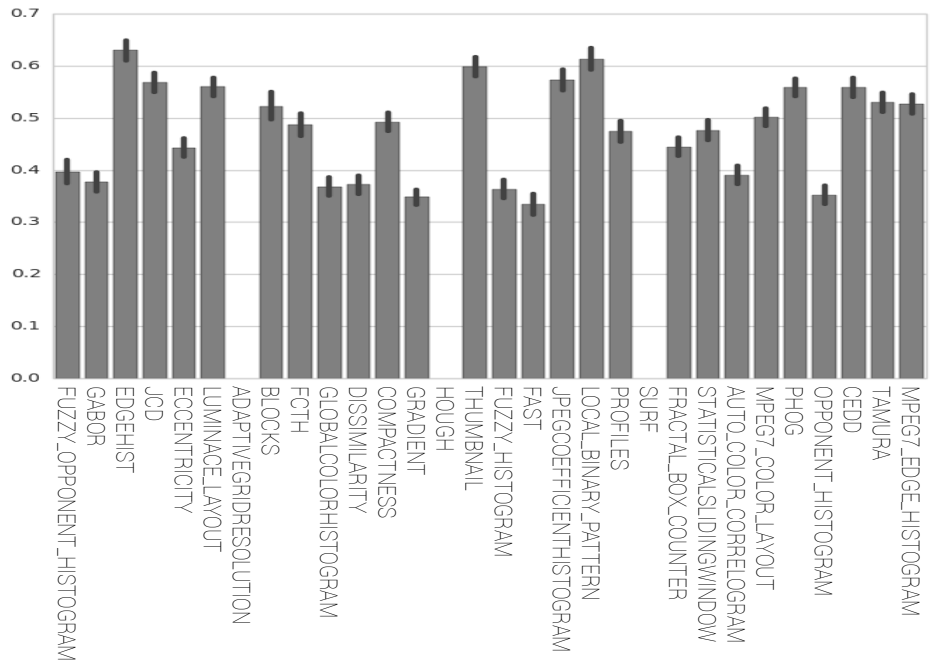


Figure 4.7 Pattern Discrimination (C4): High mean values indicate high distances between feature vectors for the individual patterns (CI=95%).

or for none base pattern. In our analysis, we found that MPEG7_COLOR_LAYOUT and GABOR performed best across all modification functions.

Pattern Discrimination (C4)

Our last criteria focuses on how much an individual FD is able to discriminate different patterns, i.e., blocks, lines, etc. A high discrimination means that an FD allows distinguishing between patterns. The respective DDM must look similar to Figure 4.5(a).

To measure discrimination for a given FD, we investigate the pairwise distances between the feature vectors v for all patterns. In order to obtain a single measure for all the pairwise differences, we report on the mean values of distances, as well as the standard deviation. High mean distance values can be interpreted as high average distances among feature vectors for different patterns, suggesting a high ability to discriminate between patterns. A high standard deviation can be interpreted as a balance between pair-wise distances, and suggest that this particular FD can discriminate only a few patterns.

Figure 4.7 shows mean values and confidence intervals for all FDs. EDGEHIST shows highest means in distances while having rather small confidence intervals. Similar are THUMBNAIL and PHOG. High standard deviations are found with SLIDINGWINDOW and ECCENTRICITY. Examples for low discrimination FDs are TEX_GRADIENT and DISSIMILARITY.

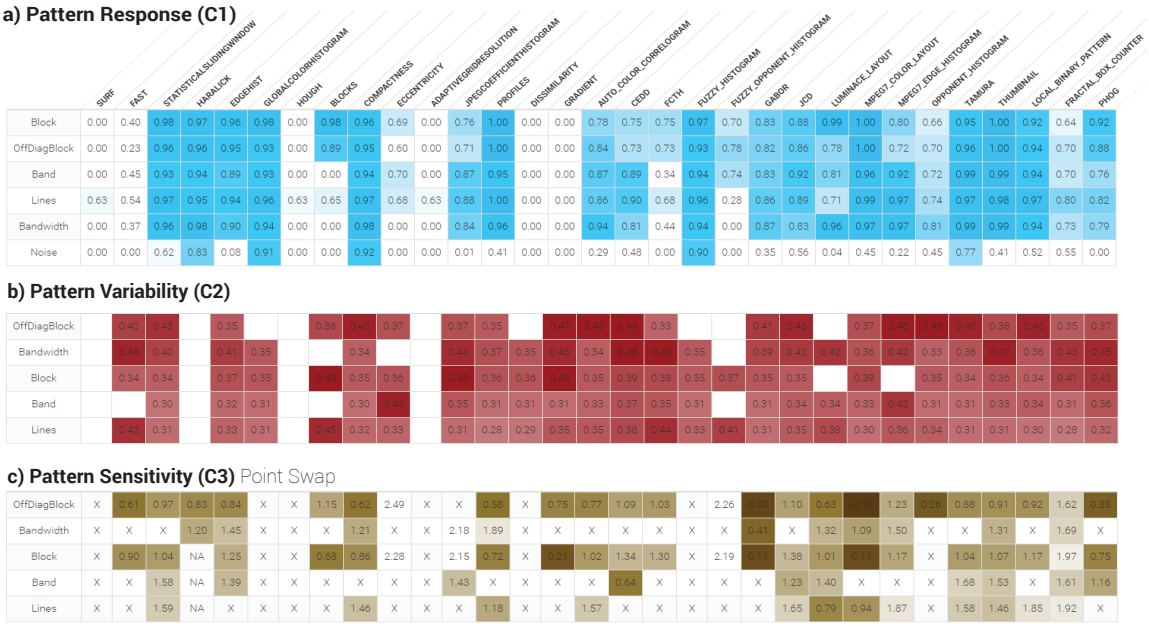


Figure 4.8 Analysis result overview for all FD and patterns: (a) C1: numbers present the F2-score; higher scores (darker) indicate better response to the pattern, (b) C2: high values (darker) indicate higher variability between patterns, (c) C3: low values (darker) indicate lower sensitivity.

Figure 4.9 Result table showing all values. Colors correspond to criteria (blue=C1 (response), red=C2 (variability), brown=C3 (sensitivity)). Black dots in each colored rectangle indicate our subjective ranking; 3 dots represent high ranks, 0 dots indicate low ranks.

Selecting Feature Descriptors

In the last Sections we quantitatively analyzed the differences between FDs with respect to their pattern response (C1), to their ability to respond to pattern variations (C2), to their sensitivity to degenerations (C3), and which patterns a FD is able to discriminate. Overall we found that some measures perform equally for the same criteria. Figure 4.9 summarizes our results across all criteria.

We can now choose a purposeful subset of FDs that allows us to describe our selection of patterns in matrices. The exact choice of FDs required for a specific application may

vary, but starting with our set of selected FDs, which together form the MAGNOSTICS set, is expected to be a suitable fit. We disregarded FDs, which did not perform well with respect to our criteria C1-C4, or which were outperformed by another FD for the same pattern.

We generally found that some patterns, e.g., blocks or off-diagonal blocks, appear to be easier to detect for our FDs than others, e.g., bands and bandwidths. This may be due to the fact that these patterns are (mostly) positioned around the diagonal/counter-diagonal. Especially, for lines our FDs allowed for little translational invariance (shifts in the x-/y- positions). For our selection we also found that we should incorporate trade-off considerations between pattern response (C1) and the pattern variability (C2), since for some patterns more variations can be expected then for others. The final choice of FDs for MAGNOSTICS is detailed in Section 4.4.3.

4.4.3 | Engineered Image Space Feature Descriptors for Matrix Structures and Patterns

We selected six FDs for MAGNOSTICS that we found most purposeful according to our experimental comparison in the preceding section. For each FD we briefly describe its name, functionality, which patterns it detects, and report on details on variability (C2), stability (C3), discrimination (C4), and show examples of real-world networks the FD could retrieved.

Block Descriptor *P1*

We designed the BLOCK descriptor as a heuristic to measures the *blockiness* of matrix plots *along* the diagonal, thus allowing us to retrieve matrices with a Diagonal-Block matrix form *P1*. The BLOCK FD, shows to be good in all the experiments (C1-4) with a F_2 score of 0.98, a high variability of 0.48 and a good sensitivity score of 0.68 for all degeneration functions.

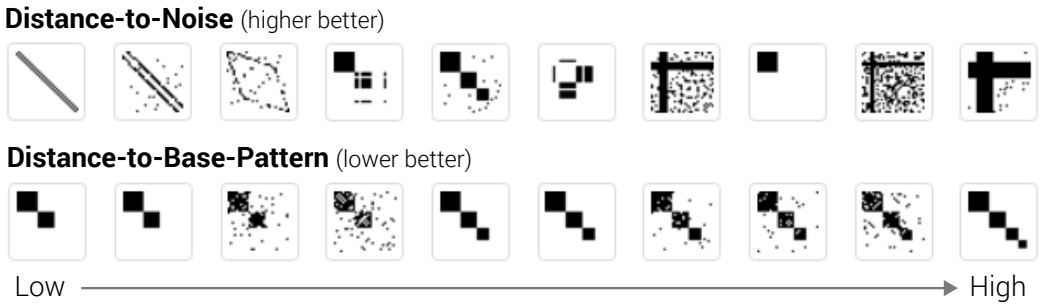


Figure 4.10 Examples for our Block Descriptor, specifically engineered to retrieve blocks around the matrix diagonal.

In a sliding window approach, the descriptor linearly scans for rectangles of a minimum size (width/height) and density (black to white ratio). Found blocks are iteratively enlarged if either the horizontal, vertical and diagonal direction leads to a block density increase. Since found blocks can overlap, we are removing in a postprocessing step all blocks, which are fully covered by other blocks. In contrast to the minimum density factor, we calculate additionally a factor describing the separatedness/distinctiveness of a block and retain only those blocks which are perceptionally distinct from the surrounding.

Local Binary Pattern Descriptor *P2*

The Local Binary Pattern (LBP) texture descriptor [HP06] is classically used for background-foreground detection in videos. In our experiments the descriptor showed to be responding to the *off-diagonal block patterns P2*. In comparison to our BLOCK FD, which is designed for blocks along the matrix diagonal, LBP adds an additional off-diagonal component to the MAGNOSTICS feature vector. Is performs good for the C1 and C2 experiments, with a F_2 score of 0.9, a high variability of 0.46, but appears to be sensible to index swaps. An alternative choice for Off-Diagonal Blocks *P2*would have been the TAMURA textual FD, which is significantly more sensible to noise.

In a sliding window approach the FD constructs histograms of pixel intensities, called local binary patterns (LBP) for a central pixel to N neighboring pixels.

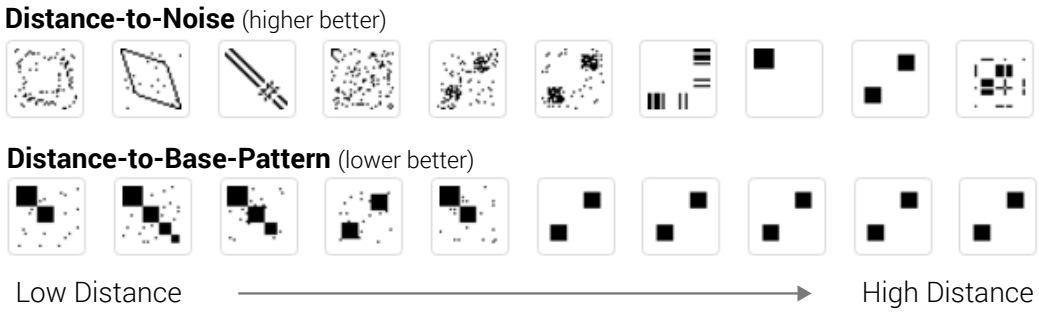


Figure 4.11 Examples for the Local Binary Pattern Descriptor.

Profile Descriptor *P3*

We designed the PROFILE descriptor with the aim to describe *lininess* characteristics *P3* (many/ few short/ long lines) in matrix plots. In our experiments the FD responds with a perfect F_2 score of 1.0, and distinguishes clearly between the base pattern variations, thus leading to a quite low variability score of 0.28. Alike all other FDs it reacts moderately to noise. However, C1 and C2 make this FD especially suited for query-by-example search tasks. The PROFILE computes two axis-aligned histograms of the plot, where every matrix row, respectively column, represents one histogram bin and the bin's value corresponds

the number of black pixels within the respective row. In order to achieve translation invariance (i.e., an otherwise empty matrix with just one row/column line should be equally scored independent of the line's location) we are computing a standard deviation from the profile histogram with the intuition that matrix plots with many lines will show high values, while nearly empty matrices or highly blocky matrices will show low values (few jumps).

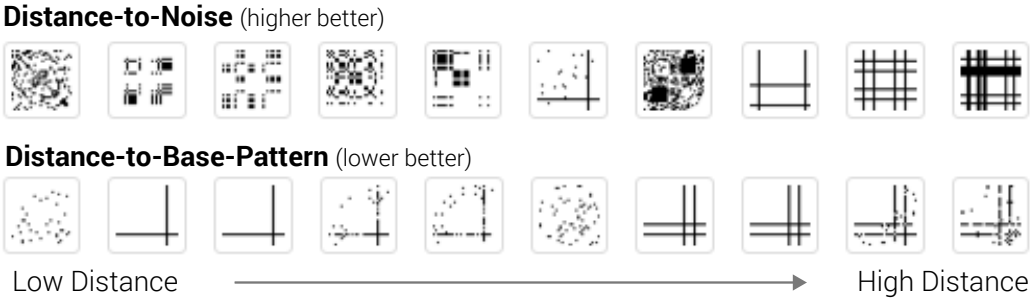


Figure 4.12 Examples for our Profile Descriptor.

MPEG7 Edge Histogram *P4*▨

The MPEG7 Edge Histogram FD [PJW00] can be used to retrieve lines along the diagonal *P4*▨. It responds most often to the band pattern (F_2 score of 0.92) and allows for some variability (0.42), but has –alike all other FDs– problems to deal with pattern degenerations, which might be due to the high specificity of the pattern. The FD subdivides an image into 4×4 sub-images. From each sub-image an edge histogram (5 bins with vertical, horizontal, 45-degree diagonal, 135-degree diagonal and non-directional edge types) is extracted [Won05].

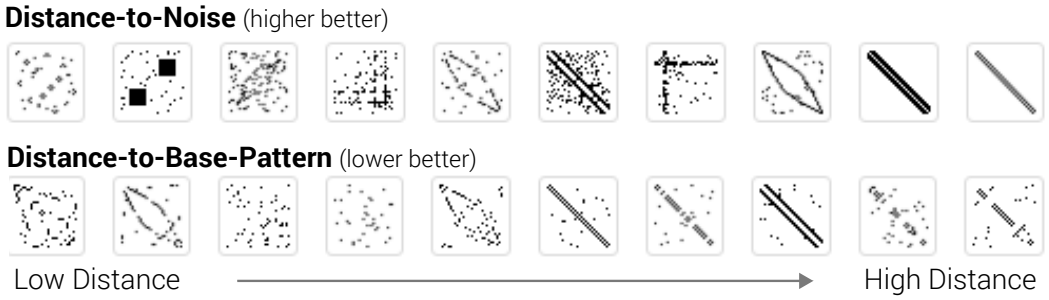


Figure 4.13 Examples for the MPEG7 Edge Histogram Descriptor.

Color and Edge Directivity Descriptor (CEDD) *A2*▨

The CEDD descriptor [CB08b] showed a good response to the bandwidth pattern. It incorporates color and texture information in a histogram form. While the texture (edge)

information uses the same MPEG7 EHD implementation as described above, the color histogram is constructed from an adaptive binning of the HSV color space.

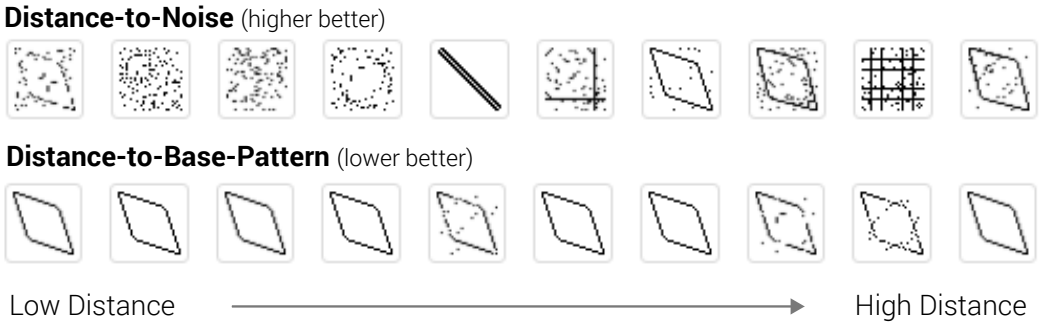


Figure 4.14 Examples for the CEDD Descriptor.

Although, CEDD has not the best scores in our MAGNOSTICS evaluation (F_2 of 0.81, Variability of 0.48), we decided to include the FD. CEDD, alike the closely related FCTH, outperform the other FDs in terms of variability, which is an important factor for bandwidth patterns (*A2*▨). These patterns are often the result of graph-based reordering methods (e.g., the Cuthill-McKee matrix reordering), which enumerate row-/column permutations in a breath-first search leaving an envelope shape behind.

Haralick Texture Descriptor *A1*▩

The Haralick FD [HSD73] is one of the classical texture descriptors for images. It responds quite reliably to the noise anti-pattern *A1*▩. For this pattern we conducted only the C1 experiments, since C2-C4 are not meaningful. We decided to include Haralick even though its F_2 score is only 0.83, which is less accurate than three other color intensity approaches (GLOBAL_COLORHISTOGRAM, FUZZY_HISTOGRAM, COMPACTNESS). However, Haralick is generally more expressive and reliable in our experiments.

We also experimented with our own STATISTICALSLIDINGWINDOW noise descriptor, which regards the sliding window values as a time series of differences for subsequent regions in the image. On this time series we calculated average, variance, and standard deviation.

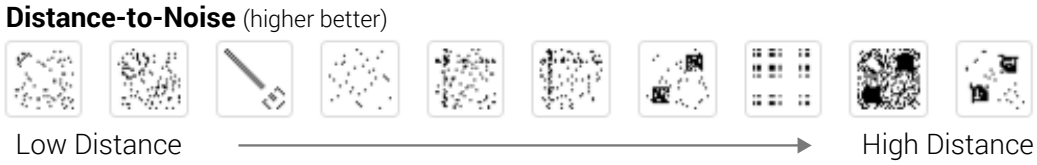


Figure 4.15 Examples for our STATISTICAL SLIDING WINDOW Descriptor.

4.5 | Data Space-Driven Analysis of Matrix Patterns

Alternatively to the aforementioned image-space feature descriptors we presented in [Beh+14b] a method that extracts pattern-related information from the data space. Unlike image space measures, which allow for a rescaling of the input images, data space feature extraction and distance measures have to be able to cope with matrices of varying sizes. Therefore, we developed a projection-based distance calculation detailed in the following sections.

4.5.1 | Projection-Based Distance Calculation for Heterogeneous Matrix Plots

We regard a matrix as a set $\mathbb{M}$ of high-dimensional row or column vectors. Two sets $\mathbb{M}_1, \mathbb{M}_2$ of vectors can be compared by computing an aggregate vector-based similarity score. If the matrices have the same size then we can easily compare them, e.g. using the Euclidean distance. However, for matrices of different size this is not possible. In the following, we introduce an alternative approach.

Figure 3.9 illustrates our distance calculation approach. A set of matrices $\mathbb{M}_1, \ldots, \mathbb{M}_n$ which vary in size is shown on the left. It is possible to interpret either the rows or the columns as high-dimensional vectors of the length of the respective matrix. Assuming a topology-preserving projection, we project the matrix vectors to the plane (see the center of Figure 3.9).

It is necessary to consider the effects of the projection for our targeted distance calculation. Projection methods are conventionally used to reduce the dimensionality of data by reducing the number of dimensions while minimizing information loss (in this case, metric properties) [CK04]. They can be separated into linear and non-linear techniques. Linear projections attempt to separate important from unimportant dimensions to form the projection space. Non-Linear projections try to preserve the high-dimensional *neighborhood* properties, such as pairwise distances, in the low-dimensional space and seek to separate projection points whenever their high-dimensional counterparts are far apart. While linear projection techniques are only able to preserve linear structures, non-linear projection techniques can also take arbitrarily shaped clusters or curved shapes into account.

As Kosinov and Caelli show in [KC02], with an appropriate choice of projection, the projected vectors will reflect the similarity relationships present in the original data. Consequently, a matrix $\mathbb{M}_i$ can be represented by a number of $|\mathbb{M}_i|$ (only rows or columns) or $2 \times |\mathbb{M}_i|$ (rows and columns) projection points in the plane. Since we only consider adjacency matrices they are always square. The use of only rows or columns applies to

symmetric matrices (undirected graphs), while non-symmetric matrices (directed graphs) benefit from simultaneous row and column projections.

In the case of projection techniques based on eigendecomposition, we can take advantage of a well-known property: The eigenvalue spectrum of a matrix is *invariant* with respect to similarity transformations. For any non-singular matrix $\mathbb{M}$ and compatible, invertible matrix $\mathbb{P}$, the product $\mathbb{P}\mathbb{M}\mathbb{P}^{-1}$ has the same eigenvalues as $\mathbb{M}$. This means that the spectrum of a graph represented by its adjacency matrix is not affected by row/column permutations [KC02]. This result is important, since it allows us to compute of the distance between two matrices irrespective of the ordering of rows/columns.

To compare two matrices, we match the projected points of one matrix $\mathbb{M}_1$ to the projected points of the another matrix $\mathbb{M}_2$. This matching has the requirement to be a close-optimal solution, meaning that (nearly) no better allocation of the projection points of $\mathbb{M}_1$ to $\mathbb{M}_2$ can be found. Having obtained an matching of the projected point sets, we compute an aggregate distance score over the matching, as shown in Figure 3.9 (right). This distance score can be used for a range of applications, such as similarity search: Given a matrix, return the most similar matrices from a larger data set to be explored; or to generate a clustering of a set of matrices. It is also possible to interpret the projected vectors (point clouds) as a complementary form of matrix visualization, as we already seen in Section 3.6.1. Also the presented distance calculation approach can easily be embedded in visual analytics tasks, which we will show in the dedicated visual analytics Chapter 5. In this way, it is possible to visually identify outliers in the two-dimensional space that correspond to outliers in the high-dimensional space. Furthermore, we can use linked interactions between the projected representation and the traditional matrix visualizations to help users interpret and interact with the matrix.

As explained before we can take advantage of the eigenspectrum's property of being invariant to row and column permutations. In our case, we can project vertex connectivity data from a graph adjacency matrix to a smaller set of its most important eigenvectors. The projection coordinates obtained this way then represent the relational properties of individual vertices relative to the others in the lower dimensional eigenspace of the graph. In this eigenvector space, *structurally similar* vertices or vertex groups will be located close to each other, which can be used for approximate comparison and matching of graphs [KC02]. This is shown in Figure 3.10 and Figure 3.11: The first depiction shows the projection of two similar adjacency matrices, while the second projection view depicts a higher distance score.

A positive side-effect of projection is that the low-dimensional representations of the input vectors may remove the impact of outliers.

As we use projection to compare matrices, we require *determinism* of the projection calculation, that is, the projection calculation needs to produce the same output on the same data set. This statement does not hold for projection techniques, in which an initial

seed population is projected, followed by the projection of all other points based on their high-dimensional (dis-)similarity. t-Distributed Stochastic Neighbor Embedding (t-SNE) [MH08] or Kohonen's Self-Organising Maps (SOM), as described in [Car97], are representatives of possibly non-deterministic projection approaches, and we therefore exclude them from our approach.

In our implementation we apply one exemplary linear and one non-linear projection technique to our data sets. Both implementations are based on eigendecomposition approaches. The linear projection we chose was *Principle Component Analysis (PCA)* [Jol86], which reduces the dimension of the data by finding orthogonal linear combinations of the original variables with the largest variances. The non-linear projection technique was *Classical Scaling (Metric Multidimensional Scaling)* [CC00b]. The implementation of PCA and Classical Scaling was taken from the Projection Explorer Framework of Paulovich [POM07].

Comparing Matrices By Graph Matching

As a result of the projection of a pair of matrices $\mathbb{M}_1$ and $\mathbb{M}_2$ into the plane, we obtain two point sets representing the matrices. We compare the matrices by solving a bipartite graph-matching problem on the point sets in the projected space. One possible solution is to calculate the minimum sum of pairwise Euclidean distances between the points' projection coordinates.

We now describe our distance calculation for the cases of equal and unequal size of the input matrices in detail.

Matrices of Equal Size. For $|\mathbb{M}_1| = |\mathbb{M}_2|$ we have to find $|\mathbb{M}_1|$ edges connecting the two vertex sets. This computational problem is referred to as the *stable marriage problem*; where a set of n men and n women, marry them off in pairs after each man has ranked the women in order of preference from 1 to n, $w_1, ..., w_n$ and each women has done likewise, $m_1, ..., m_n$. If the resulting set of marriages contains no pairs of the form m_i, w_j, m_k, w_l such that m_i prefers w_l to w_j and w_l prefers m_i to m_k, the marriage is said to be stable [GS62; PS03]. Gale and Shapley showed that stable marriages exist for any choice of rankings [GS62].

For the matrix comparison task, we can use the Euclidean distances between the projection point coordinates from $\mathbb{M}_1$ and $\mathbb{M}_2$ to compute the (men-)optimal allocation for the two matrices. The optimality proof is given in [GS62]. After having found a pair matching, we can define the sum over all pairwise Euclidean distances as the distance between $\mathbb{M}_1$ and $\mathbb{M}_2$. This is shown in Equation 4.3:

$$Dist(\mathbb{M}_1, \mathbb{M}_2) = \sum dist_{proj}(m, n) \tag{4.3}$$

where $m \in \mathbb{M}_1$, $n \in \mathbb{M}_2$ and m and n are matching partners. In our implementation, we use the *Extended Gale-Shapley algorithm* [GI89], which is a performance improvement over the classical Gale-Shapley algorithm [GS62].

Matrices of Unequal Size. When $|\mathbb{M}_1| \neq |\mathbb{M}_2|$, the matching defined previously needs to be modified. It is still possible to calculate the men-optimal allocation between the projection points, but only $\min(|\mathbb{M}_1|, |\mathbb{M}_2|)$ can be allocated. For the remaining points a penalty can be added to the overall distance score. Accordingly, the distance function from Equation 4.3 can be adapted in the following way:

$$\begin{aligned} Dist(\mathbb{M}_1, \mathbb{M}_2) = &\sum dist_{proj}(m,n) \\ &+ (\max(|\mathbb{M}_1|, |\mathbb{M}_2|) - \min(|\mathbb{M}_1|, |\mathbb{M}_2|)) \\ &\times Penalty(\mathbb{M}_1, \mathbb{M}_2) \end{aligned} \tag{4.4}$$

where $m \in \mathbb{M}_1$, $n \in \mathbb{M}_2$ and m and n are matching partners. The penalty function $Penalty(\mathbb{M}_1, \mathbb{M}_2)$ can be chosen in accordance with the task at hand. In the current implementation the penalty calculation schemes, depicted in Equation 4.5, can be interactively chosen:

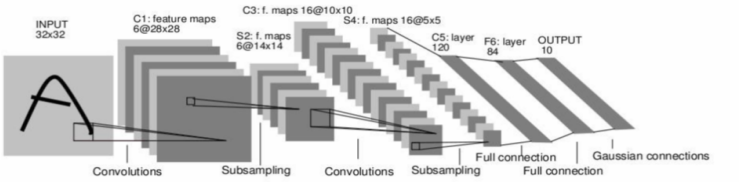
$$Penalty(\mathbb{M}_1, \mathbb{M}_2) = \begin{cases} 0 \\ \max(dist_{proj}(m,n)) \\ \max(dist_{proj}(m,n))^2 \end{cases} \tag{4.5}$$

where $m \in \mathbb{M}_1$, $n \in \mathbb{M}_2$, and m and n are matching partners. The penalty can be set to zero (`ZeroPenalty`), or multiples of the maximum distance (`MaxDist`) or its square (`MaxDistSquare`), among all matches. Whenever the matching results in a small maximum matching distance score, only a small penalty will be added. Whenever the maximum matching distance is large, a large penalty will be added. We will investigate the impact of changing the distance score penalty function in Section 5.6 were we also focus on the visual analytics aspects of the presented approach.

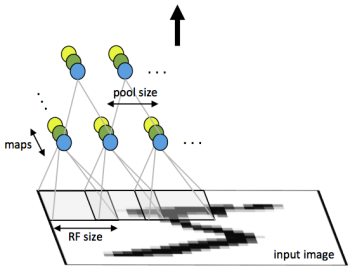
4.6 | Learned Feature Analysis for Matrix Patterns

As an alternative to engineered feature extraction approaches, unsupervised algorithmic approaches can be applied to *learn* generative models that represent the dominant features of an image. In a training phase, the learning method is confronted with many images and the learning method determines the features to be modeled.

During the last years, Convolutional Neural Networks –short CNNs– were gaining increasingly importance in the field of automatic feature extraction and image recognition.



(a) General Structure of a Convolutional Neural Network. Graphic courtesy [LeC+98].



(b) First layer of a convolutional neural network with pooling. Units of different color represent different filter maps [Ng+13].

Figure 4.16 Structure and Principles for Convolutional Neural Networks.

The core intuition behind CNNs is to find a set of locally connected neurons (called feature maps), which represent some dominant image characteristic. Feature maps are determined by an alternating sequence of the convolution operator, modeling a filter step, and a subsampling unit to reduce the feature/decision space. Figure 4.16a depicts the core structure of a CNN as presented by LeCun et al. in 1998 [LeC+98].

"A CNN consists of a number of convolutional and subsampling layers optionally followed by fully connected layers. The input to a convolutional layer is a $m \times m \times r$ image where m is the height and width of the image and r is the number of channels, e.g. an RGB image has $r = 3$. The convolutional layer will have k filters (or kernels) of size $n \times n \times q$ where n is smaller than the dimension of the image and q can either be the same as the number of channels r or smaller and may vary for each kernel. The size of the filters gives rise to the locally connected structure which are each convolved with the image to produce k feature maps of size $m - n + 1$. Each map is then subsampled typically with mean or max pooling over $p \times p$ contiguous regions where p ranges between 2 for small images (e.g., MNIST) and is usually not more than 5 for larger inputs. Either before or after the subsampling layer an additive bias and sigmoidal nonlinearity is applied to each feature map. The figure below illustrates a full layer in a CNN consisting of convolutional and subsampling sublayers. Units of the same color have tied weights [Ng+13]".

4.6.1 | CNN Architecture

The overall architecture of the convolutional neural network we used for our MAGNOSTICS Pattern Benchmark set is described in the following. Similar to the MNIST handwriting recognition benchmark dataset [LeC+98], which is classically used to evaluate the performance of pattern recognition systems, our visual matrix pattern benchmark dataset is of size 30×30 (width and height). Therefore, we assume that the good performance of CNNs

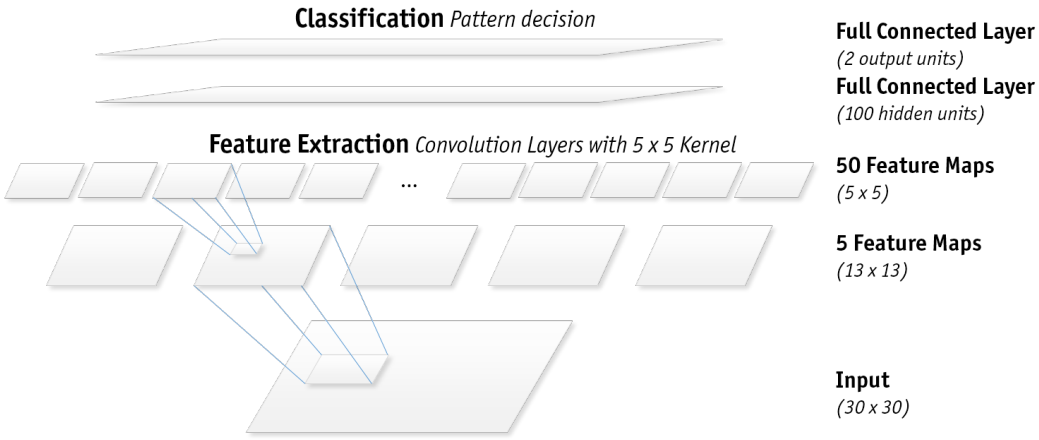


Figure 4.17 Architecture of our CNN for the Retrieval of Matrix Patterns

on the MNIST dataset is an indicator for the retrieval performance of CNNs on similar datasets. As a consequence, we apply the same layer architecture for the matrix pattern analysis.

We also tested a range of other CNN architectures, which all showed in pilot studies worse results than the once reported in the following.

As Figure 4.17 depicts, the first two layers in our architecture are convolution layers representing a trainable feature extractor. The kernel size of 5 is used for the feature extraction. The size is experimentally validated as trade-off between necessary overlap and redundant information [LeC+98]. After the convolutionary layers a universal classifier in the form of two fully connected layers is used to generate a pattern "existence" decision. The number of hidden units was also experimentally validated by LeCun et al. [LeC+98].

4.6.2 | Experiment Setup and Benchmark Dataset

Our experiment setup for the CNN pattern retrieval experiments looks as follows:

1. We constructed –similarly to the MAGNOSTICS pattern response evaluation- an appropriate classifier training dataset with less degenerated benchmark images; i.e., ≤ 4% for point-swap and noise and ≤ 4% index swaps. Our selection is detailed below. Table 4.2 gives an overview about the number of images, which fulfill these properties.
2. We conduct a quasi 10-fold cross validation, where each repetition involves the separation of 20% of all benchmark data (1,114 images) for validation purposes.
3. We derived the following performance measures/meta information for each trail: Precision, Recall, F1 Score, F2 Score, CNN Training Time, File Ranking with final binary pattern decision and classification score.

Pattern	BlackWhitePointSwap ≤ 4%	IndexSwap ≤ 4%	Masking ≤ 4%	Combined
Block	160	200	160	520
OffDiagBlock	160	200	160	520
Lines	240	300	240	780
Band	200	250	200	650
Bandwidth	160	200	160	520
Noise	50	50	50	50
Sum	970	1,200	970	3,040

Table 4.2 Composition of the evaluation dataset. The numbers in the table reflect the number of matrix images for a specific base pattern and modification method combination.

As stated above, we use the same less degenerated base pattern matrices for the MAGNOSTICS evaluation and the CNN experiments. Thus, we are able to compare the experiment results and examine the usefulness of both approaches in a comparative manner. In particular, we use all matrix images with a (1) BlackWhitePointSwap modification level ≤ 4% (2) IndexSwap modification level ≤ 4% (3) Masking modification level ≤ 4% (4) 50 random noise images. Table 4.2 shows the pattern and degeneration degree distributions for the benchmark data set used in the MAGNOSTICS approach and CNN experiments.

In order to reach a statistically more significant evaluation we repeated all experiments ten times, leaving us with 100 precision and recall values for each experiment condition (one base pattern). All experiments were conducted on the Scientific Compute Cluster in Konstanz (SCCKN), a platform for High Performance Computing (HPC) and High Throughput Computing (HTC, "Big Data") at the University Konstanz. In total, 1366h 21min 35sec computing hours were used to train all CNN classifiers.

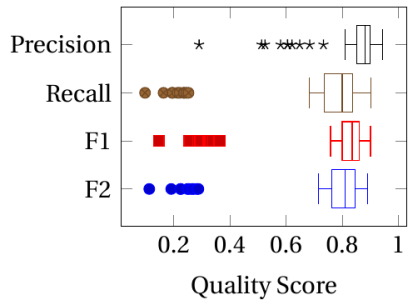
Now we will report on the standard information retrieval performance measures: precision, recall, the combined weighted harmonic mean measures of the former: F1 and F2 (favoring more precision, respectively recall). We give further information whenever the experiment results are not obvious. Additionally, we report on the Class Training Score progression, which is an indicator how much certainty is added in every iteration of the classier training.

Lastly, we conducted a multi-label classification CNN experiment in which the classifier differentiates between the six pattern class labels. This experiment is detailed in Figure 4.6.2.

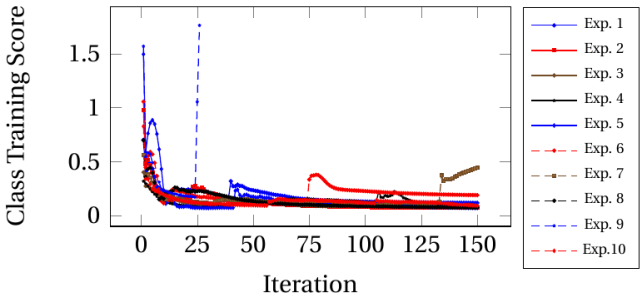
CNN Experiment Results: Block Pattern

The Block CNN shows on average a good retrieval performance with a precision of 0.85 and a recall of 0.74. The average CNN training duration was 22:50:07 for 150 iterations. Only one CNN experiment showed no training convergence, thus leading to degraded retrieval performance results.

Generally, Block patterns seam to be reasonably well recognizable with CNNs.



(a) Performance measures for Block Pattern CNN.



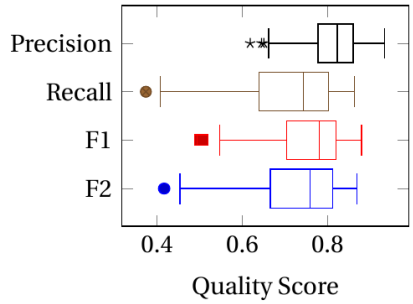
(b) Block Class Training Score Progression; Experiment 9 (max. 18,971.00) shows no convergence.

Figure 4.18 Results for the Block Pattern Retrieval CNN Experiment.

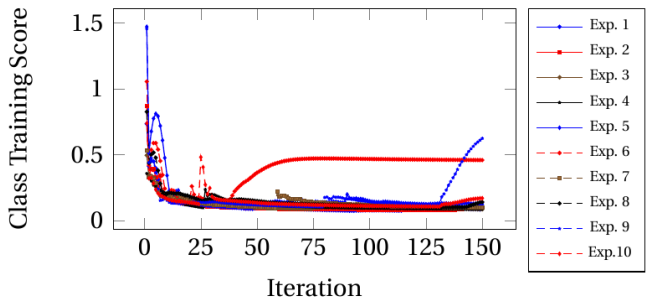
CNN Experiment Results: Off-Diagonal Block Pattern

The Off-Diagonal CNN shows a comparable retrieval performance as the Block CNN with an average precision of 0.81 and a recall of 0.71. The average CNN training duration was 22:50:07 for 150 iterations. Totally, 228h 45min 56s computing hours were used for the training. All CNN experiments showed training convergence.

Generally, Off-Diagonal Blocks seam to be reasonably well recognizable with CNNs.



(a) Performance measures for Off-Diagnoal Block Pattern CNN.



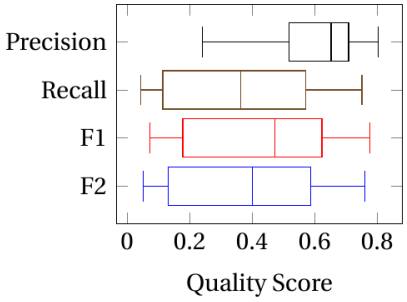
(b) Off-Diagonal Block Class Training Score Progression

Figure 4.19 Results for the Off-Diagnoal Block Pattern Retrieval CNN Experiment.

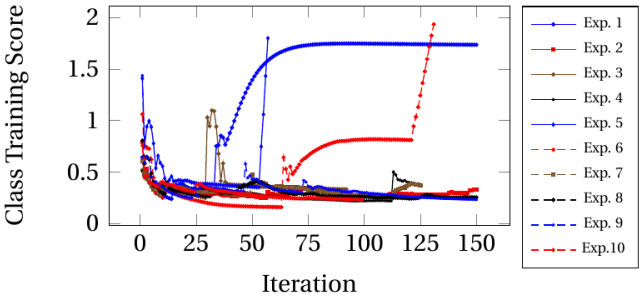
CNN Experiment Results: Star/Line Pattern

The Star/Line CNN shows a bad retrieval performance with an average precision of 0.61 and a recall of just 0.36. The average CNN training duration was 22:45:25 for 150 iterations. Totally, 227h 34min 17sec computing hours were used. Only four of the ten CNN experiments showed training convergence.

Generally, CNNs seam to have problems recognizing Star/Line patterns. This might be due to the reason that the benchmark images are quite diverse and show a lot of pattern variability (differences among the amount of lines, line width, etc.).



(a) Performance measures for Star/-Line Pattern CNN.

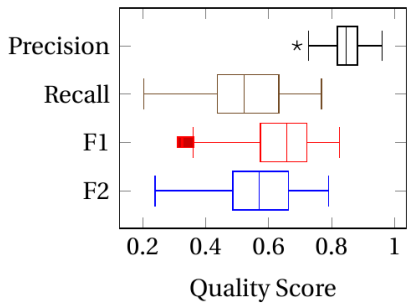


(b) Star/Line Pattern Class Training Score Progression; Exp. 3 (max. 2,698.55), 4 (max 104.99), 5 (8.19), 6 (5.38), 7 (max 74.92), 10 (max 417.42) show no convergence.

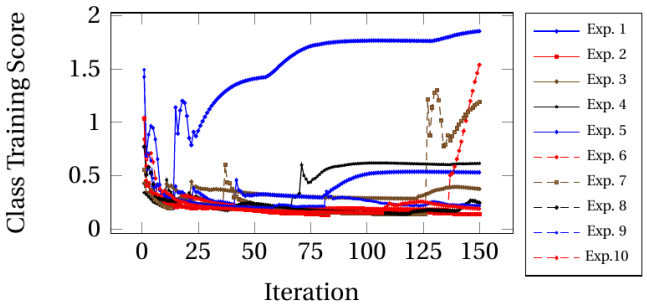
Figure 4.20 Results for the Star/Line Pattern Retrieval CNN Experiment.

CNN Experiment Results: Band Pattern

Band Patterns can be relatively well recognized with CNN classifiers. The average precision for the ten experiment repetitions was 0.85, but the recall values were averagely only 0.52. This suggests that bands can be classified/labeled with a quite reasonable certainty, but they often intervene with the Line pattern.



(a) Performance measures for Band Pattern CNN.



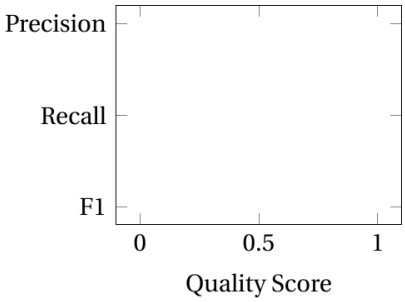
(b) Band Pattern Class Training Score Progression

Figure 4.21 Results for the Band Pattern Retrieval CNN Experiment.

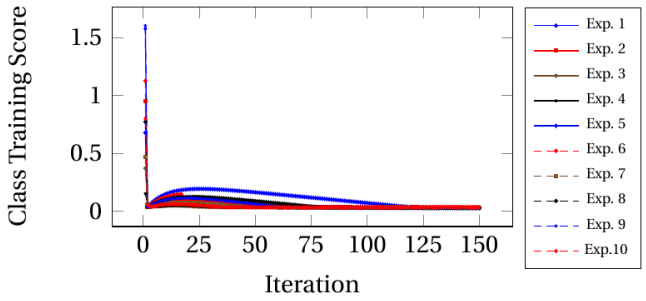
The average CNN training duration was 22:43:36 for 150 iterations. Totally, 227h 16min 02sec computing hours were used. All ten CNN experiments showed training convergence.

CNN Experiment Results: Noise (Anti-)Pattern

The noise anti-pattern CNN experiments showed consistently that CNNs are not able to reflect what the term "noise" means in matrix plots. In other words, CNNs are built to represent/reflect the dominant characteristics of a plot. Whenever these structures are highly variant or just not existent no feature can be extracted.



(a) Performance measures for Noise Anti-Pattern CNN. All the experiments retrieved 0 (zero) true positives and false positives; Thus no precision and recall can be calculated.



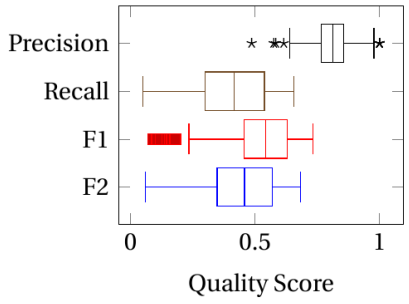
(b) Noise Anti-Pattern Class Training Score Progression

Figure 4.22 Results for the Noise Antipattern Retrieval CNN Experiment.

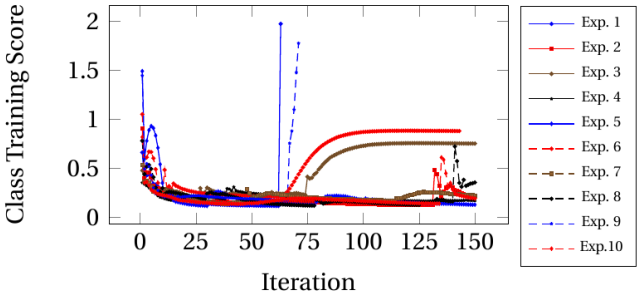
The average training duration was 22:47:52. Although all ten experiments showed consistently training convergence, none of the CNNs was able to reflect the noise anti-pattern.

CNN Experiment Results: Bandwidth (Anti-)Pattern

Similar to the Band pattern, the bandwidth pattern shows average retrieval performance. The averaged precision value is 0.82 and the average recall value is only 0.40, suggesting that the Bandwidth pattern may not be discriminated well from another pattern. A closer inspection of the results reveals that Line and Band patterns are found in the upper certainty ranks of the classifier (false positives). The average training duration for the experiments was 22h 45min 35s which is significantly faster than all other experiments (relative and absolute judgment).



(a) Performance measures for Bandwidth Anti-Pattern CNN.



(b) Bandwidth Anti-Pattern Class Training Score Progression; Experiment 1 (max. 4,305.03), 6 (max. 15.07) and 9 (max 9,147.46) show no training convergence.

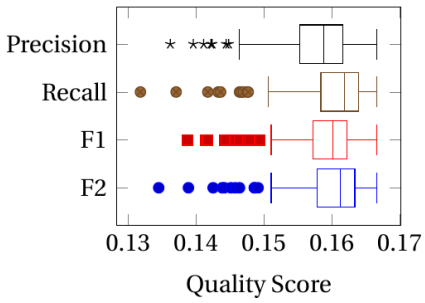
Figure 4.23 Results for the Bandwidth Antipattern Retrieval CNN Experiment.

CNN Experiment Results: Multi Pattern Classifier

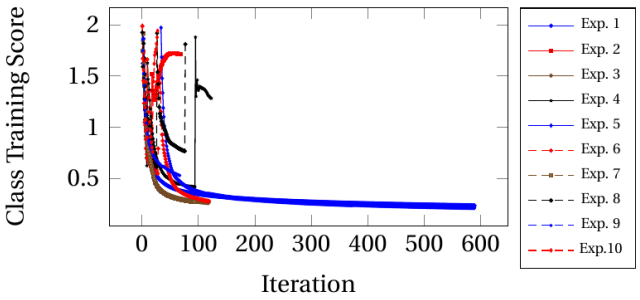
We conducted one slightly adapted CNN experiment with the purpose to classify the matrix images wrt. the patterns they contain. This refers to a multi-label classification problem were the classifier outputs not a binary "existence" decision, but rather its pattern label (block, off-diagonal block, line, etc.).

Our setup differs from the former mentioned experiments in that 589 training iterations were conducted (terminated after a longer convergence phase). In average all training passes took 47h 40min 9sec.

The retrieval results, depicted in Figure 4.24, are unexpectedly bad: Only an average precision of 0.16 and an average recall value of 0.16 were reached, thus leading to the conclusion that the CNN's expressiveness may not be sufficient/appropriate to reflect the pattern differences appropriately.



(a) Performance measures for Multi Pattern Classifier CNN.



(b) Multi Pattern Class Training Score Progression; Experiment 2 (max. 1.00E20), 3 (max. 2.98E9), 4 (max. 1,854,937.69), 6 (max. 9,413,738.738385), 7 (max. 169.47), 8 (max. 1.35E8), 9 (max. 1.69E9), 10 (max. 1.05E10) show no convergence. Only Exp. 1 and 5 showed convergence.)

Figure 4.24 Results for the Multi Pattern Retrieval CNN Experiment.

4.7 | Comparison of Pattern Analysis Approaches

After our classification experiment in the former sections we are now able to compare the learned and engineered approaches with respect to their retrieval performance and economical considerations.

Since we wanted to investigate the influence of the classifier algorithm we repeated all experiments with three distinct classifiers: (1) Random Forest (2) Naive Bayes and (3) Support Vector Machine. All implementations were taken from the Weka library [Hal+09].

Pattern	**Best CNN**	**Best FD Magnostics** Random Forest	**Best FD Magnostics** Naïve Bayes	**Best FD Magnostics** Support Vector M.
Block	0.87/0.84/0.86	0.98/0.96/0.98	0.98/0.97/0.97	**0.98/0.98/0.98**
OffDiagBlock	0.86/0.81/0.83	**0.92/0.94/0.93**	0.80/0.81/0.81	0.82/0.91/0.86
Lines	0.75/0.71/0.73	**1.0/0.99/1.0**	1.0/0.97/0.98	1.0/0.99/1.0
Band	0.88/0.70/0.78	**0.90/0.93/0.91**	0.62/0.87/0.73	0.82/0.90/0.86
Bandwidth	0.80/0.61/0.69	**0.80/0.81/0.80**	0.64/0.92/0.76	0.69/0.80/0.74
Noise	0/0/0	**0.91/0.80/0.86**	0.65/0.47/0.54	0/0/0
Time	22h 46min 31sec	686ms	**594ms**	633ms

Table 4.3 Comparison of Matrix Pattern Analysis Approaches: The table summarizes the Precision/Recall/F1 scores of the best performing CNN experiments and the best performing MAGNOSTICS feature descriptors.

Table 4.3 summarizes all results. As we can see, the Random Forest classifier shows the best overall performance in the experiments with almost always over 90% precision. Especially noteworthy is the tremendous training time difference. While the CNN approaches were trained on average several hours to come eventually to a convergence state, the other classifiers were trained under one second.

In summary, our research shows that engineered feature descriptors outperform learned feature approaches. However, other –maybe even human-influenced– classifier algorithms may improve the retrieval results.

Several other lessons learned should be mentioned here:

- CNNs show to be responsive to distinct degeneration levels.
- The CNN training batch size influences the learning performance; Our last set of experiments was therefore conducted only with one batch, i.e., all data at once, to prevent an artificially wrong data partitioning with only good or only bad examples.
- The most classical anti pattern noise cannot be detected with CNNs; CNNs are constructed to retrieve dominant features. The convolution function (in essence a form of edge detection) cannot grasp these fine structures.
- Lines and Bands seam to be too specific visual patterns for CNNs. Our hypothesis is that the high variability in these patterns cannot be reflected with the limited

expressiveness of the learned feature maps. On the other hand, if the feature map size is artificially increased the classifier tends to "overlook" coarser patterns.
- Generally, CNNs prefer more "crisp" and "edgy" patterns; Sparsity and noisiness greatly influences the performance.
- Multi-Label CNN classification showed useless results with max 20% F-Scores

4.8 | Research and Application Context

We will next show several applications of the aforementioned feature extraction techniques in various application contexts.

4.8.1 | Image-Based Pattern Analysis with MAGNOSTICS

In this section, we report on two use cases that apply the MAGNOSTICS FDs to support exploration of large matrix view data, specifically searching in a database of networks and matrices, and analyzing network changes over time. In both cases, MAGNOSTICS reduce complexity of the real data set, helping to retrieve information (matrices, or changes in a series of matrices) relevant to the respective task.

Searching in Collections of Networks

For the retrieval of similar patterns we developed a prototypical query-by-sketch interface, depicted in Figure 4.25, which can also be accessed online[4]. We will detail on the interface and its interaction possibilities in Section 5.5.

For each individually selected FD (Figure 4.25(3)) the feature vectors of the sketch image and the database images are compared (Euclidean distance) to retrieve a ranking score. The MAGNOSTICS FD set is also available: Here, a derived six-dimensional feature vector is constructed by calculating the Distance-to-Base-Pattern for every individual MAGNOSTICS FD. The query image feature vector is subsequently compared with the Euclidean distance to the image feature vectors in the database.

Figure 4.25 shows an example using the BLOCKS descriptor for the matrix reordering data set collection with 4,313 matrices [Beh+16b]. Matrices show several blocks for the drawn sketch.

Dynamic Network Analysis

Another challenge in the realm of large data collections are networks changing over time; every time step represents the network at a different time instance. Figure 4.26 shows 11 matrices from a dynamic brain connectivity network, usually comprising around 300 time

[4] http://magnostics.dbvis.de/#/sketch

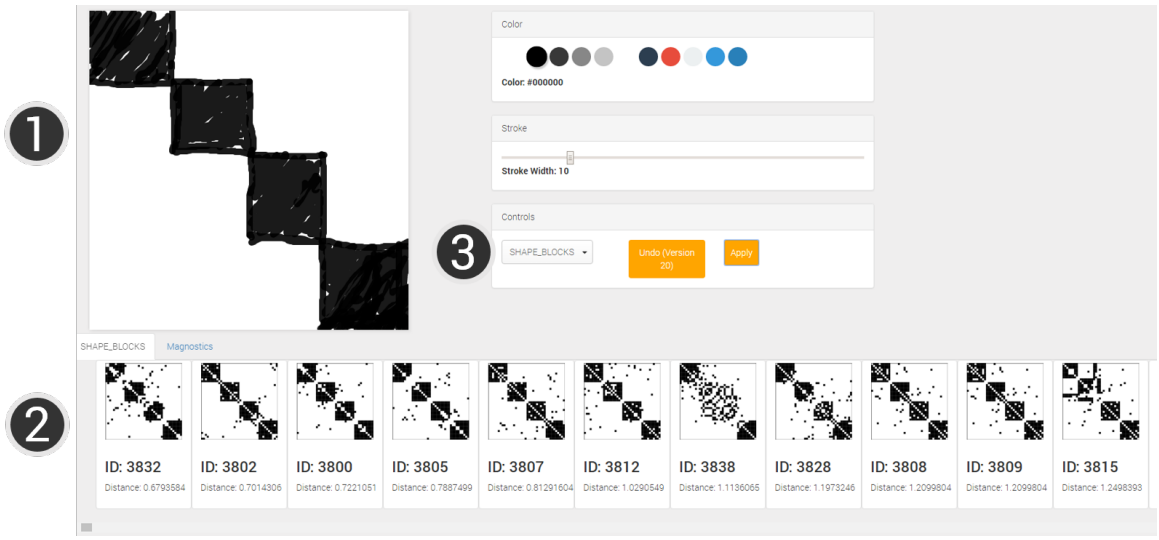


Figure 4.25 Query-By-Sketch interface for exploring large collections of matrix plots. The user can sketch in the canvas (1) an approximated matrix pattern and retrieve a ranked result list (2) according to a selected MAGNOSTICS FD (3).

points (individual matrices). Brain connectivity refers to the co-variance (connection) between activity in brain regions (regions of interest, ROI). Neuroscientists are specifically interested in clusters (Blocks) and their evolution, as well as to spot noisy time periods. Here, a time point represents 2-seconds, and dark cells indicate high connectivity (binarized to obtain un-weighted network). Rows and columns in the matrices are ordered to optimize the visual patterns *independently* for each matrix.

To find time points with major changes with respect to noise and blocks, we first calculate a feature vector v_i for every matrix (time point) t_i with HARALICK (noise) and another one with BLOCKS. Then, we calculate the difference for each feature vector v_i and the following time step v_{i+1} (separately for both HARALICK and BLOCKS). High distances indicate high changes between two consecutive time points, low distances indicate little change (with respect to noise).

Figure 4.26 shows both distances plotted (red=BLOCKS, blue=HARALICK). For noise we can observe major differences between time points 11-14, while the type of blocks changes most between time points 6-8. Both changes are observable in the respective matrices. Further, we observe, between 9 and 15, a constant change (constant distances) for the BLOCK FD (red), indicating a salient trend; a possible explanation could be the change from two separate blocks (clusters) to one larger block.

4.8.2 | Clustering of Matrix-based Representations

Cluster-based navigation systems divide the exploration space into a range of distinctive clusters, such that each grouping corresponds to a meaningful data subselection. In the

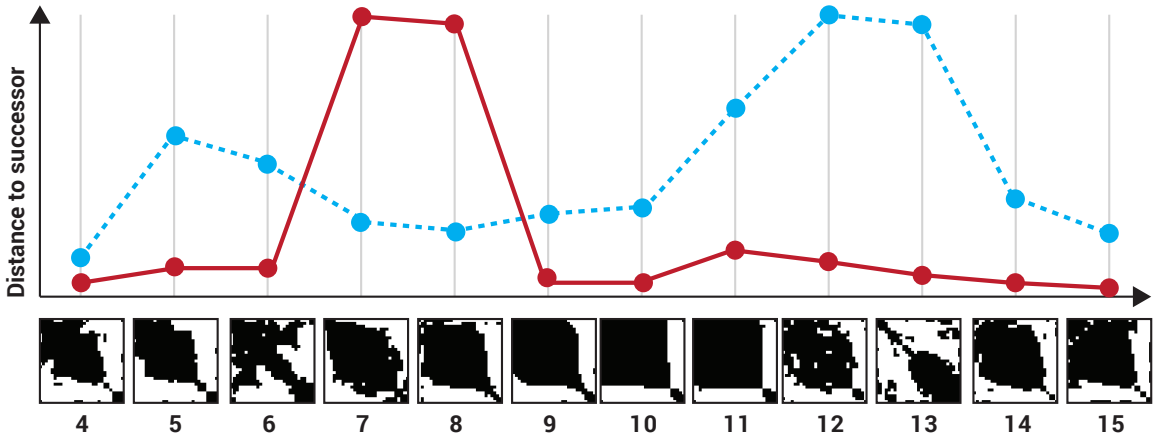


Figure 4.26 Detail from a dynamic network representing brain connectivity (300 time records).

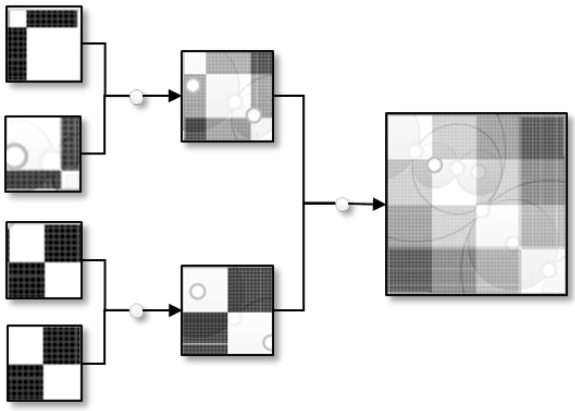


Figure 4.27 Visual depiction of a cluster protoype. All cluster entities are visually overlaid such that their relative opaqueness value aggregates to 1.

case of pattern-driven exploration, a "good" feature encoding applied on a clustering approach will result in groupings that reflect pattern-cluster memberships. However, how many clusters are meaningful and appropriate for an analysis dataset varies. Therefore, classical clustering approaches, such as k-means [Mac67], may only be applicable if the data set is supposed to be partitioned into k distinctive clusters.

In our visual exploration focus we are dealing with another problem: While the clustering itself may be useful and valid, it can often only be verified with a cluster visualization. Even more, clusters need to be represented by cluster prototypes that aggregate the information of its respective cluster. While many approaches try to give an example-based cluster prototype, such as most medoid entity (e.g., in [PK06]), other approaches try to aggregate the cluster information (e.g., in [Str+12b]). For our purposes to understand clustering results on matrix data we developed an aggregation method for matrix views as Figure 4.27 depicts.

As Figure 4.27 depicts, we construct a cluster representative from all entities contained in the cluster. We visually overlay each matrix image within one cluster such that their relative opaqueness value sums up to 1 (fully opaque). This opaqueness consideration shows intuitively the *(un-)certainty* of a hierarchical clustering with respect to the cut-level.

We developed a hierarchical clustering visualization for matrix-based representations –as depicted in Figure 4.28– as an alternative to the standard dendrogram visualization for hierarchical clustering. In this clustering visualization, we make extensive use of the cluster prototype approach to understand the clustering solutions.

As Figure 4.28 depicts we can use matrix cluster prototypes to retrieve a meaningful dendrogram split level and thus the amount of clusters in the dataset. In the example we cluster 22 manually selected matrix plots containing the block pattern with our hierarchical clustering approach; we are using texture descriptor Haralik [HSD73] for our feature encoding. As we can see in Figure 4.28b, the right split corresponds to all matrices with large off-diagonal blocks, whereas the left split contains mostly blocks along the diagonal, thus representing a good split level.

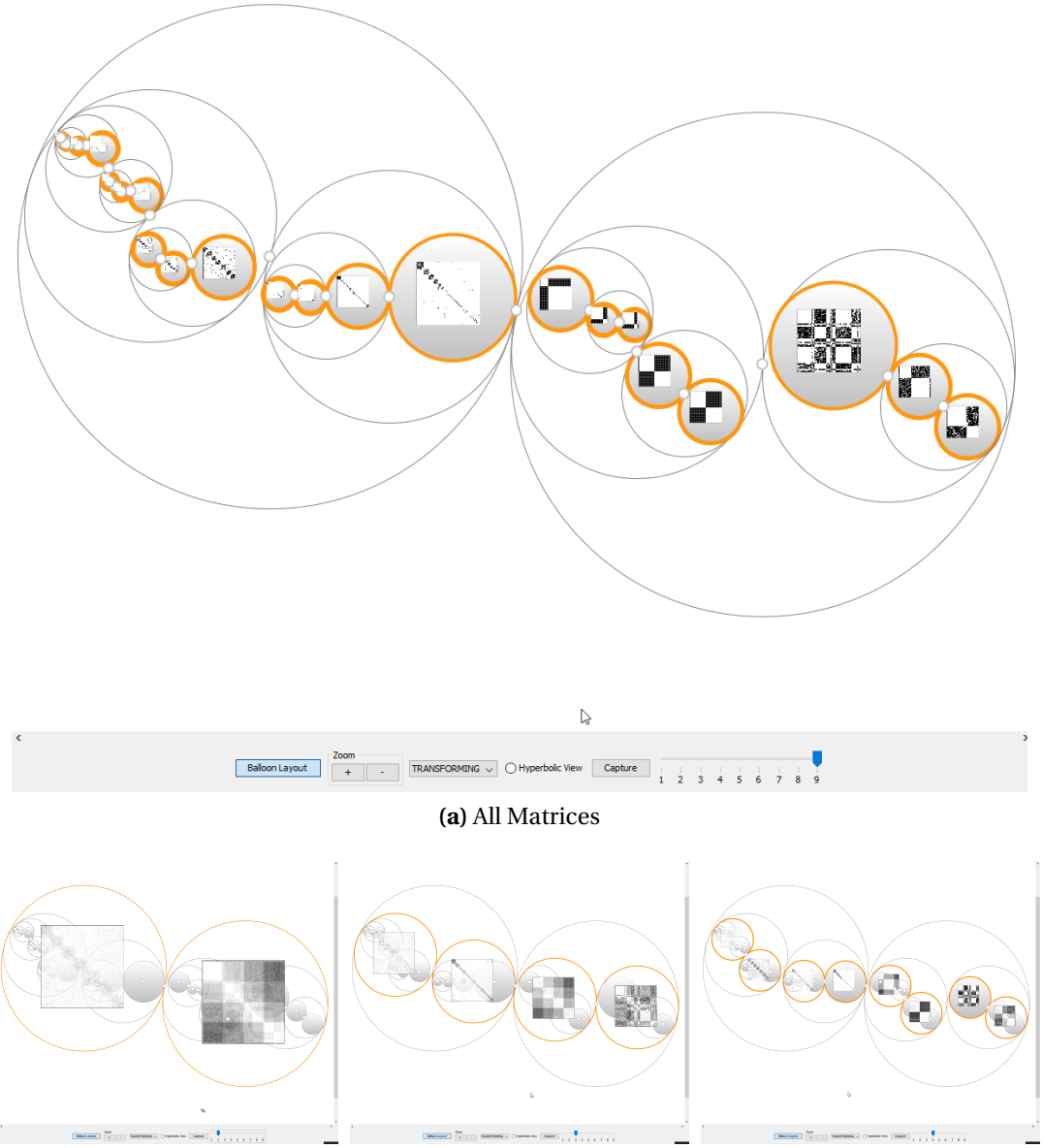
4.8.3 | Matrix Reordering for Glyph Matrices

Extracting meaningful information out of vast amounts of high-dimensional data is challenging. Prior research studies have been trying to solve these problems through either automatic data analysis or interactive visualization approaches. Our grand goal is to derive representative and generalizable quality metrics and to apply these to amplify interesting patterns as well as to mute the uninteresting noise for multidimensional visualizations.

In cases where a comparative analysis of high-dimensional data is facilitated with matrices, the matrix ordering is inherently important. However, an ordering of a matrix with glyph-representations for the high-dimensional data cells is non-trivial. During our research we made initial studies, in which we investigated a quality metrics-driven approach to achieve our goal for scatterplot matrices (SPLOMs).

We rearrange SPLOMs by sorting scatterplots based on their locally significant visual motifs, which are obtained from a discretized version of the original scatterplot distributions. Using our approach, we enable scatterplot matrices to reveal groups of visual patterns appearing adjacent to each other, helping analysts to gain a clear overview and to delve into specific areas of interest more easily.

Extracting meaningful information out of vast amounts of high-dimensional data is a challenging task. General exploration- and retrieval tasks, such as finding relevant dimensions, selecting meaningful projections, or investigating outliers, are significantly more challenging in high-dimensional data analysis. Multi-dimensional data visualization also carries its own set of challenges like, above all, the limited capability of any technique to scale to more than a couple of data dimensions. Prior research studies developed many



(a) All Matrices

(b) Cluster Level 2 **(c)** Cluster Level 3 **(d)** Cluster Level 4

Figure 4.28 Hierarchical clustering visualization for matrix visualizations. The adapted dendrogram visualization encodes the size of the respective cluster in the bubble size (dengrogram split). The orange border highlights the aggregation level for each cluster prototype, which can be interactively modified with the slider below.

visualization techniques to achieve the goal, such as parallel coordinates, scatterplots, and glyphs. However, mere visualization of all variables may introduce clutter and blurs interesting patterns in visualizations.

Researchers have been trying to solve these problems through either automatic data

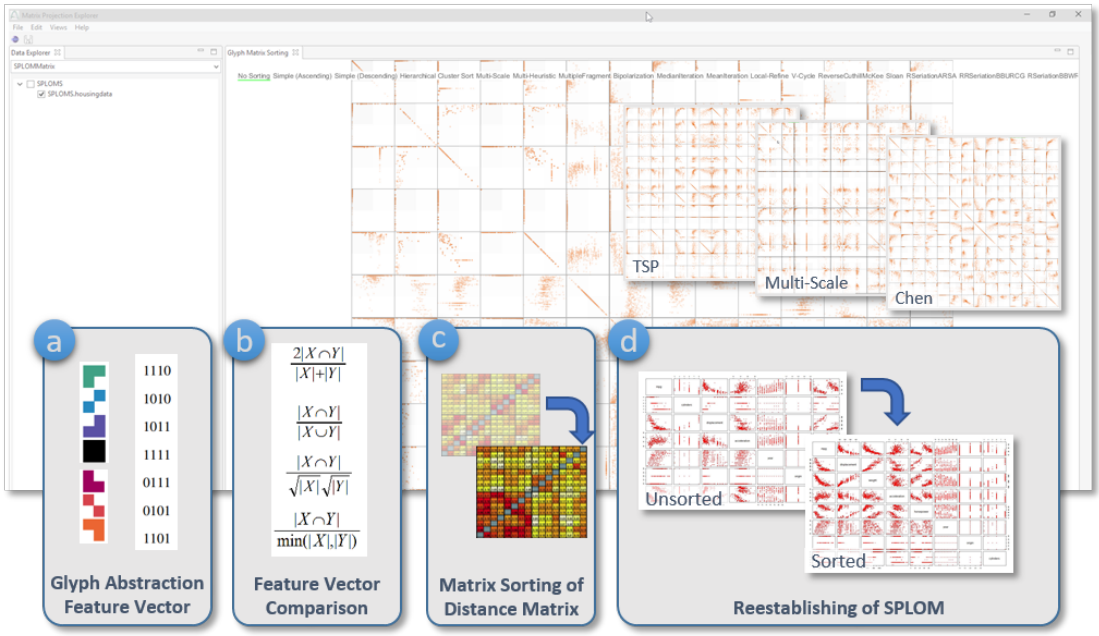


Figure 4.29 SPLOM Reordering Pipeline: Scatterplots are encoded by their visual motifs and encoded into a binary feature vector. A pair-wise comparison of all scatterplot motifs results in a distance matrix, which can be sorted with standard 2D numeric sorting algorithms (e.g., TSP- , Multi-Scale-, Chen ordering) to determine a visually coherent SPLOM ordering.

analysis or interactive visualization approaches. We propose a mixed approach, where the system – based on *quality metrics* – automatically searches through a large number of potentially interesting views, and the user interactively steers the process and explores the output through visualizations. Our grand goal is to derive quality metrics, which amplify interesting patterns and mute the uninteresting noise for multidimensional visualizations.

In past decades, many dimension management techniques have been proposed to organize layouts automatically or interactively. Ankerst et al. [ABK98] proposed to place similar dimensions close together based upon similarity metrics. In addition, a hierarchical dimension ordering, spacing, and filtering approach automatically arranges dimensions based upon dimension similarities and allows users to interactively explore them [Yan+03]. Dimension reordering can also be used to maximize the clarity of visual patterns in scatterplot matrices by reducing unnecessary clutter [PWR04a; Yan+03]. In relation to visual patterns, Dang and Wilkinson [DW14b] used Scagnostics (Scatterplot Diagnostics) to reveal hidden patterns in large collections of scatterplots. Visual cluster verification was empirically studied in [SMT13] to determine the impact of dimension reduction techniques and different scatterplot encodings (2D, 3D and SPLOMs). Interactive approaches, such as in [EDF08a], propose to navigate and rearrange multidimensional

data based upon iteratively built queries in scatterplot matrices. Despite lacking in the definition of the quality measurements, the quality-aware sorting framework for scatterplot matrices was also suggested [Alb+09b]. Inspired by aforementioned techniques, this work proposes quality metrics and an initial framework for quality metrics driven sorting for scatterplot matrices.

Quality Metrics from Visual Space

In comparison to the earlier approaches, we intend to use quality metrics derived from the visual space rather than the data space. In a SPLOM, the distribution of general patterns, called *scatterplot motifs*, are more interesting than the point distribution within one scatterplot cell. Hence, the effectiveness of a SPLOM, like many other matrix visualizations, is affected by its ordering. Thus, finding a good SPLOM ordering helps to reveal motif patterns and their distributions regardless of the dimensions under consideration.

Our approach aims to improve the visual coherence in SPLOMs by reordering the matrix, such that adjacent cells appear visually similar to each other and motifs groups form structural patterns. Furthermore, our approach is directly linked to the *input-output* model of parameter space analysis [Sed+14] in which input and output data are 2D scatterplots. To demonstrate our approach, we created several SPLOMs of the UCI housing [Lic13] dataset with different orderings, as shown in Figure 4.29. It can be seen that the different matrix sorting algorithms promote different patterns (e.g., Multi-Scale groups line motifs, while TSP and Chen group similar patterns in adjacent locations).

The Pipeline of Our Approach

Our approach to find a visually coherent SPLOM ordering is as following: 1) we calculate visual similarity between scatterplots, and 2) we compare all scaterplots using the similarity score, which determines the final SPLOM ordering. Our approach for the ordering process is depicted in Figure 4.29.

Abstraction-Based Scatterplot Feature Descriptor Inspired by the work of Yates et al. [Yat+14], we abstract the scatterplots by their contained scatterplot motifs. In case of a 2×2 grid, 16 unique motifs can be derived and encoded in a binary vector form. In this vector, a 1 represents a scatterplot segment with a point density above a user selected threshold. Using the coding scheme in [Yat+14], we form a space-filling z-curve to traverse the scatterplot segments. Users may adjust grid sizes to steer the ordering process in the feature descriptor approach.

Feature Descriptor Comparison The binary feature vector, representing a scatterplot motif, allows comparing visual appearances using overlap comparison approaches. As Figure 4.29 (b) depicts, we can calculate similarity scores based on the Dice-, Jaccard-, Cosine-, and Overlap coefficients.

Distance Matrix Sorting As Figure 4.29 (c) illustrates, a pairwise calculation of the visual distances results in a distance matrix. Every cell in this symmetric matrix corresponds to the visual similarity score of the "pivot" scatterplot to another "comparison" scatterplot. We can apply a wide range of matrix sorting algorithms to reorder the numeric distance matrix. Currently, we are experimenting with the *R package Seriation* to obtain an implementation of the matrix sorting algorithms (see also [HHB08]).

Reestablishing of the SPLOM The sorted distance matrix can be directly translated back into its ordered SPLOM correspondence or into a sorted Glyph Matrix. Therefore, we retrieve the distance matrix ordering vector and apply it to the SPLOM rows and columns. Hence, the scatterplot with the highest or lowest–depending on the matrix sorting algorithm–visual similarity to the rest of scatterplots are placed in the top-left corner of the SPLOM, as shown in Figure 4.29 (d). Other scatterplots are subsequently arranged with respect to their distance values.

5 Visual Analytics for Pattern Retrieval in Matrix-based Representations

Contents

This chapter of the thesis collects all visual analytics contributions. We will motivate our work in the light of the "interesting view" problem for large view spaces in Section 5.1 and summarize related work in Section 5.2.

In Section 5.7, Section 5.6 and Section 5.4 we shows three distinct examples of visual analytics systems. These approaches focus on (a) the user-centric and system-supported definition and learning of view interestingness, (b) the user-controlled definition and adaption of similarity functions for ranking, clustering and classification tasks and (c) a user-steerable matrix reordering approach, which allows emphasizing visual patterns of interest.

The core contribution of this chapter is to bridge the gap between the fully automatic approaches (as shown in Chapter 4) and the exploration and navigation approaches (as shown in Chapter 3). The user is able to steer the algorithmic procedures in an interactive fashion. Most of the techniques not only allow steering of an implicitly given model, but also show a model visualization, which helps to understand the algorithmic black-box.

This chapter is based on the following publications:

"Feedback-Driven Interactive Exploration of Large Multidimensional Data Supported by Visual Classifier"

Behrisch, Korkmaz, Shao, and Schreck.
Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on, IEEE CS Press, 2014, 43-52. [Beh+14a]

“Visual Analysis of Sets of Heterogeneous Matrices Using Projection-Based Distance Functions and Semantic Zoom”

Behrisch, Davey, Fischer, Thonnard, Schreck, Keim, and Kohlhammer. Computer Graphics Forum, Eurographics Conference on Visualization (EuroVis 2014), The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd., 2014, 33, 411-420. [Beh+14b]

Parts of the Motivation Section 5.1 and the Overview Section Section 5.3 are adapted and/or taken from the text/figures I have written/developed for the German Research Foundation (DFG) research proposal “Transregional Collaborative Research Center 161 Quantitative Methods for Visual Computing.”

5.1 | Motivation

Visual interactive approaches (c.f. Chapter 3) and purely automatic assessment methods of patterns (c.f. Chapter 4) are cornerstones for our central research vision of a pattern-driven exploration in matrix plots. While their combination in visual interactive systems proves already to be a powerful match, it can further be leveraged with a user-centric view: Not the system explores the dataset, but the user. Supporting the user in this iterative process can boost the efficiency of retrieval systems beyond the performance of each retrieval component. Visual analytic methods can support the exploration of patterns through steering algorithms and models towards the user's notion of interestingness.

Research Objectives: Accordingly, we are deriving several research questions related to the integration and interaction of the user, the model and the view in the visual analytics process.

1. Which interaction approaches allow expressing and gradually refining user preferences?
2. Which visual analytics methods help the user to better navigate and explore large relational datasets?
3. Can we build integrated visual analytics systems, in which an automatic decision support system (semi-)automatically supervises the user's decisions made during the exploration?
4. Which level of integration between an automatic decision support system and the user is necessary and desired?
5. How can we represent a user's –potentially complex– mental model, such that retrieval systems can take advantage of this understanding?

Patterns are seen as the gold nuggets of the data [FPS96a], but are mostly hidden in the dimensionality. One reason for this is that the number of possible representations, which might contain relevant information grows exponentially with the amount of data dimensions. This pattern discovery process is subject to an exploratory search. One classical approach, are *Overview+Detail* systems that lead the user in an overview to areas of interest and let him/her explore these areas with drill-down mechanisms [MS93]. Overview+Detail systems are an established and approved method, but often tend to be expert systems, restricted to specific data set characteristics or user interactions. Moreover, Overview+Detail systems potentially introduce misleading abstractions which ultimately can lead to wrong explorations paths.

Alternatively, exploration processes can be supported with novel querying mechanisms, such as query-by-example or sketch-based interfaces. Here, the user expresses and iteratively refines a fuzzy understanding of the patterns under investigation. On the other hand, an explicit definition of rules of interest is time consuming, particularly if these rules need to be updated every time a viewer's interests change. As a result, *implicit* and *iterative preference estimation/expression* may be better suited for cases where the manual exploration of the entire view space becomes ineffective or even infeasible. This concept is also described in [Kai+15] as "bottom-up exploration" and applied in the context of large, heterogeneous networks.

Besides navigation and comparison, visual-interactive approaches can also help users to visually specify queries for data. Sketch-based query formulation has been considered as an example-based access method in multimedia retrieval for some time now. For example, sketch-based approaches have been proposed for real-world image data [LZC11] or 3D object data [Eit+12]. Less work has addressed visual search interfaces for retrieving view data in Information Visualization. An early example is Time Searcher [HS04], which demonstrated how –based on interactive selection of time series data– users can intuitively search for similar data segments in large time series databases. Time Searcher is an important example wrt. this work, because it supported interactive search in a complex data set (time series) which was based not on a numeric or textual representation of the data, but on a visual abstraction (line chart in this case). In that way, users do not have to change the access modality (e.g., switch to a SQL prompt) but can stay within the given visual representation while exploring and analyzing the data. In [Hao+07b], a query-driven navigation approach for pixel-oriented visualization was proposed. The idea was that users interactively mark interesting sections in a pixel view, and the system retrieves other data sections for display, based on similarity to the marked data. We are transferring these concepts to matrix-based representations, where the user interactively and iteratively expresses his/her notion of view interstingness. As one example, in the system described in Section 5.5 the user draws sketches of matrix plot and retrieves –based on a feature encoding and similarity function– similar matrix plots from a database.

Overall, exploration and querying processes are getting increasingly complex and overwhelmingly powerful. Visual analytics, on the other hand, strives to support the user in the knowledge generation process (see also: [Sac+14]). One approach to overcome the complexity may be a visual analytics decision support system that learns from the user behavior to *supervise* and *monitor* the exploration process. These classification systems learn from the previous user decisions, while notifying the user in case of potentially wrong decision paths and major decision path divergence.

5.2 | Related Work

Interest-Driven Data Filtering for Visual Analysis Methods for visual data analysis need to handle increasingly large data sets. As the data size grows, so does the space of data views, which are possible, given large data spaces and view parameters. Then, analysts run risk of overlooking interesting views if relying only on interactive navigation. To this end, intelligent methods for compressing and filtering data for potential patterns of interest has recently become a research focus. Overview-based approaches aim to generate effective layouts over many candidate data portions, to efficiently spot patterns of interest. Examples include the Value-and-Relation display [Yan+07b], which lays out pixel-oriented views based on their data similarity. Another example is [WG11], where many time series are shown by small glyphs which are layed out based on data similarity.

Besides overview approaches, automatic filtering of views for potential structures of interest has been proposed. For scatter plots, the Scagnostics approach [WAG05b] automatically analyzes structures in scatter plots, which can be used to rank and filter. Recently, a clustering-based overview approach was presented in the ScagExplorer [DW14a]. In case class information is given, scatter plots can be filtered for discriminative views by class consistency measures [Sip+09b]. Also, projection pursuit approaches, such as initially presented by Friedman and Tukey [FT74], try to identify interesting 2D subspaces in high-dimensional data (mostly depicted by scatter plot views). Further heuristic interestingness filters for Scatter- and Parallel Coordinate plots have been discussed in [Tat+11b; DK10a] and may narrow down the potentially large search space for high-dimensional data. In [Tat+12b], an explorative overview of subspaces contained in high-dimensional data based on mutual differences and clustering quality properties was introduced.

Relevance-Driven Image Retrieval In Information Retrieval, similar to Information Visualization, users search for relevant information, but often without being able to precisely specify the pattern they are looking for. In context of document retrieval, relevance feedback [BR11b] allows to incrementally refine the user query. Based on a set of example documents, users assign a degree of relevance on them, based on the context of their information need. This assignment information in turn is used to iterate the search, e.g., by query term expansion or by weighting of query terms, based on the subset of relevant documents. This mechanism abstracts from the specific query formulation by the user, but may implicitly capture the user information need. Relevance feedback methods have also been intensively applied in content-based image retrieval [Rui+98a; Tao+06] and shown to improve the retrieval performance.

According to our observation, the majority of Visual Analytics approaches which incorporate interest-driven data filtering rely either on a) fixed heuristics for fully automatic filtering, or b) on fully interactive filter specification by users. However, fixed heuristics

may not necessarily map to a given users' information need, which may depend on data and context. Moreover, fully interactive search may not be feasible due to large search spaces. Surprisingly few works provide user-adaptive data filtering heuristics. In [Kei+07], intelligent visual analytics queries are proposed. The user marks a section within a given visualization as interesting; the system then computes certain distribution measures given in the data section, and automatically retrieves similar data segments from a larger database. The assumption is that the additionally retrieved data will add to the user information need. In [HD12], user data navigation is supported by a Bayes classification approach. The method learns to distinguish between interesting and uninteresting data sections while users pan and zoom an information landscape. The classifier is then utilized to suggest navigation paths of interest to a given user.

In [CBL12; Bou+13], a visual analysis system supports the exploration of multidimensional data sets in a guided fashion by means of an Interactive Evolutionary Algorithm (IEA). IEAs can generate new views and adapt to the user's interests. The scatter plot matrix view in this system depicts different projections of a data set. The system suggests new novel and potentially interesting dimensions during the exploration process. The user is able to (re-)define interestingnes scores for the new views, which are subsequently used to improve the retrieval performance in the next generation of the evolutionary algorithm.

Two further recent works exploit user interaction to improve the analysis process. In [Bro+12], users interact by with the marks in a 2D projection of high-dimensional data, to express their notion of data correspondences. This input is used to adapt the data similarity function and re-project the data. Along similar lines, the approach in [EFN12] allows users to interact with the positioning of documents in a 2D document landscape collection, to express document-level relationships. The system then learns and highlights the most descriptive document terms from the expressed document relationships.

Sketch-Based Methods for Exploration[1] Sketch-based approaches allow matching a user-provided sketch against image content [Eit+10]. Hence, these approaches are another possibly to explore large view spaces, because the user can iteratively refine the query by modifying/exchanging the sketch (query-by-example). Generally, content-based image retrieval systems have the goal to design functions to compare and rank images for similarity of content. Typically, various low-level image features including color histograms, edge histograms, or texture measures can be used [DKN08].

Nowadays, most sketch-based retrieval approaches consider real-world images. On the other hand, these systems can also be a viable approach to support data retrieval and in particular, data understanding and comparison. Sketch-based approaches can be applied to general data sets, provided adequate visual representations exist based on

[1] This paragraph has been taken over from the co-authored paper "Guided Sketching for Visual Search and Exploration in Large Scatter Plot Spaces" [Sha+14]

which queries can be formulated. In previous work sketch-based search in graph [Lan+10] and time series [Ber+11a] data were considered.

A problem in content-based search is often how to define a query if no example search object is available. Therefore, assisting techniques have been researched. Shadowdrawing was originally introduced to help untrained users execute appropriate sketches [LZC11]. Previous works have considered specific search methods for navigating in visualization spaces. To search for interesting local patterns in time series data, a sliding-window approach together with interactive query selection was introduced in [HS04]. In previous work, we considered search systems for time series data [Ber+10] and graph data [Lan+10].

User-Guided Similarity Definition for Visual Analytics Systems[2] Data sets containing categorical and numerical data attributes (mixed data) occur in practically all application domains. Here, typical analytical tasks like searching for nearest neighbors, grouping similar objects, detecting outliers or recognizing other interesting patterns can only be conducted if a similarity function is provided. Such a function computes distance scores between multivariate data objects, respectively. However, individual objects of these mixed data sets are hard to compare. This is especially the case if the user's mental similarity notion (short: MSN) is subject to change over time, as typically seen in sense-making loops. In these cases a user-guided similarity definition allows applying and adapt the MSN of domain experts interactively, at run-time.

As an example, the user arranges the objects Berlin, Paris, and Washington close to each other on a 2D landscape, and the system learns that the MSN of the user concerns the categorical attribute 'Capital City'. Another, more academic example might be a 2D landscape of patient data, where a doctor arranges patients close to each other who all had a typical sort of behavior (based on the MSN of the expert). The beneficial means of such a system would be that attributes would protrude for an early detection/diagnosis, possibly undiscovered as such so far. This utilization of user-defined object arrangement is already applied for numerical attributes in inspiring works of Liu et al. [Bro+12] and Mamani et al. [Mam+13]. However, the development of feedback models which also cope with mixed data remains challenging.

In a previous work, Bernard et al. presented a concept for the development of systems for user-defined similarity definitions for mixed data objects [Ber+14b]. They divided the development process of such systems in 15 detailed steps where mandatory design choices exist. One of the most crucial steps thereof regards the algorithmic mapping of the 2D object arrangement to a similarity function. One approach for this algorithmic mapping is the calculation of attribute weights, depending on the correlation of attributes to the

[2] This paragraph has been adapted from my co-authored paper "Towards a user-defined visual-interactive definition of similarity functions for mixed data" [Ber+14a]

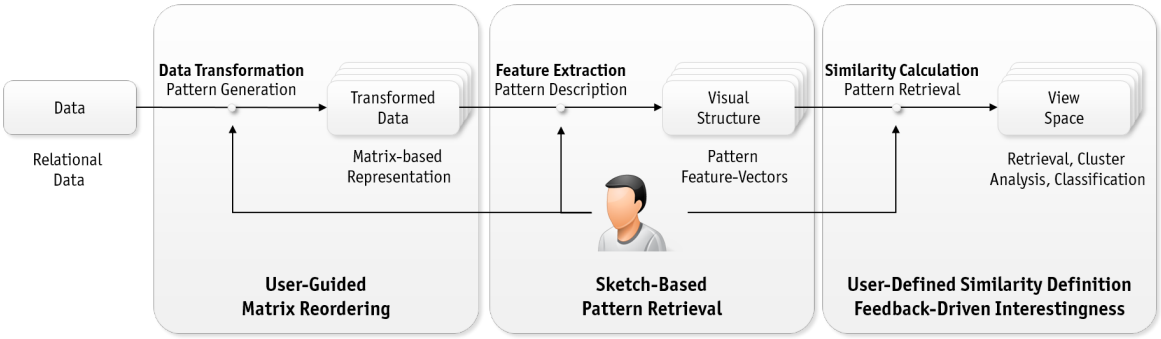


Figure 5.1 Visual Analytics Approaches for Exploring and Navigating in Large Amounts of Relational Data. In Section 5.4 a user-guided matrix reordering is presented in which the user may express his/her preference for a specific pattern. In Section 5.5 a sketch-based pattern retrieval interface is presented, which helps the user to retrieve specific matrix patterns. Third, Section 5.6 shows a user-defined similarity definition approach and Section 5.7 focuses on the retrieval of interesting views from large matrix spaces.

user-defined object arrangement, which can subsequently be utilized for the creation of similarity functions for mixed data.

However, on top of that feedback models can generate weightings for categorical and numerical attributes based on user-defined feedback objects. For this purpose, we will showcase the results of our initial visual analytics inspired similarity definition and adaption in Section 5.6.3.

5.3 | Overview

User adaption is the key topic for Visual Analytics (VA) applications. As similarity notions for complex data are many and depend on the context, user adaption has been explored as a way to (semi-)automatically adjust data transformations, similarity functions or data views to user information needs.

Feedback-driven approaches –as one instance of a user-adaptive system– rely on users denoting their interest, and the system then learning how to classify relevant against non-relevant data sections. In Information Retrieval, relevance feedback techniques provide for this adaptation [BR10; Rui+98b]. Less work has considered adapting to user context during the Visual Analysis process. In Section 5.7 we present a relevance feedback approach to support user navigation to interesting sections of a large view space, based on user markup of interesting views. Moreover, we adapt the concept of feedback-driven "bottom-up" pattern space exploration in matrix plots and present how visual classifiers and can help to explore the view space.

Also related to the research question, how interactions allow expressing and gradually refining user preferences and in line with the fundamental goal of Visual Analytics to

increase the transparency of so-called *black box* algorithms, we developed an iterative user-guided matrix reordering approach. Our approach, as presented in Section 5.4, considers matrix patterns as local areas-of-interest, which may be interactively composed to highlight a matrix plot's information. For this purpose, we modularize the reordering process by enabling users to select groups of similar rows (or columns) and to apply local sorting algorithms to those rows. In this way, users can apply their knowledge to locally optimize the results of global reordering algorithms. As a side-result, the task of matrix reordering becomes more understandable for the user. Our technique centers on a similarity-preserving, two-dimensional projection of the matrix which provides the user with a simple visual interface to interact with the vertex ordering.

As already noted in Section 4.3, the notion of similarity is important concept for a pattern-driven exploration system. In VA applications this notion of similarity may be interpreted as non-static, thus allowing the user to adapt the similarity calculation and/or the feature encoding on which a similarity score is calculated. In Section 5.6 and Section 5.7 we present our visual analytics approaches for this topic: In Section 5.6, we show how the user can interactively and visually steer a distance calculation process with the aim to improve ranking and clustering results. And in Section 5.7 we show how a (semi-)automatic decision support system may come to the conclusion that a feature descriptor change may be beneficial for the exploration task at hand.

Novel query interfaces for pattern retrieval in matrix-based representations are another focal point of this chapter. In line with the research objective to build visual analysis systems that allow users to represent their complex mental model, we developed a sketch-based search interface for matrix patterns. We will present this approach in Section 5.5. This project makes use of the feature descriptors and classification approaches presented in Section 4.4 and combines them in a visual analysis framework.

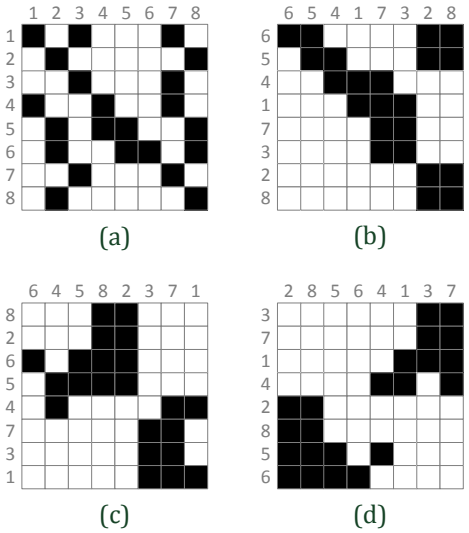
5.4 | User-Steerable Iterative Matrix Reordering.

A fundamental goal of Visual Analytics is to increase the transparency of so-called *black box* algorithms. We present in the following our work on an existing and established class of black-box algorithms dealing with the task of reordering symmetric matrices. Fundamentally, our work modularizes the reordering process by enabling users to select groups of similar rows (or columns) and to apply local sorting algorithms to those rows. In this way, users can apply their knowledge to (locally) optimize the results of global reordering algorithms. The task of matrix reordering also becomes more understandable for the user. Our technique centres on a similarity-preserving, two-dimensional projection of the matrix which provides the user with a simple visual interface. In the following, we describe an implementation of our technique as a software system for the exploration

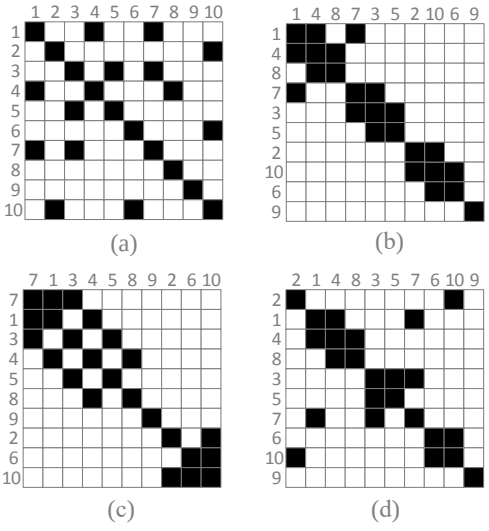
of symmetric matrices. We also describe a case study focusing on similarity matrices in clustering tasks to provide an initial validation of the technique.

As already detailed in Section 2.3 matrix reordering (or seriation) involves the permutation of the rows and/or columns of matrices to identify useful structures. Useful structures are patterns, which deliver previously unknown and exploitable information about the data set in question. However, for a matrix with n rows and m columns there are $n! \times m!$ possible permutations. Of this large number of permutations it can be assumed that very few actually deliver useful information.

To make matters worse, different permutations of the same matrix may exist, which reveal very different aspects of the data in question. A simple example of this dilemma is presented in [Lii10] and shown in Figure 5.2. Here, the same matrix is reordered once by hand and then with the help of two reordering algorithms. The result of each reordering reveals clear structures. They are, however, strikingly different.



(a) A asymmetric matrix and three reordering results adapted from [Lii10] (a) The raw data in its original order, (b) A hand-generated reordering, (c) and (d) Reordering results of two different algorithms.



(b) A symmetric matrix and tree reordering results: (a) The raw data ordered randomly (b -d) Reordering results of three different algorithms revealing different patterns.

Figure 5.2 Matrix Reordering Dilemma: Several matrix reordering algorithms can be applied to reveal potentially different visual patterns.

Typical matrices in modern real-world applications are large enough to make user-reordering prohibitively laborious. Users must resort to automated methods, which (as shown in Figure 5.2) may produce very different results. Most matrix-reordering

algorithms, such as the ones presented in [Lii10; Beh+16b], solve an optimization problem based on predefined local or global target criteria. They are more-or-less black-box algorithms; the user has no control over results beyond the choice and parametrization of quality criteria. Due to the large search space, the algorithms use heuristics and may return a local optimum in certain circumstances. In addition, their complexity is such that multiple runs with different parameterizations can be very time consuming. We feel that the introduction of interactive visualizations in the analysis process will be very helpful to improve results and user satisfaction.

In this section we present a visual analytics system, which enables the user to steer the sorting process in a more meaningful way. We allow users to *intervene* in the reordering process by manually modifying the positions of single rows and/or columns in the sequence. We also enable the selection of subsets of columns and/or rows for the local application of reordering algorithms. The user is supported in these tasks with intuitive visualizations, which clearly reveal the effects of the algorithms which were applied. The result is a more *task-specific* and data-subset specific ordering , which -according to our results- may provide more useful information than black-box methods.

For the sake of simplicity, we restrict ourselves to the case of symmetric matrices. A matrix M is said to be symmetric if $M = M^T$, i.e., the matrix is equal to its transposition. Every symmetric matrix can be interpreted as the adjacency or edge-weight matrix of an undirected graph. Conversely, given an undirected graph with or without edge weights, a symmetric matrix can be induced.

5.4.1 | Iterative User-Guided Matrix Reordering Pipeline

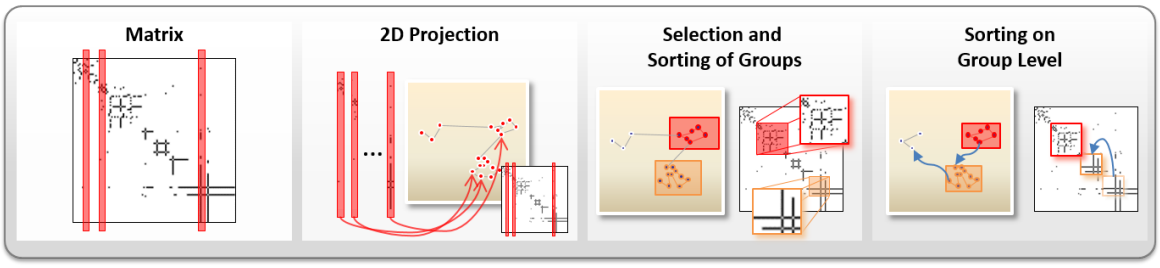


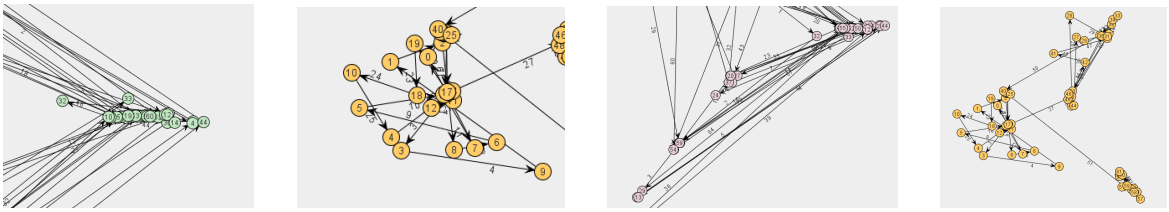
Figure 5.3 Processing pipeline for our user-steerable matrix reordering approach: the columns and/or rows of a matrix are interpreted as high-dimensional vectors (1^{st} image) and projected to two-dimensional space (2^{nd} image) forming a set of vertices. Similar high-dimensional vectors are projected to similar 2D positions. The matrix ordering (e.g. resulting from a matrix sorting algorithm) is visualized by an edge path connecting all vertices. Selecting groups of vertices allows the local application of sorting algorithms on submatrices (3^{rd} image). The edge path can be manually modified, such that locally optimized submatrices or single vertices can be placed next to another (4^{th} image).

Figure 5.3 provides an overview of our approach to user-guided matrix reordering. In the following we illustrate how the steerable reordering process can increase user understanding of the matrix representation, as well as the measurable quality of the ordering. In this work, we consider the matrix ordering quality in terms of the *linear arrangement*, which indicates numerically how well a block-ordering along the matrix diagonal can be achieved.

Our user-guided matrix reordering approach relies on a projection of the matrix into the plane, where the user can interact with the matrix elements by moving or grouping them. We realize the projection by regarding the matrix rows or columns as a set of high-dimensional vectors. For symmetric matrices (representing undirected graphs), rows and columns can be used interchangeably. For non-symmetric matrices it is left to the user to pick one of the two. A topology-preserving projection function is applied to map the row or column vectors to vertices in a 2-dimensional display that retains the main similarity (neighborhood) relationships between the rows or columns of the matrix (cf. Section 5.4.3). The vertices in this display are connected by directed edges representing a given ordering of the rows or columns. Hence, a path with $n-1$ edges visualizes the current matrix sorting. The start and end points of this path map to the first and last row or column in the matrix representation.

The 2D projection provides the user access to interactive matrix sorting operations. The projection view enables the user to perceive a basic ordering in terms of the similarity of rows or columns and reason about how these could be arranged in a structurally meaningful way (cf. Section 5.4.2). Once such structures are found, the user operates two key interactions: (1) Invoke an *automated local reordering* on a selected group of vertices, and (2) *Rearrange* vertices or groups by replacing one or more existing edges. In the case of automated local reordering, the user selects a group of vertices and then invokes the matrix reordering algorithm of choice. The algorithm operates only on the selected matrix elements, effectively sorting a subregion of the selected matrix. This sorting approach can work recursively, as user-selected groups can be organized into subgroups and each sorted accordingly (cf. Section 5.4.3 for details). The second key interaction mechanism is the direct modification of the edge set in the projection. We enable users to interactively reconnect edges in the projected matrix representation. This is primarily used to modify the global order of locally ordered groups of matrix elements.

The starting point for interaction is the selection of a matrix ordering to be improved. Figure 5.3 illustrates a user choosing a group of matrix elements (shown in red) and invoking a local reordering. The first group is then joined to a second group of matrix elements (shown in orange) to place them adjacent to one another in the matrix. The result of these interactions can be seen as a *concatenation* of multiple (locally applied) reordering algorithms.



(a) Pattern 1: Vertex groups

(b) Pattern 2: Star edges

(c) Pattern 3: Multiple "parallel" edges

(d) Pattern 4: Combination of multiple patterns

Figure 5.4 Several visual patterns become apparent in the projection space. Recognizing these patterns is beneficial for the improvement of the matrix reordering. They visually represent a mismatch between close projection points – i.e. similar column-/row vectors of the matrix – and long connection edges – i.e. to the sequential placement of dissimilar column-/row vectors by the reordering algorithm.

5.4.2 | Matrix Patterns in the Projection Space

Reordering a matrix in the projected space involves a set of patterns. We structure the interaction based on the visual patterns *vertex groups*, *star edge patterns*, *long parallel edges* and *pattern combinations*. These patterns prove to be a beneficial starting point for improving the matrix ordering. They visually represent a mismatch between close projection points – i.e. similar column-/row vectors of the matrix– and long connection edges – i.e. the sequential placement of dissimilar column-/row vectors by the reordering algorithm.

Vertex Groups Vertex Groups are a natural starting point for the improvement of matrix sorting results. In case of an initial bad group, as depicted in Figure 5.4a, a locally dense region is highly connected to vertices outside of the group. This corresponds to a matrix ordering, in which dissimilar vectors are arranged sequentially in order to improve the (global) optimization function.

Star Edges The star edge pattern is a special case of the vertex group pattern. It represents another important pattern for a local optimization. Figure 5.4b depicts this pattern in a region of the projection view. Here, a dense group of vertices is connected to multiple satellite vertices. An optimization of this pattern would be to connect all satellite vertices linearly with one edge to the dense central vertex group.

Multiple Parallel Edges Multiple parallel edges are a further important pattern for local optimization. Figure 5.4c shows one instance of this visual pattern. In this case, similar vertices are not connected optimally to their neighborhood. This pattern occurs when similar vertices were not arranged correctly into "local" subsequences.

Pattern Combinations Usually, a combination of the aforementioned patterns is found. Figure 5.4d shows one example, of the pattern "vertex group", the pattern "star edges" and "multiple parallel edges".

5.4.3 | Interaction with the Matrix in Projected Space

After sketching our idea and the interaction with matrices in the projection space, we now describe the details of our approach.

Projection of Matrix Data Our approach relies on a projection technique to map matrix rows or columns to 2D space for sorting interaction. Kosinov and Caelli considered the task of comparing two graphs with an inexact graph matching approach in [KC02; CK04]. They showed that appropriately projected matrix vectors will reflect the similarity relationships present in the high-dimensional matrix data *independent* of the matrix's initial ordering. The researchers proposed the use of *eigendecomposition-based projection techniques*, because the eigenvalue spectrum of a matrix is invariant with respect to similarity transformations. That is for any non-singular matrix $\mathbb{M}$, the product matrix $\mathbb{P}\mathbb{M}\mathbb{P}^{-1}$ has the same eigenvalues as $\mathbb{M}$. Specifically, this means that the derived spectrum of a graph represented by its adjacency matrix is not affected by row/column permutations $\mathbb{P}$ [KC02]. This insight builds the basis of our approach, since it allows the perception and manipulation of a matrix's ordering without having to recompute its projection points.

The task of a projection technique is to reduce the size of a dataset by reducing the number of dimensions, while minimizing the information loss. In the case of linear projection techniques, a value is determined for each of the high-dimensional components that represents the dimension's weight in the low-dimensional projected space. The projection vector is then a linear combination of the original dimensions. In essence, linear projection techniques attempt to separate important from unimportant dimensions to form the projection space. While linear projection techniques are able to preserve linear structures, non-linear projection techniques also take non-linear structures into account. Examples of non-linear structures include arbitrarily shaped clusters or curved shapes. These techniques try to preserve the high-dimensional *neighborhood* properties, such as pairwise distances, in the low-dimensional space. Consequently, non-linear projection techniques seek to separate projection points whenever their high-dimensional counterparts are far apart.

In our implementation we include an exemplary linear and non-linear projection technique. Specifically, *principle component analysis (PCA)* [Jol86], and *classical scaling (metric multidimensional scaling)* [CC00b]. We used the implementations of PCA and Classical Scaling from the "Projection Explorer Framework" of Paulovich [POM07].

Local Group Ordering A local group ordering corresponds to a reordering of a submatrix either with the help of an automated sorting algorithm or a user-steered manual reordering decision. It can only be applied in the matrix representation if all the row-/column vectors in the group are arranged sequentially. If this is not the case then the linear ordering of the matrix will not be consistent after a row-/ column reordering operation. To establish the selection of a group of vertices we implemented the `PartSort` algorithm shown in Algorithm 8. Based on a user selection, it arranges the corresponding matrix elements sequentially *and* ensures that the corresponding 2D projection vertices are adjacent to one another (i.e. connected by edges).

Algorithm 8 The PartSort algorithm prepares the matrix for a local reordering. It arranges the selected vertices sequentially, beginning with the first selection id.

```
 1: procedure PARTSORT ALGORITHM
Require: Original ordering of DataVectorList
 2:     while !SelectionIdList.isEmpty() do
 3:         Id1 := SelectionIdList.getFirstID()
 4:         Id2 := SelectionIdList.getSecondID()
 5:         if !DataVectorList.isNextTo(Id1, Id2) then
 6:             DataVectorList.insertNextTo(Id1, Id2)
 7:         end if
 8:         SelectionIdList.remove(Id1)
 9:     end while
        return Ordered DataVectors with sequentially connected SelectionIDs
10: end procedure
```

The `PartSort` algorithm prepares the matrix for a local reordering. After this initial grouping, all available matrix sorting algorithms can be applied on the selected submatrix as if the matrix stands alone. The last step in the local reordering of submatrices is the reassembling of the matrix. Figure 5.5 shows our matrix reassembling schema.

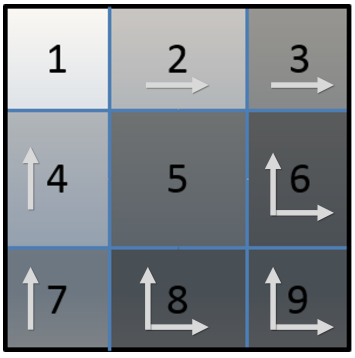


Figure 5.5 Matrix reassembling schema after local reordering. The arrows indicate the necessary re-adaption type that has to be applied.

As noted above, a 2D space vertex selection corresponds to a submatrix selection in the matrix image space. For our example case shown in Figure 5.5, the user selection

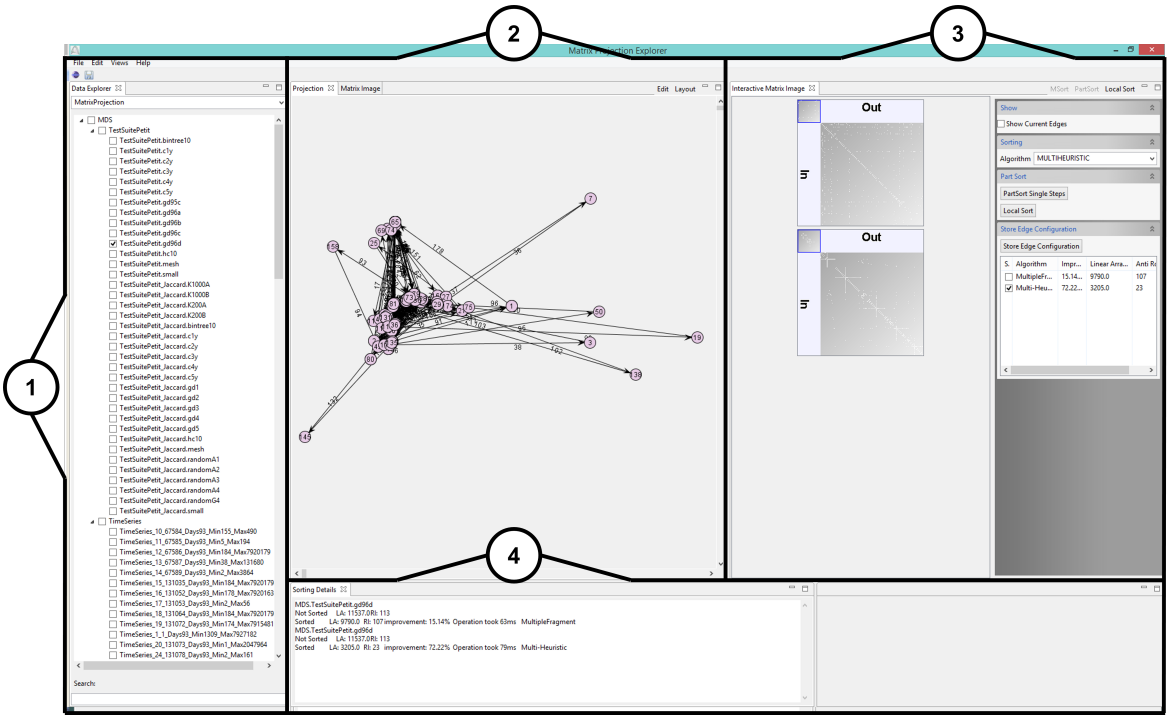


Figure 5.6 The Sorting Interaction Framework allows the assessment of the performance of matrix sorting algorithms and enables users to steer the sorting process interactively. The framework is used to visualize matrices and their projections. Here, a 180×180 matrix is rendered using two ordering algorithms. The performance of the two algorithms varies highly in terms of their linear arrangement (sorting quality criterion). The matrix's projection space allows to the visual analysis of sorting improvement potential. A set of simple interactions in the projection space lets user steer and understand the automatic sorting algorithms.

corresponds to the submatrix 5. After reordering this submatrix, the horizontal and vertical axis reordering differences have to be applied to all other submatrices, too.

While the submatrix 1 remains untouched by the local sorting (no change of the row/-column ordering), all other submatrices have to be adapted. Depending on a submatrix's position relative to submatrix 5 either a horizontal (column-wise) or vertical (row-wise) or horizontal and vertical adaptation has to be applied. For example, submatrices 2 and 3 will only be affected in the horizontal direction. The same applies for submatrices 4 and 7 in vertical direction. Therefore, a column-/row swapping helps to reassemble the matrices correctly. Adaptations to submatrices 6, 8 and 9 must be applied in both the horizontal and vertical directions.

Reordering Strategies A formally correct linear arrangement is present in the projection view if the graph is connected and there exists exactly one acyclic path that connects all vertices. Accordingly, every user intervention/manual adaption of the path inevitably

requires reestablishing this formal requirement. As an example, edge insertions introduce cycles, which make the linear arrangement ambiguous by introducing two potential paths. Generally, we can categorize edge set manipulations into two modification types:

1. **Forward Edge Modifications:** Forward edges are those edges whose linear arrangement start index (index of the inserted edge's outgoing vertex) is smaller than the end index (index of the inserted edge's incoming vertex). For example, if the user wants to place the 1^{st} row-/column vector next to the 5^{th} row-/column vector.
2. **Backward Edge Modifications:** Backward edges are those edges whose start index is larger than the end index. For example, if the user wants to place the 5^{th} row-/column vector next to the 1^{st} row-/column vector.

For both edge modification types different reordering strategies have to be applied, which ultimately have a significant impact on the matrix's overall sorting quality with respect to the linear arrangement. In general, reestablishing a correct linear arrangement is achieved by inserting so called *"reconstruction edges"* to build one acyclic path that connects all vertices.

As a proof of concept and a point of further research we implemented two different reordering strategies, which will be described in the following.

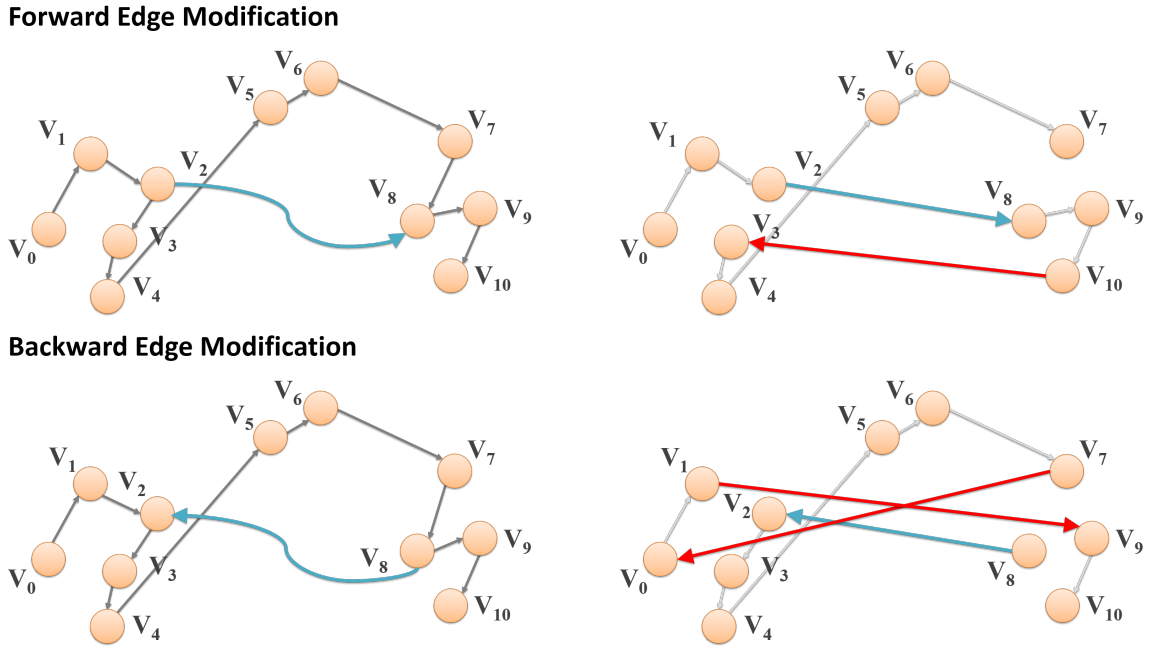


Figure 5.7 Two matrix reordering strategies are possible: A forward edge modification (upper diagram) and a backward edge modification (lower diagram). Depending on the type of reordering strategy, different approaches to reestablish a formally correct linear arrangement have to be applied.

As the upper part of Figure 5.7 depicts, one implementation of the *forward edge modification algorithm* is to split the vertex set into three disjunct subsets. The first subset contains all vertices with an index before the insertion edge's starting point (V_2) in the $Set_1 = [V_0, V_1]$. The second subset contains all edges with an index larger than the starting point and smaller than the endpoint (V_8) in the $Set_2 = [V_3, V_4, V_5, V_6, V_7]$. Finally, the third subset contains all edges with an index after the endpoint in the $Set_3 = [V_9, V_{10}]$.
The new vertex sequence is calculated as follows: First, we start the linear arrangement by retaining the vertices from the $Set_1 = [V_0, V_1]$ and then add insertion edge's start point and end point $[V_0, V_1, \mathbf{V_2}, \mathbf{V_8}]$. Then Set_3 is concatenated $[V_0, V_1, V_2, V_8, \mathbf{V_9}, \mathbf{V_{10}}]$. Finally, a new *reconstruction edge* is added to append all items in Set_2, leading to the final linear arrangement of $[V_0, V_1, V_2, V_8, V_9, V_{10}, \mathbf{V_3}, \mathbf{V_4}, \mathbf{V_5}, \mathbf{V_6}, \mathbf{V_7}]$.

As the lower part in Figure 5.7 depicts the *backward edge modification algorithm* is slightly more complicated. We also split the vertex set into three subsets. In this case Set_1 contains all vertices after the insertion edge's endpoint $Set_1 = [V_3, V_4, V_5, V_6, V_7]$. In the second set Set_2 we collect the ordering from the initial linear ordering start until the insertion edge's endpoint $Set_2 = [V_0, V_1]$. Finally, we store all vertices after the insertion edge's start point in $Set_3 = [V_9, V_{10}]$. The new linear arrangement starts with the insertion edge sequence $[\mathbf{V_8}, \mathbf{V_2}]$. Then Set_1 is attached $[V_8, V_2, \mathbf{V_3}, \mathbf{V_4}, \mathbf{V_5}, \mathbf{V_6}, \mathbf{V_7}]$ and a reconstruction edge needs to be appended to connect to Set_2; $[V_8, V_2, V_3, V_4, V_5, V_6, V_7, \mathbf{V_0}, \mathbf{V_1}]$. Finally, another reconstruction edge connects to the missing vertices in Set_3 and finalizes the linear arrangement to $[V_8, V_2, V_3, V_4, V_5, V_6, V_7, V_0, V_1, \mathbf{V_9}, \mathbf{V_{10}}]$.

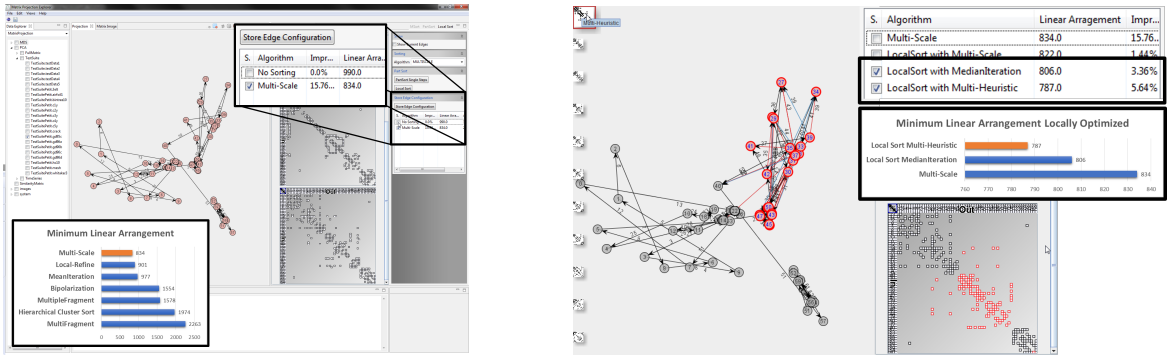
A wide range of other implementations for the edge modifications are possible. While the algorithms presented above are just one heuristic they perform well in establishing a better linear arrangement in our experiments. However, more sophisticated algorithms that minimize the reconstruction edges' length while retaining most of the input linear ordering are imaginable. These algorithms need to judge whether the user should be overruled for the sake of a better linear arrangement. Since these research questions relate to uncertainty (visualizations) and fuzzy decisions they were not investigated in this paper.

5.4.4 | Visual Components of the Sorting Interaction Framework

Figure 5.6 showcases all visual components that were implemented for our user-steerable, interactive reordering approach.

Data Explorer View This view allows the selection of a matrix for investigation (Subpart (1) in Figure 5.6).

Projection View This is the main interaction component in the framework. It allows the visualization of the chosen matrix's two-dimensional projection. Furthermore, the linear



(a) Matrix sorting with the multi-scale algorithm leads to a linear arrangement score of 834.

(b) A local reordering of a selected vertex group with the multi-heuristic algorithm leads to an improvement of 5.64% (LA of 787).

Figure 5.8 The user can steer the reordering process by invoking a localized reordering algorithm. Ordering thumbnails on the left side in (b) allow the anticipation of localized reordering results without applying the transformation to the data. Here, the user selection leads to an improvement of the linear arrangement quality measure.

ordering is represented by a path of sequentially ordered vertices. Two primary interaction mechanisms are available for the user: (1) a grouping/selection of vertices, and (2) an edge modification, which allows the user to draw new edges into the linear arrangement (Subpart (2) in Figure 5.6).

Interactive Matrix Visualization View This view allows the user to render a matrix visualization in a traditional heat-map style display. Its primary purpose is to keep track of the selections (matrix cell outlines), invoke the *PartSort* and local sorting algorithms on user selections or store the current edge configuration. A visual history of all interactions shows the progress (Subpart (3) in Figure 5.6).

Sorting Detail View This helper view shows a textual representation of the invoked sorting manipulations. Specifically, the linear arrangement score and anti-robinson events are compared before and after the manipulation, leading to a relative improvement score in percentage (Subpart (4) in Figure 5.6).

Thumbnails Thumbnails help users to anticipate which local sorting decision will improve the linear arrangement. For the currently selected submatrix, they forecast the outcome of each of the possible matrix sorting algorithms in a matrix thumbnail (overlaid in subpart (2) in Figure 5.6). The thumbnails are interactively overlaid in the projection view and appear after PartSort or a local reordering has been invoked. Their placement on the screen reflects the anticipated submatrix's sorting quality (linear arrangement score). From top-left (best linear arrangement) to bottom-right (worst linear arrangement) the

thumbnails are arranged around a virtual bounding box of the user selection. This functionality acts as a decision support system, where users can focus on their selection process, while being able to see the potential outcome of the operation.

Framework Architecture To apply our approach and conduct our experiments we developed the *Sorting Interaction* prototype. Figure 5.6 shows the front end. Essentially, the framework consists of two main parts:

1. The back end; a NoSQL database (MongoDB), to store a set a large number of matrices projected onto the two-dimensional plane with the MDS (Classical Scaling) and PCA projection techniques. Its primary purpose is to function as a cache for the time consuming projection calculation.
2. The front end, as depicted in Figure 5.6 and described in Section 5.4.4.

5.4.5 | Workflow and Interaction

Whenever the user selects a matrix in the Data Explorer, the high- and low-dimensional representations are rendered to the screen as matrix views and projected views. The two-dimensional points are rendered to the positions determined by the selected projection technique in the Projection View. A classical matrix representation is rendered in the Interactive Matrix Visualization.

A normal interaction work flow incorporates the following steps:

1. **Choice of an interaction start point by ordering the matrix:** although it is possible, a user will usually not order a matrix from scratch. Rather, the user will investigate the outcome of different matrix sorting algorithms one by one. Whenever a global matrix reordering algorithm has been chosen, the ordered matrix will be rendered in the Interactive Matrix Visualization View and the Projection View will be populated with vertices and their current linear arrangement (if the user chooses to investigate this edge configuration).
2. **Inspection of patterns in the projection space:** as described in Section 5.4.2 the user will search for patterns leading to an unwanted sorting behavior.
3. **Selection of vertices in the projection space:** after having found a pattern, the user will select groups of vertices that account for the unwanted patterns.
4. **Invocation of a local sorting algorithm on the selected vertex group:** a *PartSort* and *local reordering* can be invoked with the respective buttons in the Interactive Matrix Visualization. For decision support, thumbnails showing the potential reordering outcomes are overlaid in the Projection View. The user can choose one of the options and inspect its impact on the rest of the matrix in the Interactive Matrix Visualization View (cf. Section 5.4.3).

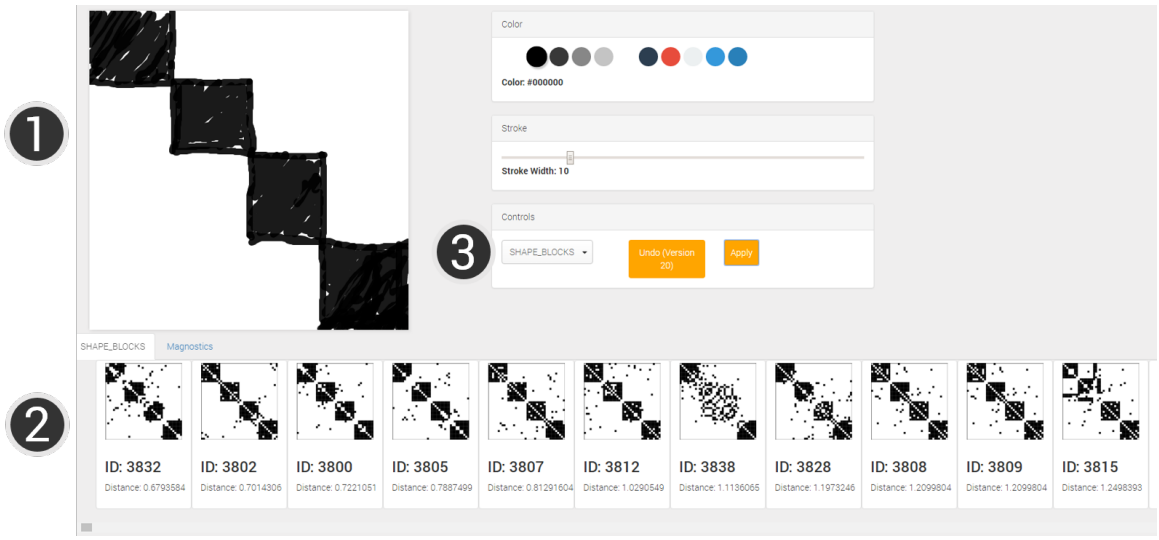


Figure 5.9 Query-By-Sketch and Query-By-Example interface for exploring large collections of matrix plots. The user can either sketch in the canvas (1) an approximated matrix pattern and retrieve a ranked result list (2) according to a selected MAGNOSTICS FD (3) or use an example from the result list image to construct a sketch.

5. **Reordering of groups in the projection space:** after (potentially) multiple iterations of the above steps, the user can rearrange the locally optimized groups. This is done by modifying the edge configuration manually in the Projection View. In an "Edit" mode the user can interactively draw new edges into the linear arrangement edge set. A valid and formally correct linear arrangement is reestablished after each edge modification automatically (cf. Section 5.4.3). Results can be stored as so called edge configurations, enabling the user to jump back and forth between the modification decisions (undo/redo behavior).

5.5 | Sketch-based Visual Search for Navigation and Exploration of Matrix Spaces

Large collections of networks may occur in many applications. As an example, von Rüden reports in [Rüd+15b] about existing matrix collections from the high-performance computing domain with more than 290,000 matrix plots with the analysis goal to retrieve matrices with similar patterns. Another example of large matrix collections is the results of the cross-product between networks/tables and their matrix reordering algorithms as presented in [Beh+16b].

However, exploring large collections of matrices can be a tedious task and may become even more challenging if the task is to retrieve a specific pattern of interest. Generally, for all comparative and similarity tasks information retrieval systems can be beneficial, since

their visual abstractions, interactions and result presentation approaches will enhance the user satisfaction and –more importantly– efficiency.

5.5.1 | Query-By-Sketch for Pattern Retrieval

For the retrieval of similar patterns in matrices we developed a prototypical query-by-sketch interface, as depicted in Figure 5.9, which can also be accessed online[3]. The query-by-sketch interface allows the user to describe intuitively the expected visual patterns by drawing a sketch of a sought matrix pattern (Figure 5.9(1)). The query can be composed by a free-hand drawing tool with undo and erasing functionality.

We compare a given sketch with a given matrix data set using a feature-based similarity function. Specifically, the user may choose feature descriptors (FDs) individually, in comparison or jointly in the form of the MAGNOSTICS feature set. We also experimented with a feature weighting scheme to combine individual FDs based on the users' intuition. For each individually selected FD the feature vectors of the sketch image and the database images are compared by means of the Euclidean distance to retrieve eventually a comparison score. In the case of the MAGNOSTICS FD set we derive a six-dimensional feature vector by calculating the Distance-to-Base-Pattern for every individual MAGNOSTICS FD. For the image database this can be done in a preprocessing step. The query image feature vector is subsequently compared with the Euclidean distance to the image feature vectors in the database.

As a result of the automatic comparison the user sees one or multiple ranked list(s) of similar images (Figure 5.9(2)) according the selected feature descriptors (Figure 5.9(3)). Figure 4.25 shows an example using the BLOCKS descriptor for the matrix reordering data set collection with 4,313 matrices [Beh+16b]. Matrices show several blocks for the drawn sketch.

5.5.2 | Query-By-Example for Pattern Retrieval

After an initial sketch search, the result image may contain a prototype of the expected pattern. Therefore, the interface allows the user to decide that a result image should be taken over to the query panel. Figure 5.10 shows how a result image is overplotted on top of an initial user sketch and modified in a subsequent user interaction.

With this interaction metaphor an iterative exploration chain of query-and-result sequences can be established and the user can interactively explores the matrix dataset at hand. During all times the user may change the FD to account for an inexpressive/not discriminative FD, leading to a (too) stable or converged result set (local optimum).

[3] http://magnostics.dbvis.de/#/sketch

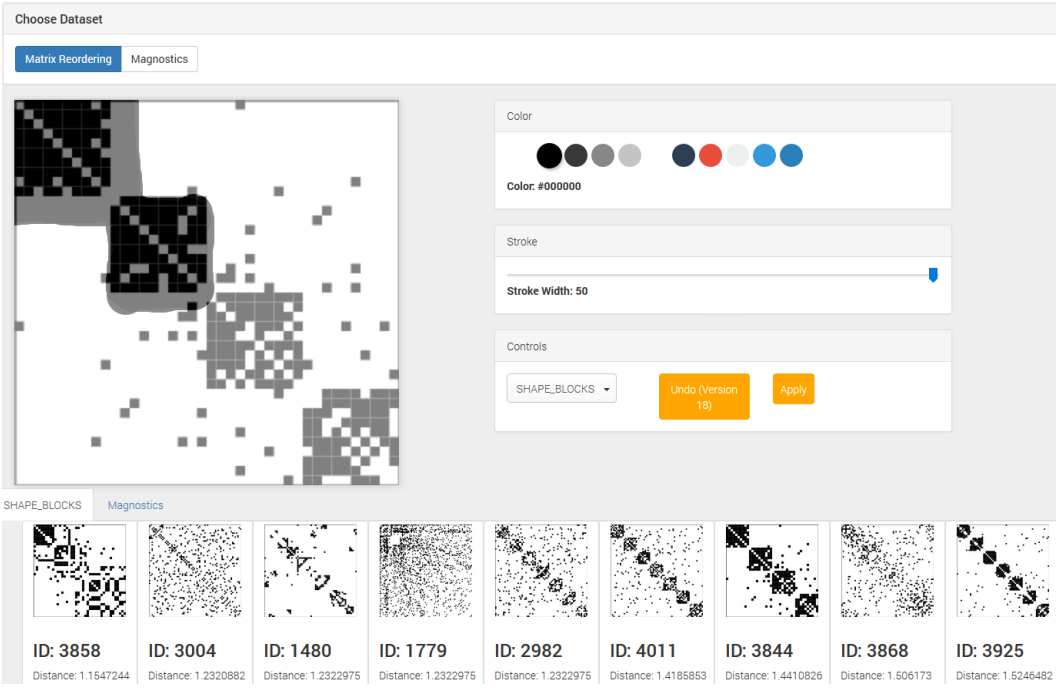


Figure 5.10 Query-By-Example and FD ranking comparison of different FDs on the same sketch image.

5.6 | User-Guided Visual-Interactive Similarity Definition

Adapting the similarity calculation is a core user-adaption in the visual analytics pipeline and has a direct impact on the algorithm and model performance. As we have already seen in Section 4.5, the projection-based similarity function allows us to visually interpret its results (c.f. Section 3.6.1) and gives rise to interpretable interactions for a similarity definition and adoption.

5.6.1 | User-guided Matrix Comparison in the Matrix Projection Explorer Framework

Our projection-based distance calculation technique, as presented in Section 4.5, is proposed as a basis for a range of applications. Specifically, we see two important domains: (1) Applications using the projection-based approach for the distance calculation, and (2) Applications that enable domain experts to draw conclusions from the projection view as a complementary view for visual matrix analysis. We developed a prototype that implements the proposed distance computation and uses it to support visual exploration

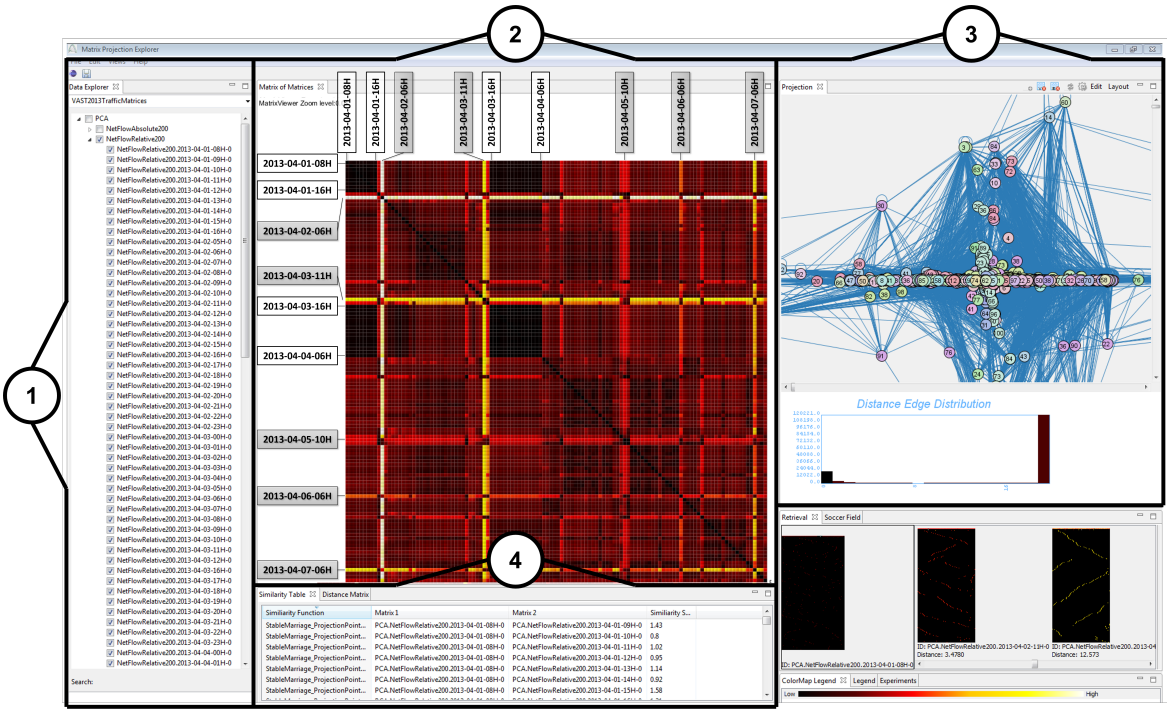


Figure 5.11 Matrix Projection Explorer is used to visualize matrices and their projections. The overview (2) shows a distance meta-matrix of all pairwise matrix distances for the VAST Challenge 2013 dataset with 120 matrices. Patterns, like closely related (dark groups) and outlying (light rows) matrices, stand out. The projection view (3) lets the user explore the selected matrices' structural similarities expressed in the projection space.

in sets of matrices (Matrix Projection Explorer). We now describe the system and proposed analytic work flows.

Our system consists of two parts: (1) A NoSQL database to store matrices and cache projections. (2) A visual front end, consisting of four components (see Figure 5.6):

1. *Data Explorer View*: Allows the selection of a set of matrices to be examined and visualized.
2. *Matrix Visualization View*: Shows matrix visualizations in a heat-map-style display (for individual matrices) or an overview of a set of matrices (sorted by distance). A semantic zoom function allows the transition between projection view ports, the matrix view and the meta-matrix overview (see e.g., Figure 5.12).
3. *Projection View*: The main interactive component, which can be found at the lowest semantic zoom level for every matrix pair, depicts the chosen matrices' two-dimensional projections. The distance calculation is represented by means of the graph matching. The user can interact visually with the algorithm by excluding or including vertices and edges from the matching or adapt the penalty calculation to the task at hand.

4. *Similarity Table*: Shows the distance scores for a set of two or more matrices selected in the Data Explorer View.

In addition, a legend allows the highlighting of matrices, a color map shows colors applied in the matrices and a retrieval view ranks the most similar matrices to the current selection in ascending order. Finally, Views for experiments allow the user to start and keep track of experiments. As an alternative to the Matrix Visualization View an interactive matrix visualization can be added, which would allow the user to highlight row/column selections in the projection pane and vice versa.

5.6.2 | Workflow and Interaction

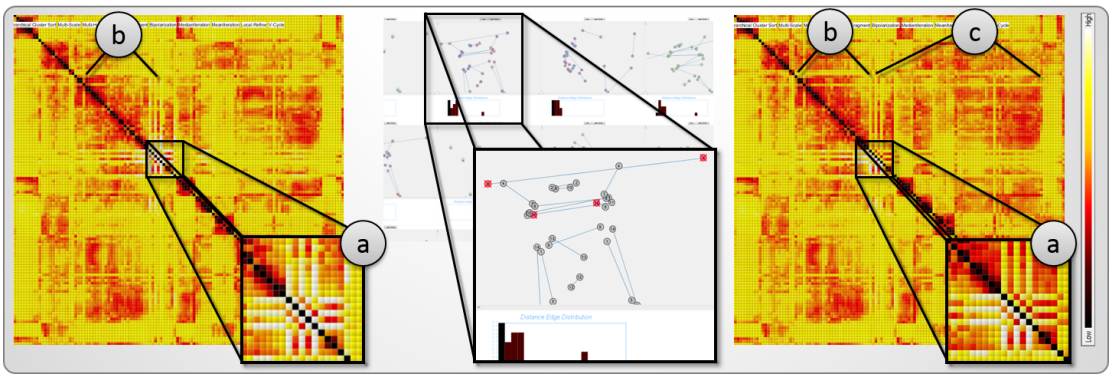


Figure 5.12 Excluding vertices from the calculation helps to filter out aspects of low importance. In this soccer analysis task it makes sense to exclude goal-keepers to find semantically similar game situations, where goal-keepers have a low impact.

When the user selects a set of matrices, their high- and low-dimensional representations are rendered on the screen as matrix views and projection views. The two-dimensional points are rendered at the positions determined by the selected projection technique in the projection view. The matrix representation is rendered following the best practices presented in [Fek04]. All pairwise distances between the matrices are also calculated and the user can switch to a *meta-matrix* representation of these. We experimented with a MDS projection of the distance values. While it proved beneficial in perceiving similarities (i.e., groupings) we rejected the idea, because it was hard to reflect interactive changes to the distance calculation in a visually traceable manner.

For inspection purposes, the user can select a calculation to show the best possible bipartite graph matching allocation, as described in Section 4.5.1. The matching is visualized by adding edges to the projection view (see Figure 5.6 (3)), connecting the matched vertex pairs. The user then has the means to visually interpret the matching and reason on its performance. In addition to this static procedure, the user can *interact* with the similarity algorithm in a feedback loop by excluding or including vertices and recomputing

the bipartite graph matching accordingly. The result is then added to the similarity view, building a possible feedback loop.

5.6.3 | User-Guided Distance Calculation

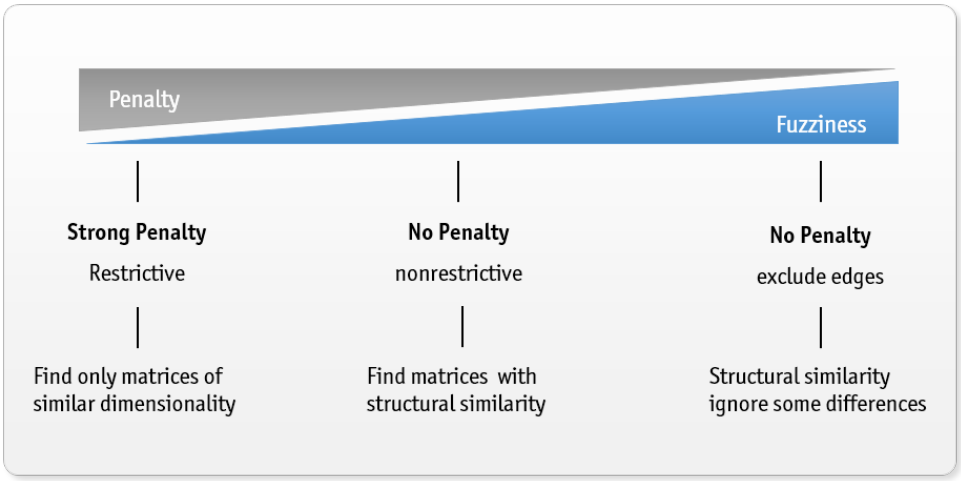


Figure 5.13 User-steerable distance score modification for projection-based distances: Users can apply a strong penalty to formulate restrictive similarity queries, emphasizing the structural and topological similarity. On the other hand, no penalty or even the exclusion of long/short edges introduces fuzziness in the process and allows matching structurally similar matrices while ignoring some differences.

Our approach explicitly supports the interactive guidance of the distance calculation by the user. As for example, Figure 3.7 or Figure 5.12 show the user can zoom from an overview distance matrix into a specific area of interest and investigate the impact of the dimensionality differences to the distance score. By zooming from the overview meta-matrix into an area of interest, the user can investigate each pairwise matrix comparison in the projection view. In this view three different user interactions are possible:

1. Selecting and deactivating projection points allows the exclusion outliers or otherwise irrelevant rows/columns;
2. Selecting edges with a lasso or by clicking on the respective distance histogram bin, allows the exclusion ranges of edge lengths from the distance;
3. Adapting the penalty function (clicking on the distance histogram penalty bin or via a specific penalty dialogue) allows modification of the effect of the matrices' size differences on the distance.

With these interactions, a large part of the fuzziness spectrum regarding the graph matching problems can be covered. Figure 5.13 describes the relationship between the penalty on the one hand and the exclusion of edges on the other hand. If no penalty is artifically added (Figure 5.13 (middle)) all projection points that can be matched will be

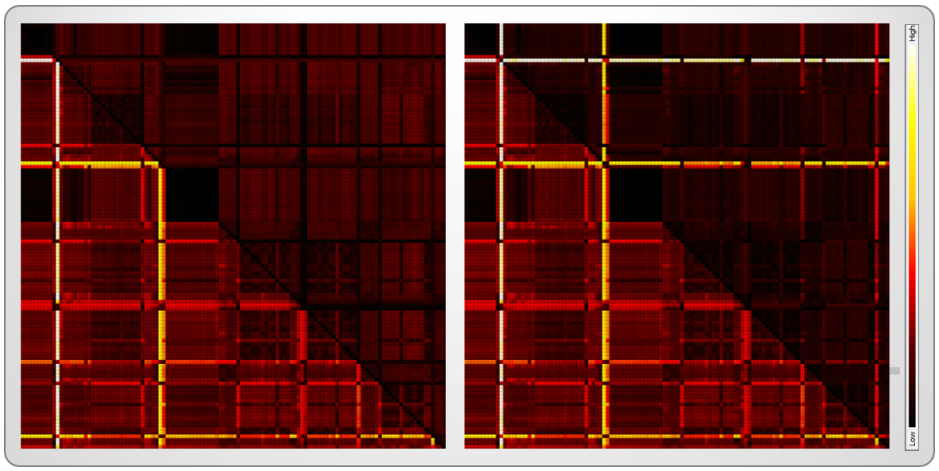


Figure 5.14 Changing the penalty function has a large impact on the appearance of the distance meta-matrix showing all pairwise matrix comparisons. From left to right, the `ZeroPenalty` and `MaxDistSquare` penalty functions are rendered in the upper diagonal part of the matrix. The lower part shows the `MaxDist`, for reference purposes.

matched and all dimensionality differences are ignored. Thus, structurally similar matrices are retrieved. If the user wants to emphasize the dimensionality differences than a strong penalty function can be applied (Figure 5.13 (left)). In contrast, the user may decide to ignore the dimensionality and even ignore some level of structural modifications by excluding long edge-ranges in the projection panel (Figure 5.13 (right)). We demonstrate the usefulness of these interaction mechanisms in the use cases in Sections 5.8.2 and 5.8.2.

Figure 5.14 visually depicts the impact of changing the penalty function for the VAST 2013 Challenge Dataset. From left to right, the `ZeroPenalty` and `MaxDistSquare` penalty functions are plotted in the upper diagonal part of the matrix. The lower diagonal part shows the `MaxDist`, for reference purposes.

The choice of penalty function is closely related to the graph matching tasks at hand. If the size differences are important a “strong” penalty must be applied. The lower the penalty, the more *fuzzy* the distance calculation becomes. In other words, a “weaker” penalty function, e.g. `ZeroPenalty`, shifts the focus to matrix comparisons that ignore size differences and try to derive a statement from the available information. Other penalty functions are possible, ranging from a static penalty score to a penalty that reflects the situation in high-dimensional space.

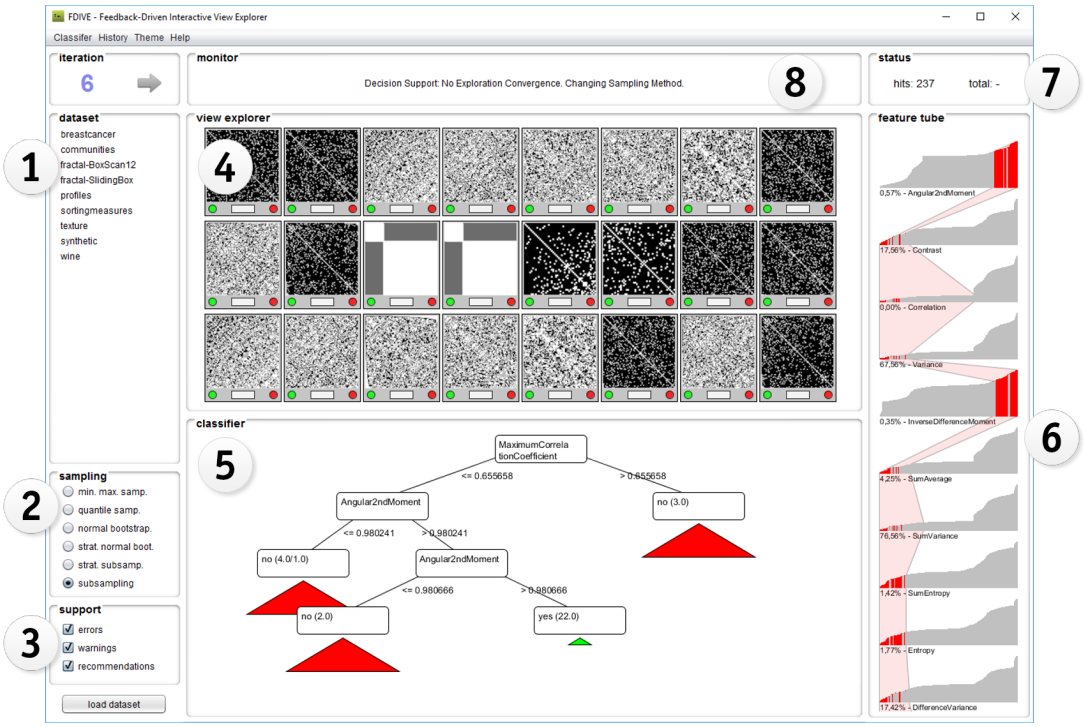


Figure 5.15 The user interacts in the View Space Explorer by choosing relevant or irrelevant examples (4) from a small sample set. An incremental decision tree visualization (5) and a feature tube visualization (6) help to assess the exploration convergence. Specific decision support intervention points can be enabled/disabled in (3). Additional decision support notifications are shown in (8).

5.7 | Feedback-Driven Assessment of Relevance for Matrix Representations

Our current data-agnostic society is driven by the prevalent perception that most data contains valuable information, which can be retrieved in a later information retrieval process. To this end, all kinds of data are stored and analyzed. The business consultancy McKinsey even forecasts that the “data scientist” will become one of the most important jobs in the US in the coming decade [McK11]. While the collected data may be rich in information, it is still highly challenging identifying appropriate views on the data sets. As an example, an n-dimensional numeric data set allows to render $(n \times (n-1)/2)$ distinctive views only by using a projection onto two distinctive dimension axis. This spans a large exploration space in which interesting views need to be identified. To make matters worse, the most valuable data views exists in relation with the users’ current tasks, intentions, and current context.

A range of approaches to deal with the *interesting view problem* were developed over the years. For example, *Focus+Context* systems [Shn96; BGS01] lead the users in an

overview to areas of interest and let them explore these areas with drill-down mechanisms. Focus+Context systems are an established and approved method, but often tend to be expert systems restricted to specific data set characteristics and/or user interactions. These systems potentially introduce misleading abstractions, which ultimately can lead to wrong exploration paths. Semantic zoom interfaces [CMS99b] help also to deal with this problem. Here, the user explores the data set at varying levels of abstraction/detail, starting with a highly aggregated version of the underlying data. The more the user "zooms" into the data, the more details become assessable. Alternatively, cluster-based navigation systems partition the exploration space into a range of distinctive clusters that are represented by a small amount of prototypes. Choosing the prototypes relates directly to the interesting views problem. A further well-understood, yet simple, approach to tackle the interesting views problem is to focus on faceted search algorithms that operate on the available meta data. This requires a manual annotation and insertion of meta data, which is often prone to errors or missing values.

Directly related to the interesting views problem is that a *query formulation* [MRS08] on complex data sets is difficult. This is primarily due to the fact that the collected data sets are multivariate and high-dimensional in nature. To tackle this problem, novel querying mechanisms, such as query-by-example or sketch-based interfaces –such as presented in Section 5.5– appear to be beneficial research directions. Here, the systems rely on the assumption that users have a priori an initial understanding of the interesting patterns under investigation. This explicit definition of interest is time consuming, particularly if interest rules need to be updated during the exploration process.

In the following we present an approach to the interesting view problem, which focuses on the interplay between the user and an automatic decision-support system. In an iterative work flow the user assesses whether a set of presented views are of interest or not. These views can be arbitrary, but suitable, visualizations for the data exploration task at hand. A classification system learns from the previous user decisions, while notifying the user in case of potentially wrong decision paths and major decision path divergence. The general idea is inspired by multimedia retrieval approaches, where the user's explicit relevance feedback on retrieval results is used to recommend additional previously unseen results [MRS08]. In contrast to the major work in this field, the presented relevance feedback mechanism is incorporated into a feedback loop, which adapts to the earlier user decisions.

Our approach relies on the basic assumption that for most and even complex data visualizations a comprehensive set of feature vector descriptors can be found, either in the data-, in the image space or in a combination of the latter, that can be mapped to its analytic benefit.

The outlined approach has to be seen as a framework for an interactive relevance feedback driven data exploration process. One of the benefits of the framework is that

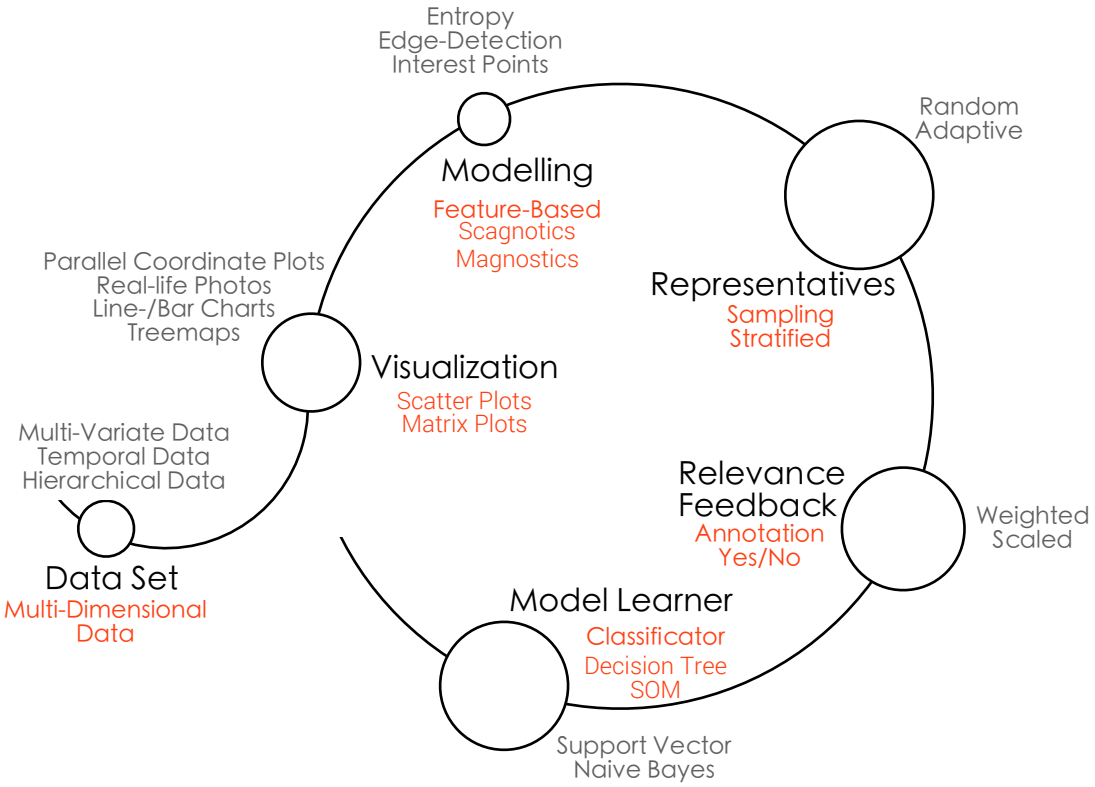


Figure 5.16 The interaction flow in the Feedback-Driven View Exploration Framework: The user chooses a dataset of interest. A meaningful visualization type is selected (automatically or manually). The underlying data is described by means of feature descriptors. A range of representatives are shown to the user, which are interactively tagged for their relevance in the exploration process. A model learner tries to reflect the user's preferences and shows a new representatives set to the user.

a great variety of design alternatives can be applied without changing the fundamental approach. We are showing one reference implementation of the framework by using an incremental decision-tree classification to guide the user in a large scatter plot exploration space. However, it has to be mentioned that we are not restricting ourselves to scatter plot visualizations, but allow any type of visualization technique as long as a descriptive feature vector space can be found.

5.7.1 | A Framework for Feedback-Driven View Exploration

The basic idea of a feedback-driven view exploration approach is to put users into a steering position to determine what they want. Figure 5.16 outlines the main work flow in the semi-automated exploration process. In a normal sequence of actions the user chooses a data set under investigation and decides for a meaningful visualization to assess the underlying data. The framework uses an appropriate feature descriptor from the data-

and/or the image space to represent the data. The resulting feature vectors are the basis for the visualizations. In case of a feature descriptor operating on the data space characteristics, such as the data distribution or compressibility, will be represented. An example would be to measure the convexity of a scatter plot or the diagonal-block related measures of a matrix plot. Image features will be used to reflect the visualization's –or depiction's– characteristics. An example would be to measure the number of interest points for a real-world image. For the overall exploration the choice of the feature descriptor is crucial, since every descriptor is only capable of reflecting certain characteristics of the underlying data.

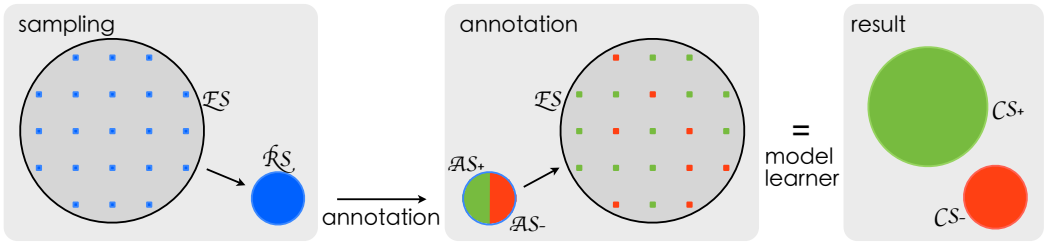


Figure 5.17 Four different sets are distinguished in the approach: (1) The exploration set $\mathscr{ES}$ contains all possible views. (2) A sampled version of the exploration set will be presented to the user ($\mathscr{RS}$). (3) The user annotates this set for interesting, respectively uninteresting, views ($\mathscr{AS}^+$ versus $\mathscr{AS}^-$). (4) A classifier learns a mapping of the exploration set into potentially interesting views ($\mathscr{CS}^+$), respectively uninteresting views ($\mathscr{CS}^-$).

As Figure 5.17 depicts, from a potentially very large exploration set of visualizations, denoted as $\mathscr{ES}$, only a limited amount can be presented initially to the user. We will denote this subset as the representation set $\mathscr{RS}$. The choice of the items in $\mathscr{RS}$ can be random, deterministic, or iteratively adaptable (cf. Section 5.7.4). In the general feedback-driven view exploration framework the representation choice adapts according to the user's decisions. In an exploratory search phase a uniformly distributed sample should be made available, while in a confirmatory search only subpopulations of $\mathscr{ES}$ need to be presented. Generally, users might not be able to manually assess the entire data set. Hence, after a broad beginning only parts of the exploration space will be presented to the user. From $\mathscr{RS}$ the users can either choose visualizations of interest or express their dislike. Thus, an implicit knowledge gets explicitly available and accessible to the framework. A model learner is used to reflect the expressed user preferences by classifying the unseen items in $\mathscr{ES}$ as potentially relevant, denoted as $\mathscr{CS}^+$, or potentially irrelevant, denoted as $\mathscr{CS}^-$. We can assess the model learner's (un-)certainty in the classification. Relevant and irrelevant items can be matched to both classification sets $\mathscr{CS}^+$and $\mathscr{CS}^-$ to find visualizations with an (un-)certain interestingness mapping.

The task is now to find a good mapping $f : \mathscr{ES} \mapsto \mathscr{RS}$, such that the user on the one hand will find interesting patterns and on the other hand is still able to explore the dataset

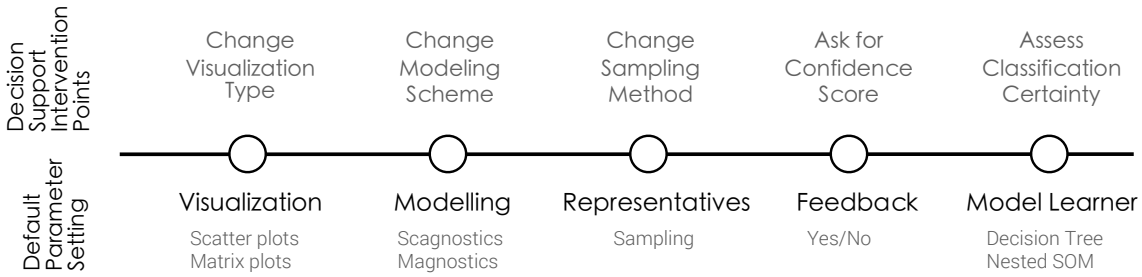


Figure 5.18 The Decision Support can change essential parameters in the Relevance Feedback Driven View Exploration Framework if the exploration convergence stagnates: For example, it can recommend switching to a more appropriate visualization type with an appropriate modeling scheme or ask for a confidence score if the user annotation is obviously misleading.

without loosing too many details. A secondary goal is to let the user iterate only a few times through the feedback loop.

Another basic idea of the feedback-driven view exploration framework is that user decisions should have an impact on the exploration. Hence, the task of a *decision support system*, as described in Section 5.7.4, is to assess the stability and convergence of the exploration path. Figure 5.18 outlines potential intervention points in the feedback-driven view exploration framework. As an example, the choice of the data set implies a (feature-based) modeling of the data. While this choice might be appropriate in an early exploration phase, it might be too restrictive, respectively broad, in the later phases. Thus, either the visualization may be adapted or the feature vector representation. One heuristic for this recommendation could be that the exploration path stays rather unspecific and does not converge even after a number of iterations. Another intervention point is the outcome of the model learner. Here, the decision support acts as a supervision instance, which, e.g, allows assessing the certainty of the classification subsystem.

5.7.2 | Exemplified Instantiation of Feedback-Driven View Exploration Framework

In the following we want to present one instantiation of the general semi-automatic exploration work flow from Figure 4.8.3. In each of the following sections we will outline the overall idea for the respective work flow step, describe the current implementation and reflect our design rationale by describing alternatives and further prospects.

Visualization The general idea of the *Visualization* step is to illustrate the given exploration set ($\mathscr{E}\mathscr{S}$) in a reasonable manner. This step initiates our framework's interaction pipeline in Figure 5.16. The choice of the visualization technique is important and depends on the given data set. Effective visualizations help in the decision-making process,

while reading ineffective visualizations can be time-consuming and potentially leads to wrong decisions.

In one of our exemplified instantiation of the feedback-driven view exploration framework we use matrix plots to represent relational data. In another version of the framework we use *scatter plot visualizations* to represent a continuous high-dimensional data set.

As mentioned above, the choice of the visualization technique depends on the data set under consideration. In our case, scatter plots are appropriate, but in case of other data types, such as temporal, hierarchical or textual data, alternative visualization techniques are better suited for the view space exploration in our pipeline. To name just a few alternative examples, line charts are suitable for temporal data, treemaps can represent hierarchical data, and word clouds can be used to depict text content.

Modeling The general idea of the *Modeling* step is to characterize all visualizations of the exploration set $\mathscr{ES}$ and to compute a uniform model for the *Model Learning* step (cf. Section 5.7.2). It computes a feature-based vector for every visualization. By this means, the similarity for each individual visualization can be automatically compared and used for further sampling or classification methods.

We experimented with the *Scagnostics* approach [WAG05b] to characterize the scatter plot contents, since it is capable of describing point distributions by meaningful measures. For the matrix plot dataset we are using the *Magnostics* related feature descriptors separately to quantify the visual patterns in the matrices.

Depending on the chosen visualization (cf. Section 5.7.2), different descriptors have to be used to extract feature vectors. While, *Kernel Density Estimators* or *Regressional features* could be used to extract suitable features for scatter plot point distributions, image-based descriptors will be more appropriate to describe real world images. In the case of structure conveying visualizations, such as treemaps, one option is to apply an *Edge-Histogram Descriptor* or line extraction algorithm [DH72] to describe the general shape of the visualizations content. For text visualizations, a dictionary-based approach can be applied to compare the textual content inside visualizations.

Representatives The general idea of the *Representatives* step is to select a manageable number of items from the exploration set $\mathscr{ES}$. This presentation set, denoted as $\mathscr{RS}$, is presented and judged by the user. Hence, its functionality highly influences the exploration process. The selection procedure is exchangeable in our implementation of the view space exploration pipeline.

In the current implementation, we are experimenting with content-based *sampling-based* approaches, such as discussed in [HKP11b], to select a range of items in $\mathscr{ES}$. A Min-/Max sampling option selects two representatives for each of the feature value ranges (cf. Section 5.7.2). For the Scagnostics example, 18 representatives can be judged by

the user: two items (one min-value and one max-value representative) for each of the nine Scagnostics features. To increase the number of items in $\mathscr{RS}$ and to reflect the data distribution, a quantile sampling, a (stratified) bootstrapping and a stratified normal sampling method can be applied. The user can select how many items should be retrieved, resulting in $|features| \times requestedSamplingItems$ items in $\mathscr{RS}$.

One reason to apply sampling is that $\mathscr{RS}$ is available instantaneously without much computational effort. One obvious disadvantage is that the sampling potentially shows a series of outliers in the data distribution. However, this effect can be neglected in case of the quantile sampling method ($amountQuantiles > 2$).

As stated above further design alternatives are possible and may be considered if the representation items are not perceived as appropriate. One computational expensive solution would be to apply a density-based clustering in every projection pane of the feature space. N modeling features would lead then to N projection panes. From the clustering results a range of representatives could be selected by choosing, e.g., the medoid of the found clusters.

Relevance Feedback The general idea of the *Relevance Feedback* mechanism in a semi-automatic exploration pipeline is to give the user the ability to steer the retrieval process. The user can categorize the presented items into relevant, irrelevant, and neutral examples. Relevant annotated items, denoted as $\mathscr{AS}^+$, represent potential *hits* for the search process. Irrelevant examples, denoted as $\mathscr{AS}^-$, depict items that lead to wrong or uninteresting search paths.

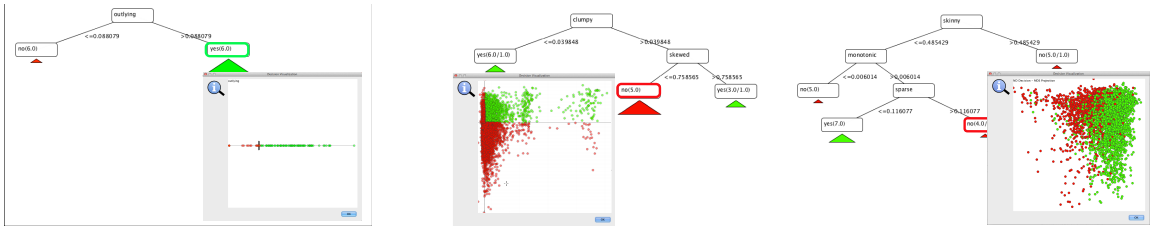
In the current implementation, depicted in Figure 5.15, the user can click on green and red buttons to express his like, respectively dislike.

Alternatively to the binary decision approach, a weighted relevance feedback could be implemented for finer granularity assessment of the user feedback. In this case, the users would have to judge the interestingness of the presented items in terms of a linear scale. Also a star rating, as it is known from product reviews, would be possible. We decided against a weighted relevance feedback system, since these kinds of decisions might be hard to judge for the user in the beginning of the exploration process and involve additional interaction overhead.

Model Learning The goal of the *Model Learning* step is to reflect the user's preferences. In the best case, the system learns the user's intention after only one iteration and retrieves only positive examples. The worst case scenario is that the model learning cannot grasp the user's intention after a finite number of steps, leading to always negative examples. The pipeline would then iterate until all irrelevant items are excluded and only relevant examples are left. Hence, the search eventually converges.

A model learner has to be able to revise and refine previous decisions. Whenever the system restricts existing decisions we assume that the user also refined his/her understanding of the explored items. Thus, we assume that the exploration path is "correct". On the other hand, a revision of existing decisions corresponds to a potentially "wrong" exploration path.

On top of the exploration steering function of this component we put another prerequisite on the system: It should externalize its decisions in a visual depiction (cf. Section 5.7.3).



(a) 1D decision on one classification attribute

(b) 2D decision on two classification attributes

(c) nD decision on n classification attributes

Figure 5.19 The incremental decision tree allows assessing the complexity of the formulated exploration query. Additional meta visualizations depict the value distribution for 1D classification decisions (decision on one classification attribute), 2D decisions in a confusion matrix and nD decisions in a MDS projection of the classified items similarity.

Incremental Decision Tree for Scagnostics: In our Scagnostics implementation we decided to use a classification system to approximate the user's preferences. Our model learner is inspired by the idea of an iterative decision tree, such as presented in [YF12]. In contrast to a normal decision tree, iterative decision trees retain most of its structure after the initial training. This allows the user to perceive the structural development over time and can only be achieved if a fundamental confinement of the decision tree algorithms gets derestrict: Nodes corresponding to a parameter (-range) can occur multiple times in the same decision tree. However, in line with the decision tree logic, these multiple occurrences may not violate already applied range restrictions (value > 0.5 leads to *yes*, but also value ≤ 0.5 should lead to *yes*).

In a standard course of action, we are expanding the decision leafs in each learning iteration of the pipeline. We are differentiating between *outer decisions* and *inner decisions*. While outer decisions modify the outer boundary of the decision space formed by the n selected features (cf. Section 5.7.2), inner decisions lead to subareas in the already excluded/included decision space, which should be included, respectively excluded, from the search.

For outer decisions two cases are possible without violating the idea of a decision tree: 1) A *yes* node, representing a set of relevant classified items, on a *yes path* gets split up. In this case, the user found that the classification is too unspecific and should be narrowed. One example for this case is shown in Figure 5.19 (c), where the monotonic feature range was modified from $[0.09, 1.0]$ to $[0.12, 1.0]$. 2) A *no* node,representing a set of irrelevant classified items, on a *no path* gets split up. In this case, the user found that the classification is too specific and should be broadened.

Inner decisions are improving constraints set in earlier decisions. Here, also two alternatives are possible without violating the decision tree idea: 1) A *no* node on a *yes path* gets split up. In this case the user found that learned constraint is limiting the exploration and should be made less restrictive. 2) A *yes* node on a *no path* gets split up. In this case the user found that learned constraint was not restrictive enough and should be strengthened. One example for this case is shown in Figure 5.19 (c), where a monotonic value above 0.006 alone would lead to a positive classification. This classification gets restricted by the sparsity feature descriptor below 0.11. In both presented cases of an inner decision parallel decision paths, or split-ups, could be a result.

Nested Self-Organizing Maps for MAGNOSTICS: For our MAGNOSTICS implementation of the Feedback-Driven Interaction pipeline we experimented with a Self-Organizing Map (SOM) model learning approach. SOMs are one instance of an artificial neural network architecture initially proposed by Teuvo Kohonen [Koh82; Koh97; Koh88]. Their central goal is to project a high-dimensional data space into a lower dimensional space in which data similarities can (still) be reflected. In our case, we assume that –after a learning phase– each cell in the SOM layout contains a distinct set of patterns. Figure 5.20 depicts one cell in which mostly off-diagonal block pattern matrices are grouped together.

Generally, SOMs are "one-time" model representatives that learn the data modalities in five consecutive steps:

1. **Initialization:** All neurons in the regular grid structure are initialized by a random vector. In our case the initialization vector is of the size of the currently chosen feature descriptor and each component is randomly assigned to a value of the predetermined feature vector value range (of the respective component).
2. **Competition:** All neurons in the network are competing for the assignment of an input vector. The neuron with the lowest Euclidean distance between the current/learned neuron feature vector and the input vector wins the competition.
3. **Cooperation:** The winner neuron stimulates its neighborhood so that not only neuron reacts to the stimulus, but rather an entire SOM region. The stimulus impact decreases relatively to the learning duration.
4. **Adaption:** All stimulated neurons are adapted in the direction of the input vector. The adaption intensity decreases wrt. the training duration and the stimulus

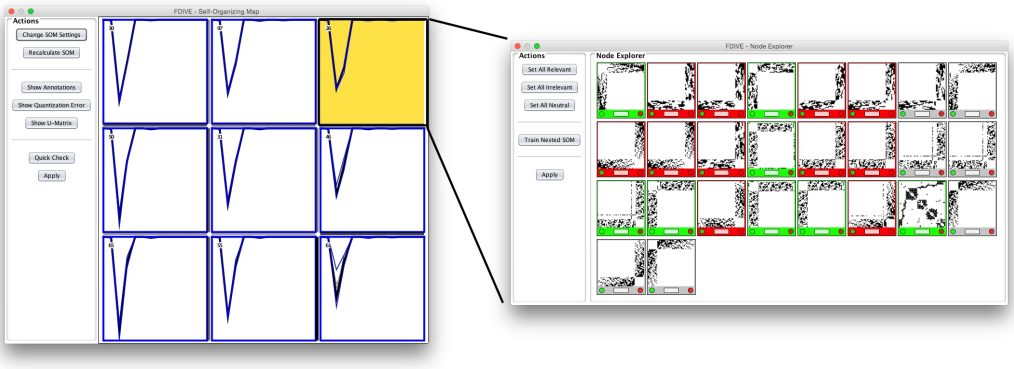


Figure 5.20 Intuition of the SOM Classification/Clustering adaption for the Feedback-Driven View Exploration Pipeline. The selected cell in the upper right shows consistently Off-Diagonal Block Patterns.

weighting. If the learning process has not converged the process is repeated (Step 2).

5. **Classification:** Not directly related to the learning process is the classification process. Here an input neuron is finally assigned to the closest matching neuron, which has to be regarded as the prototype for a whole set of seen input vectors.

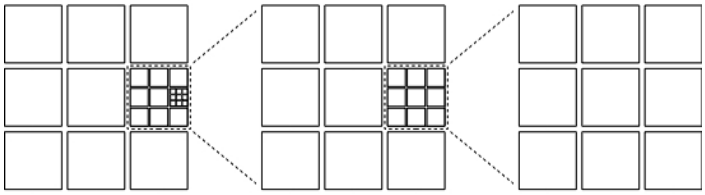
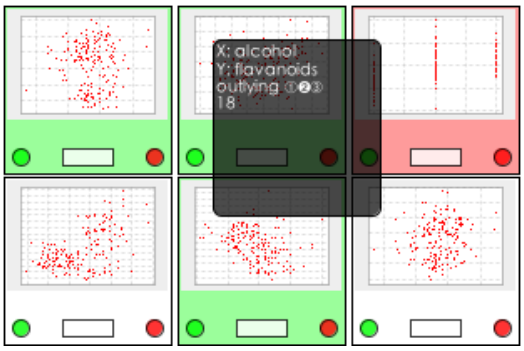


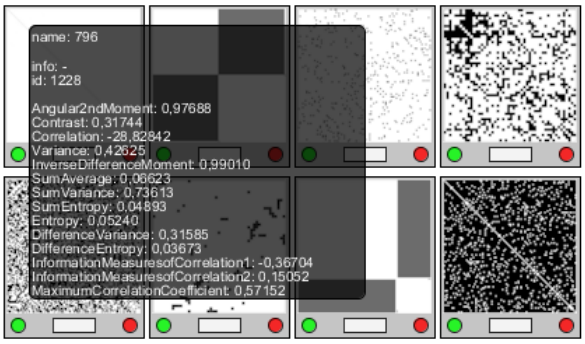
Figure 5.21 Nested Self-Organizing Map concept: Each cell can be recursively split into further SOMs, thus incrementally refining the decision boundaries in a HD decision space.

However, SOMs are on-time learners without any option to adapt to the newly generated knowledge during the exploration process. Therefore, we developed the idea further to a *nested SOM*, as depicted in Figure 5.21. Here a given grouping decision in one cell can be recursively refined with a nested sub-decision, i.e., an off-diagonal block pattern could be subdivided into L-shaped and inverse L-shaped subgroups. The more nesting levels SOM shows, the more distinct decisions have been taken by the user to iteratively refine the interestingness classification model.

Further Alternatives: Alternatively, adaptive learning systems could be applied to learn the user preference model. Here, for example multi-agent learners, such as presented in [RN03] could incorporate likelihood considerations into the learning process, which would be beneficial if many views show similar content. The application of other neural networks structures, such as the once presented in [Yeg09], could be an alternative. Both



(a) A tool tip shows that the view was selected into the presentation set $\mathscr{RS}$, since its outlying value is in the 2^{nd} quantile of the respective feature range.



(b) A tool tip depicts the current feature vector values and gives additional metadata.

Figure 5.22 Users can annotate exploration views (e.g., Scatterplots or Matrix Plots) as uninteresting, neutral, or interesting with the red, white or green buttons.

mentioned model learners are able to learn non-linear decision boundaries in high-dimensional decision spaces. Another alternative, which has not been implemented yet, is a Support-Vector machine classification. Here, visual depictions are available, such as presented in [Hei+12]. The training process is more complex than with the presented approach, but still feasible and a full training is not always necessary as [CP01] demonstrates.

5.7.3 | Pattern Retrieval in the View Space Explorer

In the following we will describe our graphical user interface for the Feedback-Driven View Exploration Framework. Figure 5.15 depicts the visual interface, consisting of the *View Explorer*, the primary interaction component, and a range of meta visualizations, which help to track changes in the exploration process.

View Explorer The View Explorer displays the representation set $\mathscr{RS}$ to the user. It is depicted in Figure 5.15(4). The displayed views are ordered according to the feature descriptor and the selected sampling method; i.e. for each applied feature descriptor one high and one low value in the case of Min-/Max sampling. While an alternative option would have been to sort the views according to their feature vector similarity, the applied ordering allows perceiving the feature descriptor's value ranges more effectively. A detail view for the user annotation options is depicted in Figure 5.22a and Figure 5.22b. A tool tip shows, next to the view id and its metadata, and a visual indication of the sampling set choice; the circled number represents the selected 2^{nd} quantile.

Meta Visualizations Two meta visualizations help the user to assess if an exploration path leads to a convergence (only interesting views). On top of that, the decision support system uses the displayed data to quantitatively assess the exploration convergence.

(1) Feature Space Tube: Figure 5.23 shows the *Feature Tube*, a stacked histogram view per feature descriptor (cf. Section 5.7.2). The histograms are sorted in ascending order to reveal the feature's value distributions. The current decision path corresponds to an interval selection in the n-dimensional feature space, where n is the number of features under consideration. In our case, nine Scagnostics feature histograms are rendered. We are showing the decision path by a tube overlay, highlighting the selected feature intervals of interest.

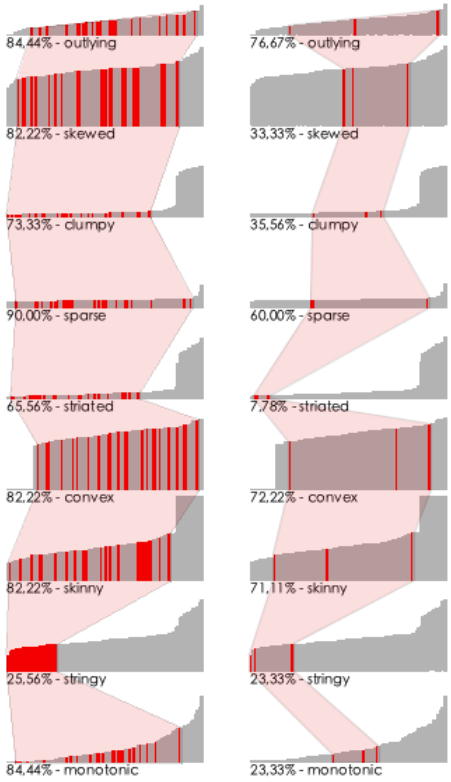
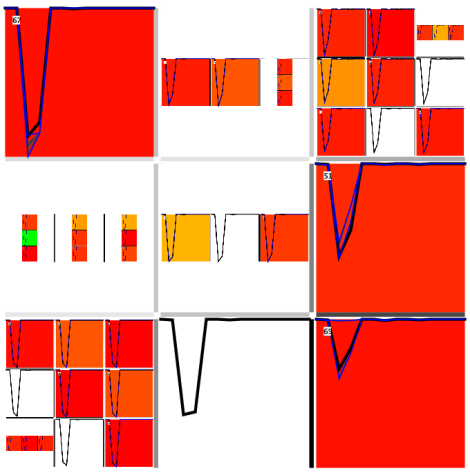


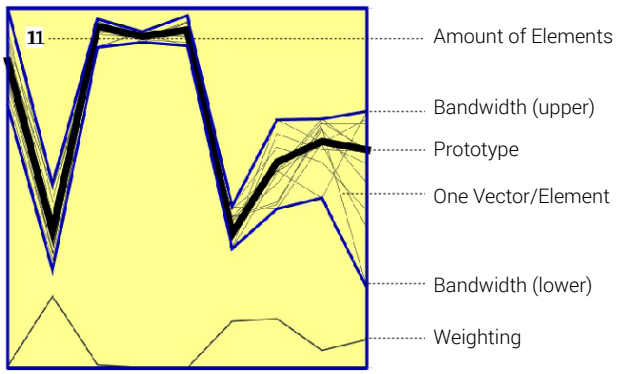
Figure 5.23 Feature Space Tube

The overlay can be used to assess the specificity of the search. A narrow tube relates to a highly specific query –potentially in an advanced status of the exploration process– while a broad tube shows that the exploration is unspecific. Selected intervals can vary in their density of contained items (or views). Dense/sparse intervals show that the current exploration specificity maps to many, respectively few, possible views. By perceiving the change of Feature Tube between two model learning phases, users are able to judge their exploration advancement. Brushing and Linking is used to retrieve a scatter plot selection from the view explorer in all feature histograms.

(2) Incremental Decision Tree: Figure 5.15 (5) and Figure 5.19 shows the *Incremental Decision Tree*, a visualization for the current classifier decisions. In an incremental decision tree every framework iteration corresponds to one level of the decision tree: Level 1 decisions correspond to the projection of the n-dimensional decision space onto the one dimensional subspace of the corresponding classification attribute. Level 2 decisions span a confusion matrix, in which the true-positive (upper right) field corresponds to all positively classified items ($\mathscr{CS}^+$), the true-negative field (lower left) corresponds to all negatively classified items ($\mathscr{CS}^-$). The items in false-positive (upper left) and false-negative (lower right) are potential mis-classifications and cannot be rejected without any reservation. Thus, they remain in the exploration set $\mathscr{ES}$. Level decisions > 2 can be depicted with a MDS projection of the pairwise item similarity. We use of the classical MDS implementation in MDSJ [Pic09] for our purposes. During the annotation phase the user sees the tree visualization, as



(a) Nested Self-Organizing Map Visualization: The regular SOM grid structure is broken up such that recursively SOM grid layouts (potentially of varying sizes) can be rendered into a parent cell.



(b) One SOM cell encodes its represented items with a form of a parallel-coordinates plot. Each item is represented by one line showing the value distributions of the currently chosen feature descriptor.

Figure 5.24 A Nested-SOM classifier visualization helps the user to steer the exploration process by recursively applying sub-decisions for groups of selected objects.

depicted in Figure 5.15 (5). However, if the user wants to get more details about the model learner meta visualizations for the 1D, 2D and nD levels are shown in a dialog on top of the incremental decision tree (Figure 5.19). The purpose of these meta visualizations is to allow the user to interpret how many items the classifier rejected from the exploration. Furthermore, oneö can perceive how many items where close to the calculated decision border.

Alternatively, we experimented with a nested Self-Organizing Map visualization to externalize SOM classifier decisions. Figure 5.24a shows the classifier visualization for a recursion level of three. The glyph within each cell is a form of a parallel-coordinate plot, showing for every feature vector one line, i.e., the 0-1 normalized feature vector values for all feature descriptor dimensions. In our figure, the small green cell on the left represents 87 matrix plots from a 4,313 matrix images dataset, which match the user preferences best. During the exploration process a user can explore each SOM cell to find out how homogeneous the represented images are.

(3) Measures of Exploration Convergence: For a quantitative assessment of change in the exploration process we are quantitatively measuring the *appearance development* of the feature tube and the incremental decision tree visualization. For the feature tube we are storing the covered tube area for each pipeline iteration and calculate the areal difference of the two shapes. If the difference area decreases in two subsequent iterations, the search converges gradually and we are able to measure a *convergence factor*. If the area change stagnates or even increases the user did not advance in the view exploration. For the incremental decision tree, we are able to measure a binary convergence factor.

Either, in the negative case the model learner returns the same decision tree several times (no further learning improvement) or new tree leafs need to be added (learning progress). If the classification training results in the same tree twice we are interpreting the result as an exploration stagnation.

A wide range of other convergence measurements are possible. The simplest is to relate the number of items in $\mathscr{ES}$ before and after an application of the classifier. Another option is to calculate a similarity value for the decision tree appearance before and after the application of the classifier, as e.g., presented for general tree structures in [Hes+14]. However, this option can only be applied if the decision tree is built from scratch, rather than not incrementally. In the future we are planning to experiment with an adaptive convergence measure that takes multiple decision criteria into account. In all cases of a slow or stagnating convergence we are applying *counter measurements* to steer the exploration process. The first measurement is to intervene in the representation finding (cf. Section 5.7.2). In our case, we are changing the used sampling function. The next level of intervention is to increase the number of suggestions for annotation candidates (cf. Section 5.7.4). While the number of suggestions decreases in the normal convergence cases, we are here boosting the lowest similarity, such that more (even less similar) items are recommended for annotation.

5.7.4 | Enhanced Decision Support for Feedback-Driven View Exploration

One of the primary advantages of the presented exploration framework (cf. Section 5.7.1) is that it allows for an automatic supervision of the exploration process. This supervision can be used to investigate and monitor the actions taken by the user. Thus, it becomes possible to make use of a user feedback loop whenever an action is not meaningful, potentially incorrect, or could be improved on the fly.

Users are notified about a potential intervention with the help of dialogs. These dialogs contrast the current user selection with an automatically created suggestion. Most importantly, the decision support system forecasts both options' outcome and presents them to the user. In the case of conflicting decisions between the user and the decision support system, the user decision are preferred to the algorithmic decisions.

In the following we are referring to our implementation of the feedback-driven view exploration framework as it is described in Section 5.7.2. Specifically, we are rendering matrix plots modeled/described by one of the Magnostics features (Blocks) feature descriptor. We are applying a sampling-based approach to find representatives. The user gives binary feedback, whether an item is relevant or rather irrelevant; the incremental decision tree algorithm classifies the exploration set $\mathscr{ES}$ into positive $\mathscr{CS}^+$ and negative classified items $\mathscr{CS}^-$.

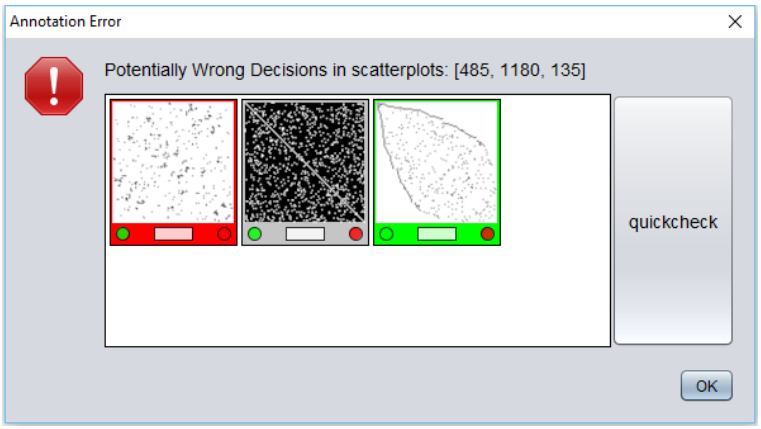


Figure 5.25 Potentially wrong decisions are intercepted by the decision support system to keep the model learning in a consistent state. The outcome of each decision can be anticipated without applying it to the model learner by using the quick-check functionality button.

Handling Potentially Wrong Feedback Decisions

Decisions are ambiguous and potentially wrong if the same view (scatter plot) has been marked both irrelevant and relevant in the current iteration. In both cases the user has to revise and disambiguate the current decision in an *Error Dialog*, depicted in Figure 5.25.

The error dialog allows previewing the decision outcome with the help of a *quick check functionality*. Its purpose is to anticipate the $\mathscr{CS}^+$ and $\mathscr{CS}^-$ outcome without applying the decisions to the classification model learner.

If this kind of error occurs multiple times the decision support system suggests enabling an auto-highlighting functionality that keeps track of the annotation sets and holds them in a consistent state.

Handling Missing Decisions

Missing decisions can occur whenever the same scatter plot is shown multiple times in one presentation set and the user marks a scatter plot as relevant, respectively irrelevant, but does not apply the same choice on the second occurrence of the scatter plot. Multiple occurrences can happen, e.g., when applying sampling-based representation finding approaches on a small data set. In case of Min-/Max sampling, multiple presentations of the same item are even likely and cannot be ruled out.

The decision support system keeps track of these missing values and fills them automatically to retain a consistent learning model.

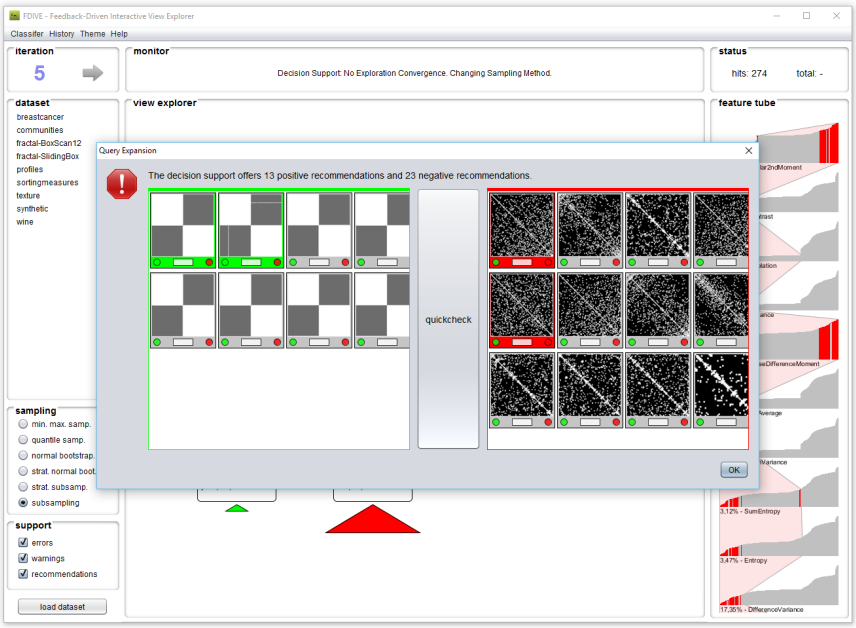


Figure 5.26 Additional meaningful decisions can be recommended to the user by retrieving the most similar matrix plots to the already relevant, respectively irrelevant, annotated views.

Recommending Additional Decisions

The decision support system is able to do more than a mere failure handling. If the user is satisfied with her/his relevance feedback in one iteration, the intra-presentation set similarity to the positive and negative examples is calculated. If the similarity for an unannotated scatter plot is high to one of the items in $\mathscr{AS}^+$ or $\mathscr{AS}^-$ it becomes an annotation candidate for the respective annotation set. More specifically, we are calculating for every view in $\mathscr{AS}^+$ and $\mathscr{AS}^-$ a ranked list of similar views from $\mathscr{ES}$. We are using the Euclidean distance on the Scagnostics feature vectors for calculating the similarity score. These ranking lists are unified for each annotation class by taking into account (a) a minimum similarity threshold –since we want to show only highly similar views– and (b) the potential reoccurrence of one view in the candidate lists –since we want to eliminate duplicate candidate views. The outcome of including annotation candidates into $\mathscr{AS}^+$ and $\mathscr{AS}^-$ are presented to the user in the *query-expansion dialog* shown in Figure 5.26. Here again, the user has the functionality to anticipate (quick-check) the results without applying them to the model learner.

Exploration Set Expansion

Another decision support system functionality aims at assessing the search stability and convergence. For example, if the decision tree classifies more than 50% (user-parameter)

of $\mathscr{ES}$ as irrelevant in the first iteration a great variety of potential patterns may be lost. If the selected parameter is exceeded, the decision support system evaluates the *classification certainty* by comparing all irrelevant classified items ($\mathscr{CS}^-$) to the items in the annotation set $\mathscr{AS}^+$ and $\mathscr{AS}^-$. We construct a certainty ranking for all items in $\mathscr{CS}^-$. Again, we are using the Euclidean distance on the Scagnostics feature vectors for our scatter plot dataset to calculate the ranking score. The subset of items in $\mathscr{CS}^-$ that have a distance higher than an adaptive threshold are treated as uncertain classification decisions and may be taken again into the exploration set $\mathscr{ES}$ for a further refinement. The certainty threshold increases with the number of feedback iterations. In other words, the fewer decisions have been taken by the user the less uncertainty is accepted. Figure 5.27 shows the exploration set expansion dialog.

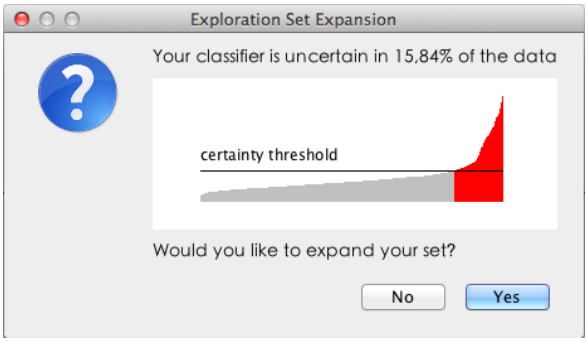


Figure 5.27 The classification system's certainty is assessed in the background. A histogram view shows the number of (un-)certain decisions. If the classifier eliminates a great variety of scatter plot patterns (red bars), the user may decide to retain uncertain scatter plots. Uncertain decisions correspond to the scatter plots whose distance to the negative annotated set is larger than an adaptive certainty threshold.

Adaptive Choice of a Descriptive Feature Vector

Another important decision support system functionality aims at assessing the descriptiveness of the currently chosen feature descriptor wrt. user's current exploration phase. Therefore, we developed an adaptive feature descriptor exchange functionality, i.e., during all times the modeling of the examined items/visualizations may be exchanged with a more descriptive/discriminative FD. We are calculating a FD "quality ranking", as depicted in Figure 5.28.

We derive a quality ranking from the first annotation round, but adapt to the user decisions in later exploration phases. For every feature descriptor an average pairwise distance between all relevant annotated items and all relevant annotated items ($\mathscr{AS}^+$ and $\mathscr{AS}^-$)is calculated. The basic intuition is that, the higher the average difference of the pairwise distances is the more levels of differentiation/discriminativeness can be modeled by the feature descriptor.

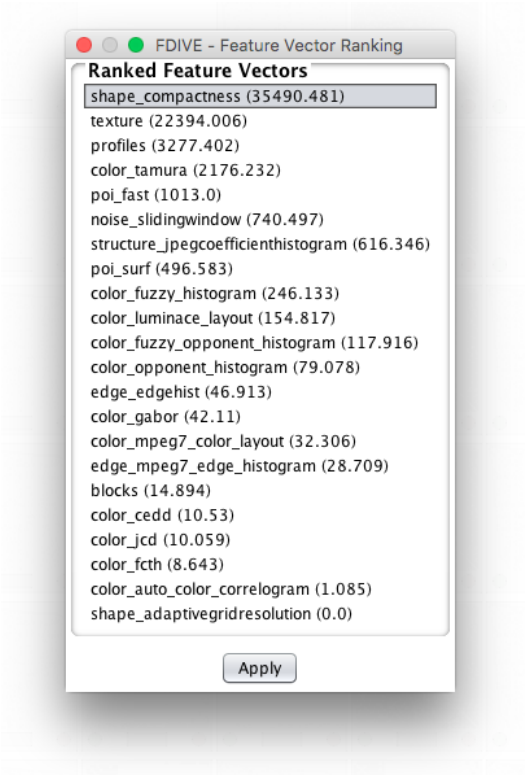


Figure 5.28 A list view of the Feature Descriptor Ranking: Feature descriptors with a high average pairwise distance between all relevant and irrelevant annotations are considered as more discriminative wrt. the feature modeling.

5.8 | Research and Application Context Work

We will next demonstrate the usefulness of our developed visual analytics approaches in several applications contexts.

5.8.1 | Usage Case Demonstration of our User-Steerable Iterative Matrix Reordering

To present the applicability of our user-steerable matrix reordering approach we showcase two exemplary scenarios. In the linear arrangement improvement scenario we show how we take advantage of the 2D projection to improve a state-of-the-art matrix ordering algorithm. In the second scenario, we will show how we use our approach to gain a better understanding of matrix substructures in a real-world scenario.

Linear Arrangement Improvement

Figure 5.8a and Figure 5.8b showcase the applicability of our approach by interactively improving the results of the matrix ordering algorithm "Multi-Scale", developed by Koren and Harel [KH02b] in 2002. The algorithm's results represent a baseline for our approach, since it has the target of minimizing the linear arrangement of a matrix. Furthermore, it is accepted as the state-of-the-art algorithm for the problem.

In our exemplary matrix reordering experiment, we are investigating the `GD95c` matrix from the J. Petit test suite [Pet03] developed for evaluating matrix ordering algorithms. As Figure 5.8a depicts, the initial ordering of the matrix with the multi-scale algorithm results in a linear arrangement (LA) score of 834. From a visual perspective we can see a combination of several *vertex group* and one *star edges* pattern in the upper right part of the plot (cf. Section 5.4.2), which are an appropriate starting point for a reordering optimization. We select the vertices that are grouped in 2D, but are still connected via rather long edges to other groups of vertices. This is also the reason, why the dense group at the bottom of the 2D projection visualization is a rather bad starting point for an optimization, because it is highly interconnected within its own group and is only connected via one edge to the rest of the vertices. As Figure 5.8b depicts, clicking on the *local reordering* button allows us to preview the local reordering results of all available algorithms. After randomly several of the suggestions that all lead to an LA improvement between 1.44% and 3.66%, we chose the topmost submatrix ordering thumbnail. The details panel in Figure 5.8b reveals that a local optimization of this subgroup with the *multi-heuristic* algorithm improved the linear arrangement from 834 to 787 LA.

In this experiment a non-expert user incrementally decreased the LA by 5.64% within four reordering interactions: one global sorting (multi-scale algorithm) and three local sorting suggestion choices. Whenever a group of vertices is selected the user can switch

between the local sorting suggestions by clicking on the matrix sorting thumbnails (cf. Section 5.4.4). Generally, not all vertex selections allow reducing the LA. But, we found in our experiments that the visual patterns described in Section 5.4.2 show to be beneficial starting points.

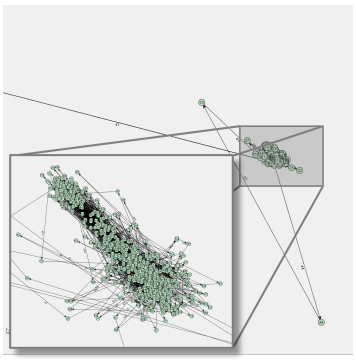
Improving User Understanding

The Figure 5.29 (a) to (f) show us a real-life scenario: Here, a large, dense 1000×1000 matrix is ordered using the multi-scale algorithm. The data results from a similarity calculation between domain names of an IP network and would be used as part of the input for a graph-based clustering algorithm. Reviewing a matrix representation of the similarity calculation is an important intermediate step to assess the performance of the subsequent graph-based clustering. Figure 5.29a shows us an overview of the 2D projection and the applied multi-scale ordering. In Figure 5.29d we see the corresponding matrix image. Clearly, the matrix ordering result is inappropriate and substructures are not visible. Other matrix ordering algorithms show a similar result (Figures are omitted).

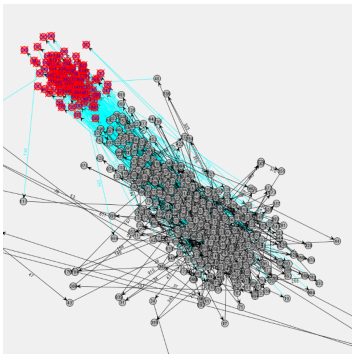
In this specific case, the matrix density is so large that substructures cannot be extracted without compromising the global quality criterion (minimum linear arrangement). However, a manual steering of the ordering algorithm allows hidden patterns to be found. Figures Figure 5.29b and Figure 5.29e show the brushing and linking selection of a locally dense subgroup of vertices. This subgroup stands out, because it is highly connected to a larger subgroup of vertices (main data distribution), but still stands on its own in the projection space. While the selection of matrix rows/columns the image space seams arbitrary (cf. Figure Figure 5.29e), applying the multi-scale algorithm *locally* on this group reveals a dense submatrix that is connected to other matrix members. When taking all matrix data items into consideration matrix ordering algorithms have severe problems finding a trade-off between global and local optimizations. This becomes obvious in Figure Figure 5.29d. Steering the ordering transformation more towards the local aspect enables a fine-grained investigation of the globally dense matrix. Figure 5.29f shows the end result of the local ordering steering.

5.8.2 | Use Case Demonstration of our Projection-based Similarity Definition and Adaption

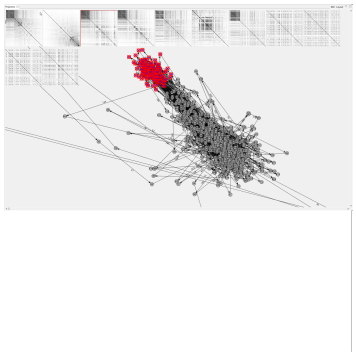
We now show the use of the Matrix Projection Explorer to analyze various data sets, demonstrating its applicability to potential analysis cases.



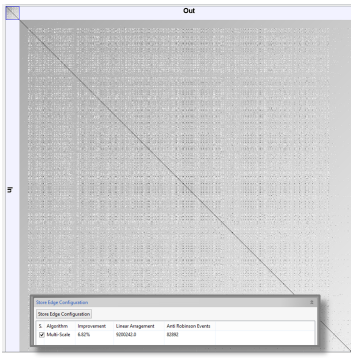
(a) Overview of the 2D Projection of a large, dense 1000 × 1000 matrix sorted with the multi-scale algorithm.



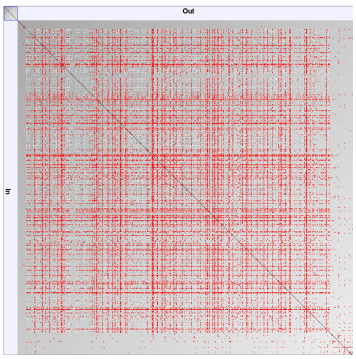
(b) The zoomed 2D Projection shows a large, dense group (cluster) of vertices with one subcluster.



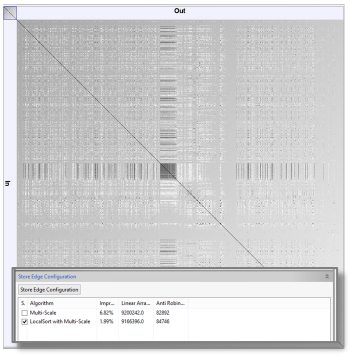
(c) The upper dense subgroup in the cluster is sorted with a local submatrix sorting.



(d) The matrix image representation of the multi-scale sorted matrix data.



(e) The cluster's substructures are revealed by applying local sorting optimizations.



(f) The user can steer the submatrix reordering process by choosing the most appropriate local sorting decision.

Figure 5.29 In cases where established matrix sorting algorithms do not lead to a satisfying sorting result, as depicted in (a), a manual steering intervention can help to reveal hidden matrix substructures. In our case, we are able to extract a locally dense submatrix from the globally sparse matrix. This transformation leads to a linear arrangement decline, which is inevitable to be able to reveal this globally highly interconnected pattern.

Application to Dynamic Computer Network Analysis (VAST 2013 Challenge Dataset)

Monitoring large computer networks is a challenging, but a highly important task for network operators. While there are many tools to plot and explore time-series for single hosts and network ports, most tools lack an overview of the whole computer network independent of the number of underlying hosts. We use the VAST Challenge 2013 dataset [CGW13] to demonstrate the usefulness of our tool on a realistic dataset in a network security scenario. In a fictive enterprise situational awareness scenario, network monitoring operators in a special control room should be able to keep track of important events in their global computer network. The network consists of several hundred thousand computers and grows over time. We preprocessed the data to fit our matrix-based data model, imported the first week of NetFlow traffic of Mini Challenge 3 into our system and aggregated the dataset hourly. Each one-hour interval is represented as a matrix in Figure 5.30, which conveys the number of bytes transferred from any source IP address to any destination ports. Obviously, the matrix sizes are quite diverse, because in each time interval there might be a different number of computers and even a different number of active destination ports. These common properties make the dataset a challenge to analyze and visualize in an overview for a long period.

The interactive overview visualization is shown in Figure 5.11. It depicts the overall distance of aggregated connections active for each hour. Each matrix cell represents a distance value between two different hours.

At a first glance, several horizontal and vertical lines stick out as patterns. They represent individual hours, which have highly different underlying connection matrices compared to all other points in time. The event on 2013-04-02 06:00 to 06:59 is indeed a denial-of-service (DoS) attack to one of the company's web servers, which is confirmed in the VAST Challenge scenario. Another confirmed DoS attack is clearly visible on 2013-04-03 around 11:00. Such events have a high impact and are outstanding in this overview, because they are highly different from normal network behavior.

More subtle patterns are visible in the visualization as well. There are several rectangles (2013-04-01 08:00 until 16:00, 2013-04-03 16:00 until 2013-04-04 05:00), which are almost black, representing time intervals with quite homogeneous connection matrices. The insight can be confirmed, that the traffic in those hours seem to be quite similar. Further analysis reveals that during the mentioned time period less unique IP addresses are involved than usual. A possible reason could be a crashed server during the aforementioned DoS attack. However, such hypotheses can only be confirmed by taking further log data into account, because the NetFlow data does not provide enough detail to answer such questions.

Interestingly, there are at least three very visually outstanding hours (2013-04-05 10:00, 2013-04-06 06:00, 2013-04-07 06:00), similar to those hours in which DoS attacks occurred. These should be investigated further by the analyst. For this dataset a modification of

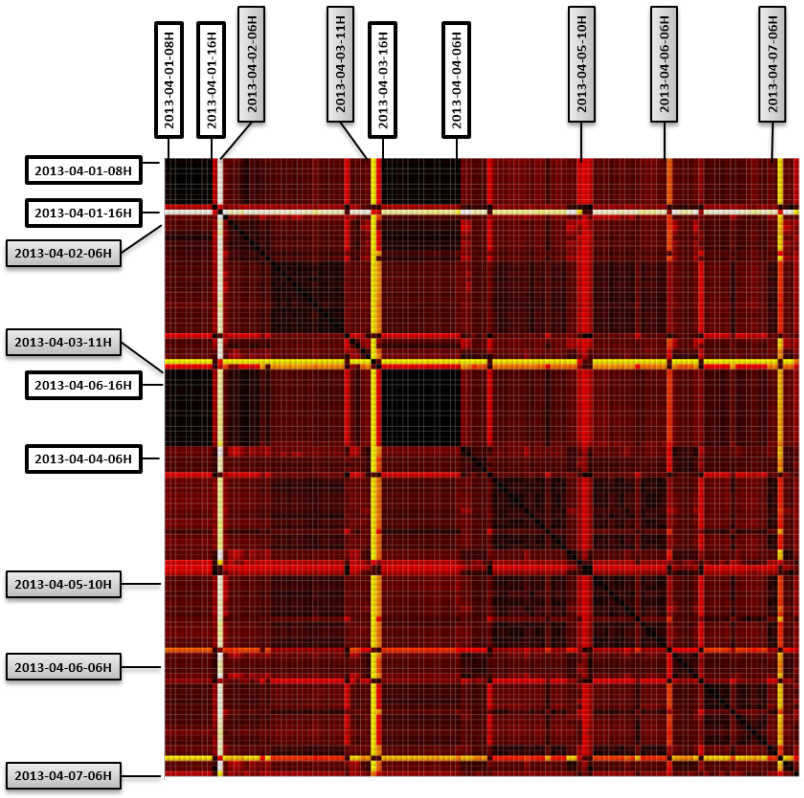


Figure 5.30 A distance meta-matrix of all pairwise matrix distances for the VAST Challenge 2013 dataset with 120 matrices. Patterns, like closely related (dark groups) and outlying (light rows) matrices, stand out visually and can be interpreted in this use case as Denial-of-Service Attacks.

the distance penalty calculation proves to be useful. As Figure 5.14 shows, changing the Penalty calculation to `ZeroPenalty` visually boosts areas of homogeneity, while applying the `MaxDistSquare` penalty function gives outliers a higher visual impact.

Overall, the visualization does support the identification of behaviour-related patterns. The main advantage is a compact, but dense overview, based on the notion of matrix similarity for different points in time, which proved to be quite effective for the VAST Challenge 2013 dataset.

Application to Soccer Game Analysis (DEBS 2013 Challenge Dataset)

As a second example, we consider the ACM International Conference on Distributed Event-Based Systems (DEBS) *soccer game analysis* challenge dataset from 2013 [JZ13]. The dataset originates from a real-time movement tracking system deployed on a football field of the Nuremberg Stadium in Germany. For the duration of a game, real-time wireless sensors embedded in the player's shoes and the ball recorded the respective positions as a function of time. To transform the data into matrices, we extracted a distance matrix

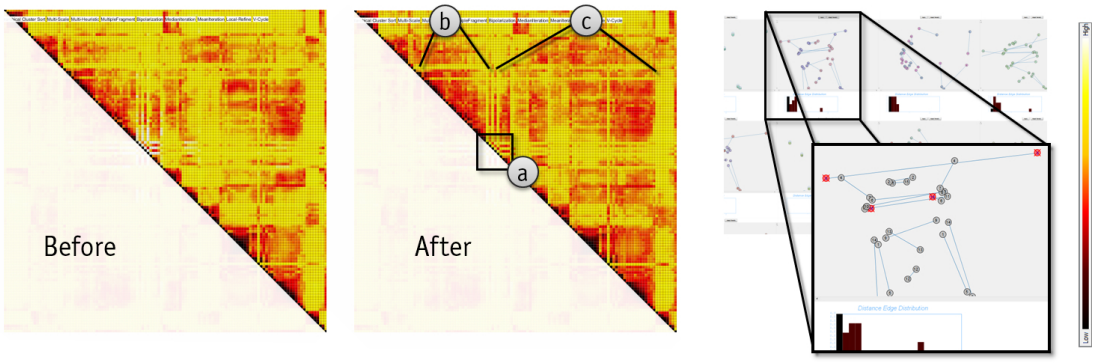


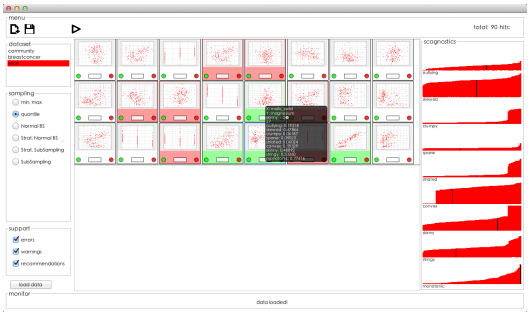
Figure 5.31 Excluding vertices from the simililarity calculation helps to filter out aspects of low importance. In this soccer analysis task it makes sense to exclude goal-keepers to find semantically similar game situations, where goal-keepers have a low impact.

between all players for every second of the game; i.e. a set of 4,126 matrices depict every 1-second game situation. Every row, respectively column, in the matrices corresponds to one player and every matrix cell represents this player's distance to another player at that specific time instance of the game. Figure 5.31 shows a range of situations during the second half in the meta-matrix overview. The view is sorted by time and shows similar game situations adjacent to each other. We can see that two patterns occur: (1) The pattern [a] depicts a corner-kick (2) the diffusion-like pattern [b] refers to a goal-kick.

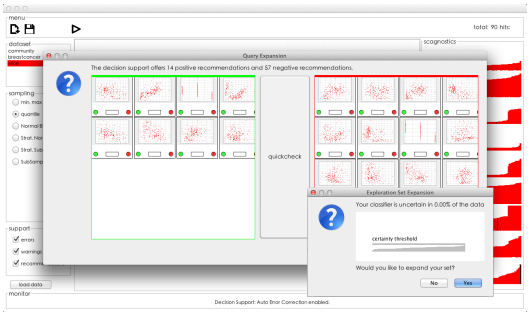
One possible task could be to find similar game situations in a soccer game, such as corner kicks, goal-kicks, or shots at the goal. Other tasks are exemplified in [PVF13]. Most of these tasks tend to be related to fuzzy queries, in the sense that player positions may change between semantically similar situations. Also, certain players may not even be important for a game situation (e.g., the offensive goalkeeper is normally less important for a corner kick if not attacking directly). In these cases it seems appropriate to exclude data vectors from the calculation, whenever they have no impact on the task at hand. This typical work flow is shown in Figure 5.31, where the user zoomed into an area of a known corner-kick (left) and selects vertices to be excluded from calculation (middle). The influence is shown in the adapted overview (right), where the area [c] changes significantly.

5.8.3 | Usage Case Demonstration of our Feedback-Driven View Exploration

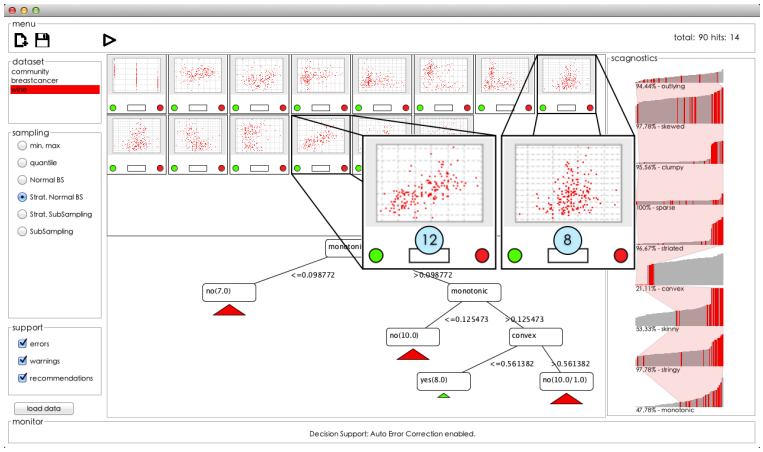
To present the applicability of our View Space Explorer we showcase two exemplary exploration scenarios. The first scenario uses the well-known, real-life numeric *Wine* dataset from the UCI Machine Learning Repository [BL13]. We chose the wine dataset,



(a) In the first annotation round the user selects four relevant and six irrelevevant scatter plots.



(b) The decision support recommends adding a range of similar scatter plots to both annotation sets; After the classifier is trained all decisions are judged as certain.



(c) The final result shows several data correlations. E.g. The amount of ash and flavoid influences a wine's color.

Figure 5.32 Finding correlations in the Wine data set; After three annotation iterations the exploration set with initially 90 scatter plots is reduced to 14 scatter plots, containing the annotated data correlations.

because of its low number of scatter plot axis combinations and thus its understandability. This allows us to focus on the decision support's interventions in the exploration process.

The second dataset focuses on matrix pattern retrieval and is thus significantly more complex. However, it allows us to focus –next to the presentation of (hidden) patterns– on the convergence development like it is necessary for information retrieval systems. We conducted our case studies with Ph.D. and Master students from the computer science area.

Wine Data Set

In the first case study we are searching for correlations in the scatter plot axis combinations. The exploration of data correlations is challenging, since users must be able to describe the correlation to be found. In the feedback-driven view exploration approach we assume that users do not know in advance, which (Scagnostic) feature might be beneficial for the current question. Query formulations like "Find correlations in the data" are normally not possible with most query formulation systems. Interactive retrieval systems, such as query-by-sketch interfaces, might only result in positive correlations, while negative correlations might be interesting, as well. Pure metadata searches will fail to allow these searches. Therefore, a feedback-driven semi-automatic retrieval approach can be beneficial.

The Wine data set results from a chemical analysis of 178 wines sub-categorized into three classes (red, white, rose wines). In total the dataset comprises 14 numeric continuous attributes, such as alcoholic strength, color intensity, or magnesium. We derived 90 axis combinations and converted them to scatter plots -short SPs- with 178 data points each.

Figure 5.32 shows us a sequence of interactions on the wine dataset. In the first iteration (Figure 5.32 (a)) the user annotated four relevant and six irrelevant SPs from 24 initially presented SPs. The presentation set $\mathscr{RS}$ results from a normal bootstrapping sampling of the initial 90 SPs. A review of the four relevant annotated SPs shows that they share a high *monotonic* value. Satisfied with the first annotation round the user clicks on *Apply* and sees a dialog, in which the decision support recommends adding 15 SPs to $\mathscr{AS}^+$ and 68 SPs to $\mathscr{AS}^-$ (Figure 5.32 (b) background). The user declines both recommendations. Hence, the classifier is trained on the initial annotation set and results in 27 positively and 63 negatively classified SPs. A review of the classification uncertainty shows that all decisions were certain (Figure 5.32 (b) front). Thus, the user can assume that no SP is lost due to a misclassification. The feature tube reveals that the highest concentration is in the monotonic value. It appears to be beneficial to choose SPs with a high monotonic value for this task. In the second feedback round the user selects seven relevant and six irrelevant SPs and the decision support recommends adding one relevant and four irrelevant SPs. Once again the user declines all recommendations. The subsequent classification results –again– in the same 27 positively and 63 negatively classified SPs. No more exploration progress is apparent and thus the user will see the same SPs in each subsequent feedback round. Accordingly, the decision support switches the sampling method to "Stratified bootstrapping". In a final annotation round six relevant and five irrelevant SPs are chosen and the exploration finishes with 14 SPs.

In the 14 result SPs semantically interpretable data patterns become visible (Figure 5.32 (c)). For example, the SPs 8 and 12 show that the wines' color is positively correlated with the amount of ash and the amount of flavonoids. A meta research reveals: "Flavonoids are antioxidant compounds found in plants, as well as tea, red wine and chocolate, ..." [Hop14].

Matrix Pattern Retrieval and Interpretation

Our second use case focuses on the retrieval of matrix patterns. We are using the Matrix Reordering dataset, as presented in Section 2.3. The dataset contains 4,313 matrix plots generated by 35 matrix reordering algorithms on 150 distinct graph data sources.

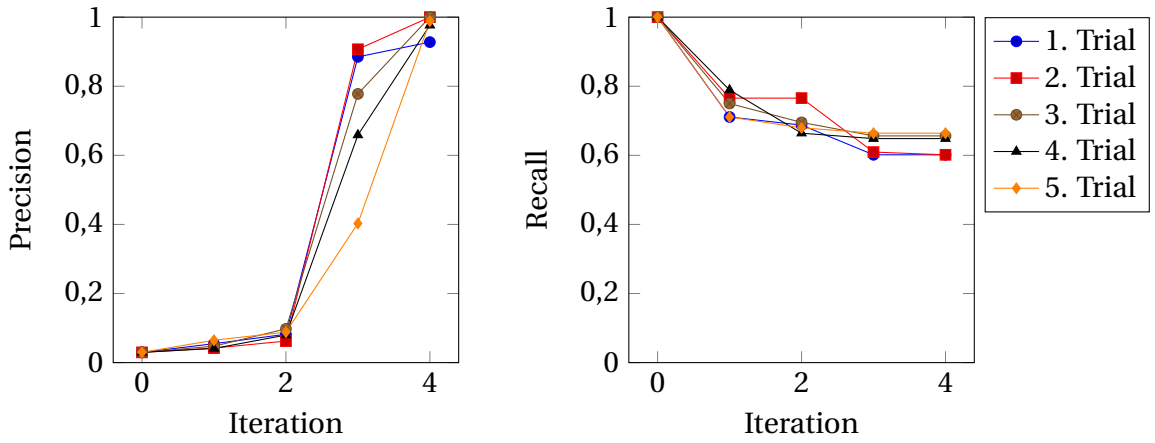
In order to demonstrate our feedback-driven analysis mechanisms we are primarily focusing on the retrieval of a specific (anti-)pattern of interest: the bandwidth anti-pattern, as depicted in Figure 5.33. We found in our prior studies that matrix reorderings generated with the Cuthill–McKee algorithm tend to produce this pattern. Alternative pattern exploration goals could be to extract only the matrix plots containing diagonal blocks (highly connected components in the graph).

Figure 5.33 Cuthill-McKee's Bandwidth anti-pattern.

We have generated the Matrix Reordering dataset in a structured fashion. Hence, we know a priori how many plots have been generated with the Cuthill–McKee algorithm and are able to quantify our retrieval performance with the standard precision, recall and F1-score measures. In order to improve our statistical significance we repeat all exploration trials five times. All trials follow the same procedure: In the initial annotation round all bandwidth patterns are selected as relevant, while all other plots are marked irrelevant. Subsequently, a SOM with a regular 3×3 is trained and recursively refined in a later annotation round, if necessary. The core difference between the annotation rounds is that a more and more fine-grained annotation decision is used. Next we will detail on all annotation rounds separately:

1. In the first iteration (Figure 5.35a) a coarse-grained subdivision into relevant and irrelevant data is carried out. Since the majority of grid cells contains more than 300 elements we are calculating nested SOMs up to a recursion level of two. Cells containing only the pattern of interest are marked entirely as relevant, while all other cells are marked as irrelevant.
2. In the second (Figure 5.35b) and third annotation round Figure 5.35c) our procedure almost identical, though we are also judging cells with less than 60 elements and a small amount of expected patterns as "still" relevant. This procedure simulates an uncertain –potentially inexperienced- user, who is not willing to sacrifice any retrieval results for a (more) efficient process.
3. In the final and forth iteration (Figure 5.35d) only a few elements have to be examined. Thus, most of the cells can be marked entirely as relevant. In our case, nearly all cells are judged as relevant, thus we can expect that our search has converged.

Table 5.1 shows the iterative search space reduction and refinement over the five experiment runs. On average 83 plots of the 128 existing plots were retrieved. This is also



(a) Precision value development.

(b) Recall value development.

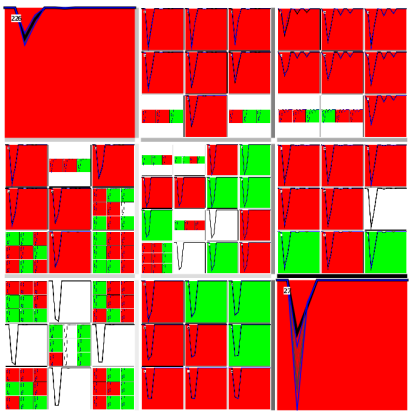
Figure 5.34 Development of the precision and recall values over the five iterations for each of the five experiment runs/trials for the Matrix Pattern Retrieval Experiments.

reflected precision and recall value development depicted in Figure 5.34. The precision value increased significantly after the second annotation round. Here in the trials the decision support system was able to change the FD to a more descriptive choice (shape_compactness). Later the already good precision values get only marginally improved, which can also be seen in the search space reduction (averagely 40.6 more items were classified as irrelevant in this step). The recall values, on the other hand, show that our approach is able to retrieve around 70% of the expected 128 plots. While this seems to be a quite low number, a closer inspection reveals not all of the 128 plots contains a bandwidth pattern. However, a more accurate precision and recall assessment can only be facilitated with a large, consistent and well-annotated pattern benchmark dataset, which remains future work.

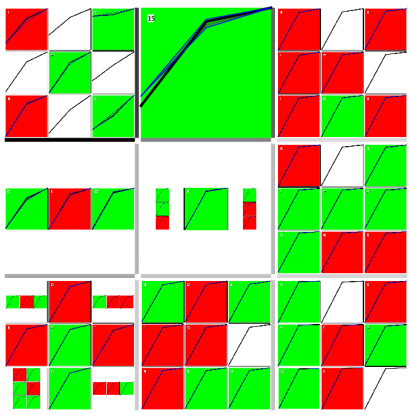
Trial	Iteration				
	0	1	2	3	4
1	4,313	1,660	1,073	87	83
2	4,313	2,329	1,571	86	77
3	4,313	2,020	911	108	84
4	4,313	2,465	1,066	126	85
5	4,313	1,406	986	211	86

Table 5.1 Search space reduction from 4,313 elements to on average 83 relevant elements containing the requested bandwidth pattern. In total the dataset contains 128 plots generated with the Cuthill–McKee algorithm.

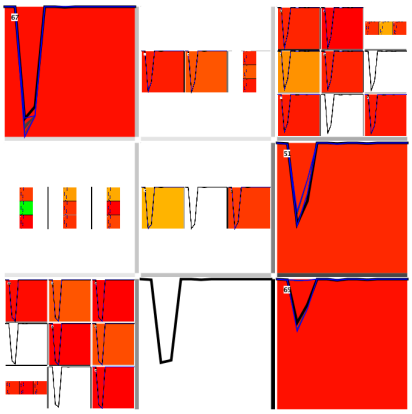
All in all, our feedback-driven view space exploration proves to be useful for matrix pattern retrieval tasks. Moreover, we can see that the decision support system improves significantly the retrieval efficiency. This is due to the fact that potentially wrong decisions are intercepted and –which is even more important– a novel concept of an adaptive changeable feature modeling proves to be beneficial for these exploration tasks.



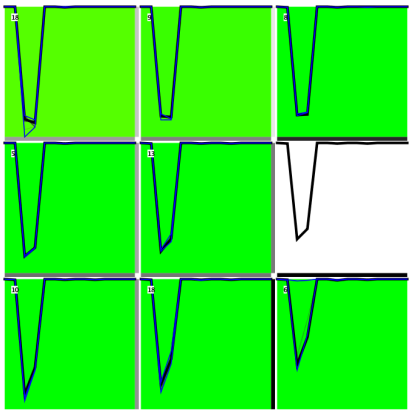
(a) Completely annotated SOM at the end of the first annoation round. The SOM was Initially trained with the Haralick texture feature descriptor. The nested SOM concept is clearly visible. Only a small amount of (nested) SOM cells is marked as relevant.



(b) Second annotation round: The decision support system changed the feature modeling to the shape_compactness FD, as the cell glyph depicts. Significantly more items are marked as relevant in this view.



(c) Third annotation round. Here the decision support changed again to the feature modeling of with the texture FD. As the coloring depicts, a large amount of cells can be annotated with irrelevant (good discriminative FD properties).



(d) Completely annotated SOM at the end of the fourth annoation round. In this step the feature descriptor did not change and the coloring depicts that very few items needed to be marked as irrelevant.

Figure 5.35 Four annotation rounds are necessary for a matrix pattern retrieval use case with the FDIVE pipeline and the nested SOM classifier.

6 | Concluding Remarks and Perspectives

Contents

This final chapter of the dissertation connects all described research approaches to the broader context of their research questions and highlights the individual contributions. Moreover, we conclude with an outline of future and open research questions.

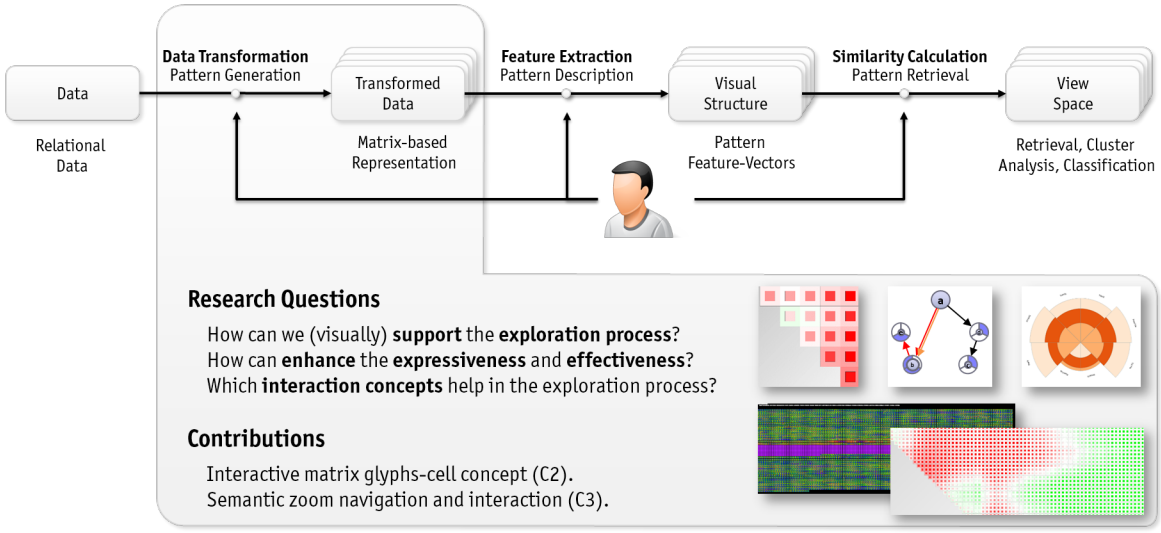


Figure 6.1 Contributions and Research Questions focused in the Dissertation Chapter: "Visual Interactive Support for Exploring Matrix-based Representations"

6.1 | Contributions and Future Perspectives

In the following we will review the contributions of this dissertation separately for each chapter and connect the research questions and -problems to their research achievements. Furthermore, we and pinpoint to open challenges in the field.

6.1.1 | Visual Interactive Support for Exploring Matrix-based Representations

The visual interactive support for an exploration of matrix-based representations is focused in Chapter 3. Its core contribution lies in the visual justification and proof of concept that an enhanced cell design for matrices can enhance the analytic expressiveness and usefulness of matrix-based representations.

As Figure 6.1 depicts exemplary, we contribute with a range of novel matrix cell glyph encodings that are developed for their specific problem domain and use case. For example, the ranking and ordering glyph can be used to visually compare a set of feature vector rankings. Sunburst, Inner-Outer rectangle glyphs and text thumbnail glyphs have been developed in earlier works (e.g., [Beh+12c]), but most of them were not applied yet in the matrix layout setting. Another contribution is the interactivity aspect of these glyphs. While earlier glyph matrices were static depictions we extend the idea to make interactive visualizations available in the matrix cells, which have the power to modify the appearance of the entire (overview) matrix plot.

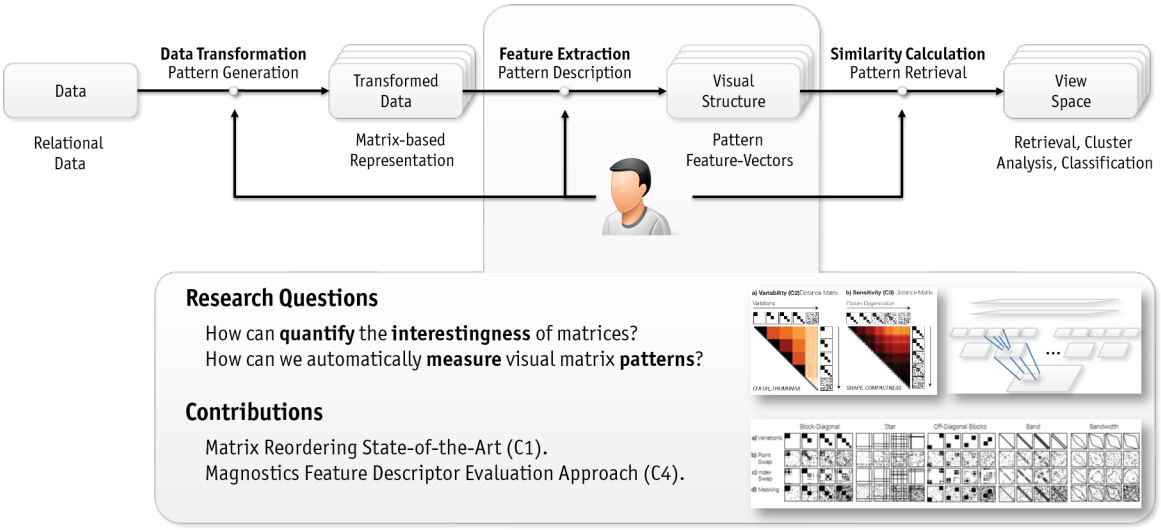


Figure 6.2 Contributions and Research Questions focused in the Dissertation Chapter: "Automatic Support for Pattern Retrieval in Matrix-based Representations"

In a broad context, this work contributes to the idea of a GPLOM (Generalized Plot Matrix), which was partially formalized by Im, McGuffin, and Leung in [IML13] after the first published glyph matrix approaches have appeared. The idea is to couple the good visual scalability properties of matrices with the expressive power of glyph designs. While it was earlier accepted that matrices are just beneficial for depicting large amounts of numerical relationships we can now see that the expressiveness and effectiveness of matrix-based representations can be used extended significantly with glyph representations.

Learning from the outcome of this chapter, the following main directions are indicated for future research:

- First work has been conducted by Ghoniem, Fekete, and Castagliola in [GFC04] to empirically examine the perceptional properties of matrix-based representations. Further research should focus on how much information density can be added to matrix visualizations until they reach the mental capacity level of the user, i.e., get overly complex.
- Another line of research should focus on navigation and interaction aspects in GPLOMs. The combination of Off-Screen visualization approaches and semantic zoom may be one specific idea.

6.1.2 | Automatic Support for Pattern Retrieval in Matrix-based Representations

One of the primary chapters of this dissertation focuses on the automatic analysis of matrix patterns. Therefore, we develop and contrast in Chapter 4 several approaches for an automatic quantification of matrix patterns.

As Figure 6.2 depicts, we pose the research question "How can we quantify the interestingness of matrices?" Therefore, it is necessary to examine the entire process that leads to the final matrix plot, which might be interesting, respectively uninteresting, due to the visual and interpretable matrix patterns it contains. As a direct consequence we examined in [Beh+16b] the state-of-the-art for matrix reordering algorithms with the goal to reason why specific algorithmic approaches tend to produce specific visual patterns. The core contribution of this chapter lies in the quantitative performance evaluation of feature extraction methods for the suitability to detect matrix patterns. With the presented feature descriptor evaluation approach MAGNOSTICS, a pattern-driven visual exploration of becomes feasible for the first time.

Learning from the outcome of this chapter, the following main directions are indicated for future research:

- A novel class of matrix reordering approaches can be derived from MAGNOSTICS. Matrix reordering algorithms need two prerequisites: (1) an efficient enumeration approach to iterate the enormous search space and (2) a quality criteria to compare two permutation solutions. With MAGNOSTICS the user can express that he/she is interested to see only matrices with this specific pattern, since this pattern might be well-interpretable in the current context.
- Another line of research should focus on the development of further –potentially perception inspired– feature detector approaches to quantify the existence of matrix patterns.

6.1.3 | Visual Analytics for Pattern Retrieval in Matrix-based Representations

Visual analytics strives to combine the expressiveness of visualizations with the interpretation capabilities of humans. Therefore, we show in Chapter 5 several approaches, which allow the user to expose his/her knowledge to the computer system and thus enables the user to steer the otherwise purely automatic decision process.

The core contribution of this chapter is to bridge the gap between the fully automatic approaches (as shown in Chapter 4) and the exploration and navigation approaches (as shown in Chapter 3). For example, in Section 5.4 an approach to steer interactively a matrix reordering algorithm is shown. These algorithms were up-to-now black-box

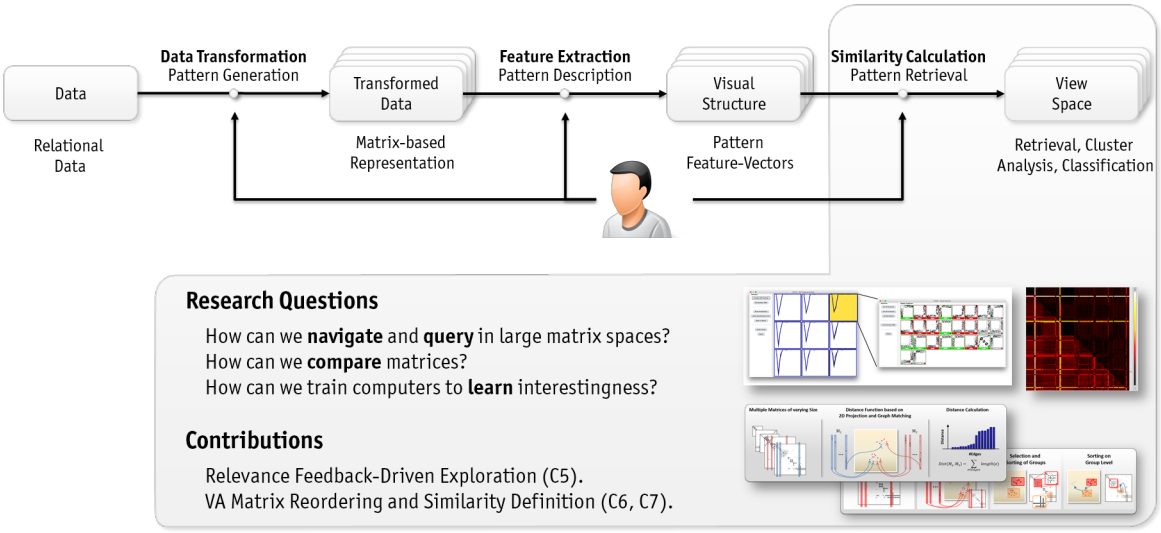


Figure 6.3 Contributions and Research Questions focused in the Dissertation Chapter: "Visual Analytics for Pattern Retrieval in Matrix-based Representations"

algorithms. However, with our novel approach the user can visually inspect and modify the algorithmic process.
An interactive system for the definition and adaption of similarity is another contribution of this thesis. Generally, the concept of an interactively adaptable similarity defintion is novel and has a direct impact on ranking, clustering and classification results [Ber+14a; Ber+14b]. With our approach we can interactively steer the similarity notion for the specifically challenging use case: the comparision of matrices of varying dimensionality, such as occurring in distinct snapshots of a peer-to-peer network.
The last, but most impactful, contribution of this chapter focuses on the user-guided learning of interestingness in an exploration process. While Overview+Detail approaches try to guide the user visually to areas of interest, our relevance feedback-driven approach shows in a structured fashion a representative set of potentially relevant views. The user rates these views and trains an adaptive classifier that iteratively learns from the user's decisions. A visual analytics-driven innovative decision support system keeps track of the exploration path convergence by adapting the algorithmic procedures to reach a better precision/recall.

Learning from the outcome of this chapter, the following main directions are indicated for future research:

- Feedback-driven exploration proves to be powerful, but may be combined with classical Overview+Detail systems. These adaptive Overview+Detail systems will change their overview according to relevance/interestingness decisions that the user communicated to the system in one of the drill down steps.
- Interactive, steerable and progressive algorithms and their visualization should be researched more intensively in the future. Related to this aspect entirely new evaluation approaches have to be established that are able to track the time- and interaction-dependent aspects of an exploration convergence.

6.2 | Concluding Remarks

This dissertation set out from initial question: "How can we explore effectively and efficiently large amounts of potentially multi-variate and/or time-dependent relational data?".

During the course of my Ph.D. I found that this question has to be answered with regards to the number of relational data items, which should be explored simultaneously. While in the single matrix analysis scenarios the depiction of relationships is in the focus, a multi matrix analysis scenario focuses on the retrieval of interesting and interpretable visual patterns.

For the single matrix analysis we followed the research direction to combine the recent advances in the Information Visualization domain and the high scalability of matrix-based visualizations. Specifically, we contribute novel matrix cell glyph encodings, with the aim to boost the expressiveness of matrix-based representation. Glyph-based representations help in this case to explore and depict multi-variate relationships.

On the other hand, we intensively conducted research with the aim to quantify the existence of visual patterns for multi-matrix analysis scenarios. Retrieving visual patterns is a complex task, but at the same time it can be seen as a cornerstone for a fundamentally new exploration process: a pattern-driven exploration of (relational) data. In this process the user is (visually) confronted with the existence of salient visual patterns and can focus on relating and interpreting these patterns in the current domain specificities and with regards to his/her prior knowledge.

All in all, this dissertation demonstrates several approaches combining visualization, automatic analysis, and interaction for exploring large amounts of relational data. The complexity and amount of relational data requires for a scalable, expressive and effective visualization technique, which we found in matrix-based representations. The strengths of interactive matrix-based representations combined with automatic pattern analysis and -generation processes allows discovering interesting, unknown facets in these data sets. Many interesting and practically relevant research questions encourage the expectation that more researchers contribute to the field of matrix research in the upcoming years.

List of Figures

List of Tables

List of Algorithms

References

Authored and co-authored publications

[Beh+14a] Michael Behrisch et al. “Feedback-Driven Interactive Exploration of Large Multidimensional Data Supported by Visual Classifier”. In: *Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on*. IEEE CS Press, Oct. 2014, pp. 43–52. DOI: 10.1109/VAST.2014.7042480 (cited on pages 5, 7, 112, 117, 154).
Contribution Clarification: Conceptualization and Contributions: The main research focus of the paper is to develop a classifier-supported, semi-automatic exploration process for the exploration of very large view spaces. I initiated the project and developed the iterative feedback exploration framework idea. I had the idea of a user-supervision system that tracks the user's exploration process. Implementation/Instantiation effort: All code was developed by F. Korkmaz. I only contributed conceptionally and helped writing the decision support system. Authorship: I wrote/edited most of the text myself. F. Korkmaz contributed most of the system screenshots and wrote the Case Study section 7. L. Shao, T. Schreck and I worked jointly on the structure, textual revision, the didactical concept of the paper. Supervision: T. Schreck supervised the paper project and commented on the contributions.

[Beh+16a] Michael Behrisch et al. “Magnostics: Image-based Search of Interesting Matrix Views for Guided Network Exploration”. In: *Visual Analytics Science and Technology (VAST), 2016 IEEE Conference on*. IEEE CS Press, Oct. 2016, pp. 43–52. URL: magnostics.dbvis.de (cited on pages 7, 110).
Contribution Clarification: Conceptualization and Contributions: The main research focus of the paper is to identify image feature-descriptors that are able to quantify the presence/absence of visual patterns in matrices. The project was initiated jointly with T. Schreck. We developed together the idea to explore, next to the data-space, also the image space for a pattern quantification. L. von R[[[ERROR FOR PACKAGE inputenc]]]den and B. Bach contributed to the conceptualization. Implementation/Instantiation effort: B. Bach and I jointly developed the evaluation code. I conducted most of the experiments and collected/migrated the feature descriptors into one evaluation framework. L. von R[[[ERROR FOR PACKAGE inputenc]]]den also contributed FDs (Noise-Dissimilarity, Gradient). Some of the novel FDs (Block FD, Statistical Noise FD) were developed by students (M. Delz, Bianca Orita, Manuel Hotz and Raffael Wagner), who signaled/documented that their code may be used in my publications. Authorship: I coordinated all paper writing

efforts and wrote/edited most of the text myself. B. Bach and I worked jointly on the structure, textual revision, the didactical concept of the paper. M. Hund contributed the text to Section 5.2 on Pattern Variability, T. Schreck contributed Section 2.1 and 2.2 (Related Work). B. Bach contributed Section 7.2 (Dynamic Network Analysis). The ideas for the mentioned chapters were developed in joint effort. I developed the website magnostics.dbvis.de. Supervision: J.-D. Fekete and T. Schreck contributed with textual revisions, supervised the paper project and commented on the contributions.

[Beh+16b] Michael Behrisch et al. "Matrix Reordering Methods for Table and Network Visualization". In: *Computer Graphics Forum* 35.3 (June 2016), pp. 693–716. ISSN: 1467-8659. DOI: 10.1111/cgf.12935. URL: http://dx.doi.org/10.1111/cgf.12935 (cited on pages 5, 20, 117, 145, 164, 174, 175, 214).
Contribution Clarification: Conceptualization and Contributions: The main research focus of the paper is to survey the matrix reordering algorithm landscape. The project was initiated by J.-D. Fekete. I developed the idea to enhance the focus towards a algorithm design impact. With this idea the research question changed to "Which algorithm designs tend to produce which patterns?". I contributed the pattern-related discussions. Implementation/Instantiation effort: B. Bach and I jointly developed the evaluation schema. I conducted most of the experiments and collected the algorithms. Authorship: I conducted the primary literature research, collected and summarized the main approaches in the field. Many of the references were given by J.-D. Fekete and N. Henry Riche. B. Bach and I worked jointly on the structure, textual revision, the didactical concept of the paper. J.-D. Fekete contributed the Background section. Supervision: J.-D. Fekete, N. Henry Riche and T. Schreck contributed with textual revisions, supervised the paper project and commented on the contributions.

[Beh+12b] Michael Behrisch et al. "Matrix-Based Visual Correlation Analysis on Large Timeseries Data". In: *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*. Institute of Electrical & Electronics Engineers (IEEE), Oct. 2012, pp. 209–210. DOI: 10.1109/VAST.2012.6400549. URL: http://dx.doi.org/10.1109/VAST.2012.6400549 (cited on page 78).
Contribution Clarification: Conceptualization and Contributions: The main research focus of the paper is to develop an approach for the visual, off-set invariant correlation analysis for time-series data. The project was initiated jointly with T. Schreck and J. Davey. I developed the idea to use a semantic zoom interface and the time-series glyph representations. J. Davey and T. Schreck reviewed the ideas and contributed to the conceptualization. Implementation/Instantiation effort: I developed the code and all its components. J. Davey and T. Schreck reviewed and pointed to open questions/missing features. Authorship: J. Davey and I worked jointly on the structure, textual revision, the didactical concept of the paper. T. Schreck reviewed and edited the text. I wrote/edited most of the text myself. Supervision: J. Kohlhammer, D. Keim and T. Schreck supervised the paper project and commented on the contributions.

[Beh+15] Michael Behrisch et al. "Quality Metrics Driven Approach to Visualize Multidimensional Data in Scatterplot Matrix". In: *Eurographics Conference on*

Visualization (poster paper). 2015 (cited on page 110).
Contribution Clarification: Conceptualization and Contributions: The main research focus of the paper is to develop an matrix reordering approach for Scatterplot matrices. I developed the idea to use the image-space is a proxy for the similarity computation between cells. L. Shao and J. Buchm[[[ERROR FOR PACKAGE inputenc]]]ller reviewed the ideas and inspired the discussions and the general framework idea. Implementation/Instantiation effort: I developed all code for the SPLOM matrix reordering approach. Authorship: L. Shao and I worked jointly on the structure, textual revision, the didactical concept of the paper. Supervision: T. Schreck supervised the paper project and commented on the contributions.

[Beh+12c] Michael Behrisch et al. "The News Auditor: Visual Exploration of Clusters of Stories". In: *Proceedings EuroVA International Workshop on Visual Analytics*. Eurographics, 2012, pp. 61–65. DOI: 10.2312/PE/EuroVAST/EuroVA12/061-065. URL: http://dx.doi.org/10.2312/PE/EuroVAST/EuroVA12/061-065 (cited on pages 81, 88, 212).
Contribution Clarification: Conceptualization and Contributions: This paper is not a contribution of my thesis, but has been developed in the course of my master project/thesis. The paper was written during Ph.D. time. The main research focus of the paper is to develop a system, which helps the user to visually compare large text corpora and to retrieve textual similarities. M. Krstajic initiated the project and conceptualized the ideas. Implementation/Instantiation effort: I developed the entire system, the glyph. and interaction design and conducted the experiments. Authorship: M. Krstajic and I worked jointly on the structure, textual revision, the didactical concept of the paper. I wrote/edited most of the text myself. M. Krstajic framed and contributed to the Introduction and the Problem Description Section 3. Supervision: T. Schreck and D. Keim supervised the paper project and commented on the contributions.

[Beh+14b] Michael Behrisch et al. "Visual Analysis of Sets of Heterogeneous Matrices Using Projection-Based Distance Functions and Semantic Zoom". In: *Eurographics Conference on Visualization (EuroVis 2014)*. Vol. 33. 3. The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd., July 2014, pp. 411–420. DOI: 10.1111/cgf.12397. URL: http://dx.doi.org/10.1111/cgf.12397 (cited on pages 7, 33, 78, 96, 133, 155).
Contribution Clarification: Conceptualization and Contributions: The main research focus of the paper is to develop a visually interpretable and visually steerable approach for the comparison of matrices of potentially varying sizes. The project was initiated jointly with T. Schreck and J. Davey. I developed the idea to use the projection-space for the comparison and the similarity-steering. J. Davey and T. Schreck reviewed the ideas and contributed to the conceptualization. Implementation/Instantiation effort: I developed the Matrix Projection Explorer Framwork with all its components. J. Davey and T. Schreck reviewed and pointed to open questions/missing features. Authorship: I coordinated all paper writing efforts and wrote/edited most of the text myself. J. Davey and I worked jointly on the structure, textual revision, the didactical concept of the paper. F. Fischer and O. Thonnard contributed the Section 6.1 on the VAST Challenge Use Case. T. Schreck reviewed

and edited the text. Supervision: J. Kohlhammer, D. Keim and T. Schreck supervised the paper project and commented on the contributions.

[Beh+13] Michael Behrisch et al. "Visual Comparison of Orderings and Rankings". In: *EuroVis Workshop on Visual Analytics*. Ed. by M. Pohl and H. Schumann. The Eurographics Association, 2013. DOI: 10.2312/PE.EuroVAST.EuroVA13.007-011. URL: http://dx.doi.org/10.2312/PE.EuroVAST.EuroVA13.007-011 (cited on pages 6, 78).
Contribution Clarification: Conceptualization and Contributions: The main research focus of the paper is to develop a visual comparison approach for ranking/ordering data. The project was initiated by T. Schreck. I developed the glyph design for the ranking comparison. T. Schreck developed the idea to use the glyph design in a matrix layout. J. Davey and T. Schreck reviewed the ideas and contributed to the conceptualization. Implementation/Instantiation effort: I developed the code and all its components, esp. the glyph. J. Davey and T. Schreck reviewed and pointed to open questions/missing features. I conducted most experiments (except the Sequence Data Experiment). S. Simon contributed the Sequence Data Analysis data set and interpreted the results. Authorship: J. Davey, T. Schreck and I worked jointly on the structure, textual revision, the didactical concept of the paper. I wrote/edited most of the text myself. T. Schreck reviewed and edited the text and contributed to the Related work sections. S. Simon contributed the Sequence Data Analysis Case Study. Supervision: J. Kohlhammer, D. Keim and T. Schreck supervised the paper project and commented on the contributions.

[Ber+14a] Jürgen Bernard et al. "Towards a user-defined visual-interactive definition of similarity functions for mixed data". In: *Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on*. IEEE. 2014, pp. 227–228 (cited on pages 160, 215).

[Rüd+15a] Laura von Rüden et al. "Separating the Wheat from the Chaff: Identifying Relevant and Similar Performance Data with Visual Analytics". In: *Proceedings of the 2nd Workshop on Visual Performance Analysis*. VPA '15. Austin, Texas: ACM, 2015, 4:1–4:8. ISBN: 978-1-4503-4013-7. DOI: 10.1145/2835238.2835242. URL: http://doi.acm.org/10.1145/2835238.2835242 (cited on pages 12, 56).

[Sha+14] Lin Shao et al. "Guided Sketching for Visual Search and Exploration in Large Scatter Plot Spaces". In: *Proc. EuroVA International Workshop on Visual Analytics*. Ed. by M. Pohl and J. Roberts. The Eurographics Association, 2014. DOI: 10.2312/eurova.20141140. URL: http://dx.doi.org/10.2312/eurova.20141140 (cited on pages 112, 159).

Further References

[AH04] J. Abello and F. van Ham. "Matrix zoom: A visual interface to semi-external graphs". In: *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*. Vol. 1. 2. IEEE, 2004, pp. 183–190. ISBN: 0780387791. URL: http://tubiblio.ulb.tu-darmstadt.de/43658/%20http://ieeexplore.ieee.org/xpls/abs%5C_all.jsp?arnumber=1382907%20http://www.springerlink.com/index/E39T29TGEVJYLW7H.pdf (cited on pages 35, 75, 76, 81, 91).

[AS94] Christopher Ahlberg and Ben Shneiderman. “Visual information seeking using the FilmFinder”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1994, pp. 433–434 (cited on page 79).

[Aig+11] Wolfgang Aigner et al. *Visualization of Time-Oriented Data*. Human-Computer Interaction Series. Springer, 2011, pp. 1–267. ISBN: 978-0-85729-078-6 (cited on page 81).

[Alb+10] Georgia Albuquerque et al. “Improving the visual analysis of high-dimensional datasets using quality measures”. In: *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*. IEEE. 2010, pp. 19–26 (cited on pages 23, 113, 121).

[Alb+09a] Georgia Albuquerque et al. “Quality-Based Visualization Matrices”. In: *Proc. Vision, Modeling and Visualization (VMV) 2009*. Braunschweig, Germany, Nov. 2009, pp. 341–349 (cited on page 23).

[Alb+09b] Georgia Albuquerque et al. “Quality-Based Visualization Matrices”. In: *Proceedings of the Vision, Modeling and Visualization (VMV)*. 2009, pp. 341–350 (cited on page 151).

[Ale13] Eric Alexander. “Serendip : Turning Topics Back to the Text”. In: (2013) (cited on pages 37, 74, 75).

[Alt+97] S F Altschul et al. “Gapped BLAST and PSI-BLAST: a new generation of protein database search programs”. In: *Nucleic Acids Res* 25.17 (Sept. 1997), pp. 3389–3402 (cited on page 105).

[AWD12] Anushka Anand, Leland Wilkinson, and Tuan Nhon Dang. “Visual pattern discovery using random projections”. In: *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)* (Oct. 2012), pp. 43–52. DOI: 10.1109/VAST.2012.6400490. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6400490 (cited on pages 75, 76).

[And09] Mircea Andrecut. “Parallel GPU implementation of iterative PCA algorithms”. In: *Journal of Computational Biology* 16.11 (2009), pp. 1593–1599 (cited on page 49).

[AAB07] Gennady Andrienko, Natalia Andrienko, and Ulrich Bartling. “Visual Analytics Approach to User-Controlled Evacuation Scheduling”. In: *IEEE Symposium on Visual Analytics Science and Technology* (Oct. 2007), pp. 43–50. DOI: 10.1109/VAST.2007.4388995. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4388995 (cited on pages 35, 76).

[And+10] G Andrienko et al. “Space-in-Time and Time-in-Space Self-Organizing Maps for Exploring Spatiotemporal Patterns”. In: 29.3 (2010) (cited on page 76).

[ABK98] M. Ankerst, S. Berchtold, and D.A Keim. “Similarity clustering of dimensions for an enhanced visualization of multidimensional data”. In: *IEEE Symposium on Information Visualization, 1998. Proceedings*. Oct. 1998, pp. 52–60, 153 (cited on page 150).

[ACR03] David Applegate, William Cook, and André Rohe. “Chained Lin-Kernighan for large traveling salesman problems”. In: *INFORMS Journal on Computing* 15.1 (2003), pp. 82–92 (cited on page 58).

[ABH98] J. Atkins, E. Boman, and B. Hendrickson. “A Spectral Algorithm for Seriation and the Consecutive Ones Problem”. In: *SIAM Journal on Computing* 28.1 (1998), pp. 297–310. DOI: 10.1137/S0097539795285771. eprint: http://dx.doi.org/10.1137/S0097539795285771. URL: http://dx.doi.org/10.1137/S0097539795285771 (cited on pages 37, 40, 47).

[Bac13] Benjamin Bach. “Visualizing Dense Dynamic Networks with Matrix Cubes”. In: (2013), pp. 3–4 (cited on pages 35, 74).

[Bac+15] B. Bach et al. “Small MultiPiles: Piling Time to Explore Temporal Patterns in Dynamic Networks”. In: *Computer Graphics Forum* 34.3 (2015), pp. 31–40. ISSN: 01677055. DOI: 10.1111/cgf.12615. URL: http://doi.wiley.com/10.1111/cgf.12615 (cited on page 73).

[BL13] K. Bache and M. Lichman. *University of California (UCI) Machine Learning Repository*. Online. http://archive.ics.uci.edu/ml/. June 2013. URL: http://archive.ics.uci.edu/ml (cited on page 204).

[BR10] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 2010 (cited on page 161).

[BR11a] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval - the concepts and technology behind search, Second edition*. Pearson Education Ltd., Harlow, England, 2011. ISBN: 978-0-321-41691-9 (cited on page 123).

[BR11b] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval - the concepts and technology behind search, Second edition*. Pearson Education Ltd., Harlow, England, 2011. ISBN: 978-0-321-41691-9 (cited on page 158).

[BGJ01] Ziv Bar-Joseph, David K. Gifford, and Tommi S. Jaakkola. “Fast optimal leaf ordering for hierarchical clustering”. In: *Bioinformatics* 17.suppl 1 (2001), S22–S29 (cited on pages 37, 41, 43).

[BM98] Vladimir Batagelj and Andrej Mrvar. “Pajek-program for large network analysis”. In: *Connections* 21.2 (1998), pp. 47–57 (cited on page 64).

[BGS01] Patrick Baudisch, Nathaniel Good, and Paul Stewart. “Focus Plus Context Screens: Combining Display Technology with Visualization Techniques”. In: *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*. UIST ’01. Orlando, Florida: ACM, 2001, pp. 31–40. ISBN: 1-58113-438-X. DOI: 10.1145/502348.502354. URL: http://doi.acm.org/10.1145/502348.502354 (cited on page 181).

[Bay+08] Herbert Bay et al. “Speeded-Up Robust Features (SURF)”. In: *Comput. Vis. Image Underst.* 110.3 (June 2008), pp. 346–359. ISSN: 1077-3142. DOI: 10.1016/j.cviu.2007.09.014. URL: http://dx.doi.org/10.1016/j.cviu.2007.09.014 (cited on page 121).

[BD10] Fabian Beck and Stephan Diehl. “Visual comparison of software architectures”. In: *Proceedings of the 5th international symposium on Software visualization - SOFTVIS ’10* (2010), p. 183. DOI: 10.1145/1879211.1879238. URL: http://portal.acm.org/citation.cfm?doid=1879211.1879238 (cited on pages 34, 73).

[Beh+12a] Michael Behrisch et al. “Matrix-Based Visual Correlation Analysis on Large Timeseries Data”. In: *Proc. IEEE Symposium on Visual Analytics Science and Technology (Poster Paper)* (2012) (cited on pages 37, 69, 81, 96).

[Ben92] Jon Jouis Bentley. “Fast algorithms for geometric traveling salesman problems”. In: *ORSA Journal on computing* 4.4 (1992), pp. 387–411 (cited on pages 37, 57, 58, 64).

[Ber+11a] J[[[ERROR FOR PACKAGE inputenc]]]rgen Bernard et al. “A visual digital library approach for time-oriented scientific primary data”. In: *Springer International Journal of Digital Libraries, ECDL 2010 Special Issue* (2011) (cited on page 160).

[Ber+14b] Jürgen Bernard et al. “User-based visual-interactive similarity definition for mixed data objects-concept and first implementation”. In: *22nd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG 2014. Communication Papers Proceedings : June 2 - June 5, 2015, Plzen*. ISBN: 978-80-86943-71-8. European Association for Computer Graphics -EUROGRAPHICS, 2014, pp. 329–338 (cited on pages 160, 215).

[Ber+10] J. Bernard et al. “A Visual Digital Library Approach for Time-Oriented Research Data”. In: *Proceedings of the European Conference on Digital Libraries*. Vol. 6273. Lecture Notes in Computer Science. Springer, 2010, pp. 352–363. URL: ./docs/ecdl10.pdf (cited on page 160).

[Ber81] J. Bertin. *Graphics and graphic information-processing*. de Gruyter, 1981. ISBN: 9783110088687 (cited on pages 9, 10, 37, 71, 80).

[Ber73] Jacques Bertin. *Sémiologie Graphique - Les diagrammes, les reseaux, les cartes*. Paris: Éditions Gauthier-Villars, 1973 (cited on pages 9, 10, 33, 37, 59).

[BTK11a] E. Bertini, A. Tatu, and D. Keim. “Quality Metrics in High-Dimensional Data Visualization: An Overview and Systematization”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (Dec. 2011), pp. 2203–2212 (cited on page 22).

[BS06a] Enrico Bertini and Giuseppe Santucci. “Give Chance a Chance: Modeling Density to Enhance Scatter Plot Quality Through Random Data Sampling”. In: *Information Visualization* 5.2 (June 2006), pp. 95–110 (cited on pages 13, 21).

[BTK11b] Enrico Bertini, Andrada Tatu, and Daniel Keim. “Quality metrics in high-dimensional data visualization: An overview and systematization”. In: *Visualization and Computer Graphics, IEEE Transactions on* 17.12 (2011), pp. 2203–2212 (cited on pages 21, 23, 74).

[BTK11c] Enrico Bertini, Andrada Tatu, and Daniel A. Keim. “Quality Metrics in High-Dimensional Data Visualization: An Overview and Systematization”. In: *IEEE Symposium on Information Visualization (InfoVis)* 17.12 (Dec. 2011), (cited on page 113).

[Ber+11b] Enrico Bertini et al. “HiTSEE: A visualization tool for hit selection and analysis in high-throughput screening experiments”. In: *2011 IEEE Symposium on Biological Data Visualization (BioVis)*. (Oct. 2011), pp. 95–102. DOI: 10.1109/BioVis.2011.6094053. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6094053 (cited on page 75).

[Bez+10a] Anastasia Bezerianos et al. “GeneaQuilts : A System for Exploring Large Genealogies”. In: 16.6 (2010), pp. 1073–1081 (cited on pages 33, 35).

[Bez+10b] A. Bezerianos et al. “GeneaQuilts: A System for Exploring Large Genealogies”. In: *IEEE TVCG* 16.6 (2010), pp. 1073–1081. ISSN: 1077-2626 (cited on pages 72, 76).

[] *BOOST C++ Libraries*. `http://www.boost.org`. URL: http://www.boost.org (cited on page 64).

[BL12] Ingwer Borg and James Lingoes. *Multidimensional similarity structure analysis*. Springer Science & Business Media, 2012 (cited on page 49).

[BZM07] A. Bosch, A. Zisserman, and X. Munoz. “Representing Shape with a Spatial Pyramid Kernel”. In: *ACM International Conference on Image and Video Retrieval*. 2007 (cited on page 121).

[Bou+13] N Boukhelifa et al. “Evolutionary Visual Exploration: Evaluation With Expert Users”. In: *Computer Graphics Forum* 32.3pt1 (2013), pp. 31–40. DOI: 10.1111/cgf.12090. URL: http://doi.wiley.com/10.1111/cgf.12090 (cited on pages 37, 159).

[BW13] Remko B J Van Brakel and Michel A Westenberg. “COMBat : Visualizing Co-Occurrence of Annotation Terms”. In: (2013), pp. 17–24 (cited on pages 37, 71, 74, 75).

[Bra07] Ulrik Brandes. “Optimal leaf ordering of complete binary trees”. In: *Journal of Discrete Algorithms* 5.3 (2007), pp. 546–552. ISSN: 1570-8667. DOI: http://dx.doi.org/10.1016/j.jda.2006.09.003. URL: http://www.sciencedirect.com/science/article/pii/S1570866706000839 (cited on pages 37, 43).

[BN11] Ulrik Brandes and Bobo Nick. “Asymmetric Relations in Longitudinal Social Networks”. In: *IEEE Trans. Vis. Comput. Graph.* 17.12 (2011), pp. 2283–2290 (cited on page 82).

[BP07] Ulrik Brandes and Christian Pich. “Eigensolver methods for progressive multidimensional scaling of large data”. In: *Graph Drawing*. Springer. 2007, pp. 42–53 (cited on page 50).

[Bra97] R. Brath. “Metrics for Effective Information Visualization”. In: *Proceedings of the IEEE Symposium on Information Visualization*. Washington, DC, USA: IEEE Computer Society, 1997, pp. 108–111 (cited on page 21).

[Bre+16] Matthew Brehmer et al. “Matches , Mismatches , and Methods : Multiple-View Workflows for Energy Portfolio Analysis”. In: 22.1 (2016), pp. 449–458. ISSN: 1077-2626. DOI: 10.1109/TVCG.2015.2466971 (cited on pages 37, 72, 76).

[BBA75] Ronald L Breiger, Scott A Boorman, and Phipps Arabie. “An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling”. In: *Journal of mathematical psychology* 12.3 (1975), pp. 328–383 (cited on pages 37, 46).

[BH11] Sebastian Bremm and Kay Hamacher. “Interactive Visual Comparison of Multiple Trees”. In: (2011), pp. 29–38 (cited on pages 34, 74, 75).

[Bre+11] Sebastian Bremm et al. “Assisted Descriptor Selection Based on Visual Comparative Data Analysis”. In: *Computer Graphics Forum* 30.3 (June 2011), pp. 891–900. ISSN: 01677055. DOI: 10.1111/j.1467-8659.2011.01938.x. URL: http://doi.wiley.com/10.1111/j.1467-8659.2011.01938.x (cited on pages 36, 76).

[Bre+10] Sebastian Bremm et al. “Computing and visually analyzing mutual information in molecular co-evolution”. In: *BMC Bioinformatics* 11:330 (2010) (cited on page 81).

[Bre12] Color Brewer. *Heatmap Coloring Options - Color Brewer*. http://colorbrewer2.org/. Online; accessed 25-Feb-2012. 2012 (cited on page 102).

[Bro+12] E.T. Brown et al. “Dis-function: Learning distance functions interactively”. In: *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*. Oct. 2012, pp. 83–92. DOI: 10.1109/VAST.2012.6400486 (cited on pages 159, 160).

[BKS08] Michael J. Brusco, Hans-Friedrich Köhn, and Stephanie Stahl. “Heuristic Implementation of Dynamic Programming for Matrix Permutation Problems in Combinatorial Data Analysis”. English. In: *Psychometrika* 73.3 (2008), pp. 503–522. ISSN: 0033-3123. DOI: 10.1007/s11336-007-9049-5. URL: http://dx.doi.org/10.1007/s11336-007-9049-5 (cited on pages 37, 41, 52, 53).

[BS05] Michael J. Brusco and Stephanie Stahl. “Optimal Least-Squares Unidimensional Scaling: Improved Branch-and-Bound Procedures and Comparison to Dynamic Programming”. English. In: *Psychometrika* 70.2 (2005), pp. 253–270. ISSN: 0033-3123. DOI: 10.1007/s11336-002-1032-6. URL: http://dx.doi.org/10.1007/s11336-002-1032-6 (cited on pages 37, 41, 63).

[BS06b] M.J. Brusco and S. Stahl. *Branch-and-Bound Applications in Combinatorial Data Analysis*. Statistics and Computing. Springer, 2006. ISBN: 9780387288109. URL: http://books.google.de/books?id=RWo6uz6NcSwC (cited on page 37).

[Bur+11] Michael Burch et al. “Parallel Edge Splatting for Scalable Dynamic Graph Visualization”. In: *IEEE Trans. Vis. Comput. Graph.* 17.12 (2011), pp. 2344–2353 (cited on page 81).

[BCR12] Patrick Butler, Prithwish Chakraborty, and Naren Ramakrishan. “The Deshredder: A visual analytic approach to reconstructing shredded documents”. In: *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2012), pp. 113–122. DOI: 10.1109/VAST.2012.6400560. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6400560 (cited on page 72).

[Byš+15] Jan Byška et al. “MoleCollar and Tunnel Heat Map Visualizations for Conveying Spatio-Temporo-Chemical Properties Across and Along Protein Voids”. In: *Computer Graphics Forum*. Vol. 34. 3. Wiley Online Library. 2015, pp. 1–10. DOI: 10.1111/cgf.12612 (cited on page 72).

[CK04] T. Caelli and S. Kosinov. “An eigenspace projection clustering method for inexact graph matching”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 26.4 (2004), pp. 515–519. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2004.1265866 (cited on pages 113, 133, 167).

[CT15] Yuanchao Cai and Karen Tay. “Visualizing Movement in Theme Park”. In: (2015), pp. 147–148 (cited on pages 73, 75, 76).

[CBL12] Waldo Cancino, Nadia Boukhelifa, and Evelyne Lutton. “Evographdice: Interactive evolution for visual analytics”. In: *2012 IEEE Congress on Evolutionary Computation*. IEEE. 2012, pp. 1–8 (cited on page 159).

[CP05] Gilles Caraux and Sylvie Pinloche. “PermutMatrix: a graphical environment to arrange gene expression profiles in optimal linear order”. In: *Bioinformatics* 21.7 (2005), pp. 1280–1281 (cited on pages 42, 71, 229).

[CMS99a] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman, eds. *Readings in Information Visualization: Using Vision to Think*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999. ISBN: 1-55860-533-9 (cited on page 22).

[CMS99b] Stuart K Card, Jock D Mackinlay, and Ben Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999 (cited on pages 21, 182).

[Car97] Miguel[[[ERROR FOR PACKAGE inputenc]]]A. Carreira-Perpinan. *A Review of Dimension Reduction Techniques*. 1997. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.44.6279 (cited on page 135).

[CP01] Gert Cauwenberghs and Tomaso Poggio. “Incremental and decremental support vector machine learning”. In: *Advances in neural information processing systems* (2001), pp. 409–415 (cited on page 191).

[Cen14] Pew Research Center. *Six new facts about Facebook*. Online. Article published on February 3, 2014; Numbers from 2014. Feb. 2014. URL: http://www.pewresearch.org/fact-tank/2014/02/03/6-new-facts-about-facebook/ (cited on page 2).

[Cha+83] J. M. Chambers et al. *Graphical Methods for Data Analysis*. New York: Chapman and Hall, 1983 (cited on pages 82, 105).

[CG80] Wing-Man Chan and Alan George. “A linear time implementation of the reverse Cuthill-McKee algorithm”. In: *BIT Numerical Mathematics* 20.1 (1980), pp. 8–14 (cited on pages 37, 55).

[Cha+07a] Remco Chang et al. “Legible cities: focus-dependent multi-resolution visualization of urban relationships.” In: *IEEE transactions on visualization and computer graphics* 13.6 (2007), pp. 1169–75. ISSN: 1077-2626. DOI: 10.1109/TVCG.2007.70574. URL: http://www.ncbi.nlm.nih.gov/pubmed/17968061 (cited on pages 34, 75).

[Cha+07b] Remco Chang et al. “WireVis: Visualization of Categorical, Time-Varying Data From Financial Transactions”. In: *2007 IEEE Symposium on Visual Analytics Science and Technology* (Oct. 2007), pp. 155–162. DOI: 10.1109/VAST.2007.4389009. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4389009 (cited on pages 35, 75, 76).

[CB08a] S. A. Chatzichristofis and Y. S. Boutalis. “FCTH: Fuzzy Color and Texture Histogram - A Low Level Feature for Accurate Image Retrieval”. In: *Image Analysis for Multimedia Interactive Services, 2008. WIAMIS ’08. Ninth International Workshop on*. May 2008, pp. 191–196. DOI: 10.1109/WIAMIS.2008.24 (cited on page 121).

[CB08b] Savvas A Chatzichristofis and Yiannis S Boutalis. “CEDD: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval”. In: *Computer vision systems*. Springer, 2008, pp. 312–322 (cited on pages 121, 131).

[Che02] C. H. Chen. “Generalized association plots: information visualization via iteratively generated correlation matrices”. In: *Statistica Sinica* 12 (2002), pp. 7–29. URL: http://www.stat.sinica.edu.tw/statistica/j12n1/j12n11/j12n11.htm (cited on pages 37, 46, 229).

[CMP09] Jin Chen, Alan M MacEachren, and Donna J Peuquet. “Constructing overview+detail dendrogram-matrix views.” In: *IEEE transactions on visualization and computer graphics* 15.6 (2009), pp. 889–96. ISSN: 1077-2626. DOI: 10.1109/TVCG.2009.130. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3165051%5C&tool=pmcentrez%5C&rendertype=abstract (cited on pages 34, 74).

[CC00a] Yizong Cheng and George M Church. “Biclustering of expression data.” In: *Ismb*. Vol. 8. 2000, pp. 93–103 (cited on pages 37, 60–62, 229).

[CCB12] F. Chevalier, C. Collins, and R. Balakrishnan. “Facilitating Discourse Analysis with Interactive Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (Dec. 2012), pp. 2639–2648. ISSN: 1077-2626. DOI: 10.1109/TVCG.2012.226. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6327270 (cited on pages 72, 75).

[Chi00] Ed H. Chi. “A taxonomy of visualization techniques using the data state reference model”. In: *Proceedings of the IEEE Symposium on Information Visualization*. 2000, pp. 69–75 (cited on page 22).

[CV07] Rudi L. Cilibrasi and Paul M. B. Vitanyi. “The Google Similarity Distance”. In: *IEEE Trans. on Knowl. and Data Eng.* 19 (3 Mar. 2007), pp. 370–383. ISSN: 1041-4347. DOI: 10.1109/TKDE.2007.48 (cited on page 102).

[CGW13] Kris Cook, Georges Grinstein, and Mark Whiting. *VAST Challenge 2013*. 2013 (cited on page 202).

[Cor+96] L.P. Cordella et al. “An efficient algorithm for the inexact matching of ARG graphs using a contextual transformational model”. In: *Pattern Recognition, 1996., Proc. of the 13th Int. Conference on*. Vol. 3. IEEE. 1996, 180–184 vol.3 (cited on page 113).

[CC00b] T F Cox and M A A Cox. “Multidimensional Scaling”. In: *DOI: 10.1214/08-AOAS165SUPP. 31 de Leeuw*. Chapman, 2000 (cited on pages 135, 167).

[Cro58] Georges A Croes. “A method for solving traveling-salesman problems”. In: *Operations research* 6.6 (1958), pp. 791–812 (cited on page 58).

[CDS09] Patricia J Crossno, Daniel M Dunlavy, and Timothy M Shead. “LSAView : A Tool for Visual Exploration of Latent Semantic Modeling”. In: (2009), pp. 83–90 (cited on pages 75, 76).

[CKX11] Ross E Curtis, Peter Kinnaird, and Eric P Xing. “GenAMap: Visualization strategies for structured association mapping”. In: *2011 IEEE Symposium on Biological Data Visualization (BioVis)*. (2011), pp. 87–94. DOI: 10.1109/BioVis.2011.6094052. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6094052 (cited on page 75).

[CM69] E. Cuthill and J. McKee. “Reducing the Bandwidth of Sparse Symmetric Matrices”. In: *Proceedings of the 1969 24th National Conference*. ACM ’69. New York, NY, USA: ACM, 1969, pp. 157–172. DOI: 10.1145/800195.805928. URL: http://doi.acm.org/10.1145/800195.805928 (cited on pages 37, 54–56, 64, 229).

[DW14a] Tuan Nhon Dang and L. Wilkinson. “ScagExplorer: Exploring Scatterplots by Their Scagnostics”. In: *Pacific Visualization Symposium (PacificVis), 2014 IEEE*. Mar. 2014, pp. 73–80. DOI: 10.1109/PacificVis.2014.42 (cited on page 158).

[DW14b] Tuan Nhon Dang and L. Wilkinson. “Transforming Scagnostics to Reveal Hidden Features”. In: *Visualization and Computer Graphics, IEEE Transactions on* 20.12 (Dec. 2014), pp. 1624–1632. ISSN: 1077-2626 (cited on page 150).

[DK11] A. Dasgupta and R. Kosara. “Adaptive Privacy-Preserving Visualization Using Parallel Coordinates”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2241–2248. DOI: 10.1109/TVCG.2011.163. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6064989 (cited on page 22).

[DK10a] A. Dasgupta and R. Kosara. “Pargnostics: Screen-Space Metrics for Parallel Coordinates”. In: *Visualization and Computer Graphics, IEEE Transactions on* 16.6 (Nov. 2010), pp. 1017–1026. ISSN: 1077-2626. DOI: 10.1109/TVCG.2010.184 (cited on pages 112, 113, 117, 158).

[DKG15] A. Dasgupta, R. Kosara, and L. Gosink. “VIMTEX : A V isualization I nterface for M ultivariate, T ime-Varying, Geological Data E xploration”. In: *Computer Graphics Forum* 34.3 (2015), pp. 341–350. ISSN: 01677055. DOI: 10.1111/cgf.12646. URL: http://doi.wiley.com/10.1111/cgf.12646 (cited on pages 73, 75).

[DK10b] Aritra Dasgupta and Robert Kosara. “Pargnostics: Screen-Space Metrics for Parallel Coordinates”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 1017–1026. ISSN: 1077-2626. DOI: 10.1109/TVCG.2010.184. URL: http://dx.doi.org/10.1109/TVCG.2010.184 (cited on page 79).

[DKN08] Thomas Deselaers, Daniel Keysers, and Hermann Ney. “Features for image retrieval: an experimental comparison”. In: *Information Retrieval* 11.2 (Apr. 2008), pp. 77–107. ISSN: 1386-4564. URL: http://dx.doi.org/10.1007/s10791-007-9039-3 (cited on pages 114, 159).

[DM71] Stephen B Deutsch and John J Martin. “An ordering algorithm for analysis of data arrays”. In: *Operations Research* 19.6 (1971), pp. 1350–1362 (cited on pages 37, 51, 52, 229).

[DRW15] K. Dinkla, N. Henry Riche, and M.A. Westenberg. “Dual Adjacency Matrix: Exploring Link Groups in Dense Networks”. In: *Computer Graphics Forum* 34.3 (2015), pp. 311–320. ISSN: 01677055. DOI: 10.1111/cgf.12643. URL: http://doi.wiley.com/10.1111/cgf.12643 (cited on pages 73, 75).

[DWW12] K. Dinkla, M.A. Westenberg, and J.J. van Wijk. “Compressed Adjacency Matrices: Untangling Gene Regulatory Networks”. In: *Visualization and Computer Graphics, IEEE Transactions on* 18.12 (2012), pp. 2457–2466. ISSN: 1077-2626. DOI: 10.1109/TVCG.2012.208 (cited on pages 34, 35, 37, 74, 76, 81).

[DPS02] Josep Dı[[[ERROR FOR PACKAGE inputenc]]]az, Jordi Petit, and Maria Serna. “A Survey of Graph Layout Problems”. In: *ACM Comput. Surv.* 34.3 (Sept. 2002), pp. 313–356. ISSN: 0360-0300. DOI: 10.1145/568522.568523. URL: http://doi.acm.org/10.1145/568522.568523 (cited on page 54).

[DH72] Richard O. Duda and Peter E. Hart. “Use of the Hough Transformation to Detect Lines and Curves in Pictures”. In: *Commun. ACM* 15.1 (Jan. 1972), pp. 11–15. ISSN: 0001-0782. DOI: 10.1145/361237.361242. URL: http://doi.acm.org/10.1145/361237.361242 (cited on page 186).

[EW94] Peter Eades and Nicholas C. Wormald. “Edge crossings in drawings of bipartite graphs”. In: *Algorithmica* 11.4 (1994), pp. 379–403. ISSN: 1432-0541. DOI: 10.1007/BF01187020. URL: http://dx.doi.org/10.1007/BF01187020 (cited on page 53).

[ESS92] Stephen G. Eick, Joseph L. Steffen, and Eric E. Sumner Jr. “Seesoft-A Tool for Visualizing Line Oriented Software Statistics”. In: *IEEE Trans. Softw. Eng.* 18 (11 Nov. 1992), pp. 957–968. ISSN: 0098-5589. DOI: 10.1109/32.177365. URL: http://dl.acm.org/citation.cfm?id=141344.141348 (cited on pages 82, 88).

[EMR06] Stephen Eick, Justin Mauger, and Alan Ratner. “Visualizing the Performance of Computational Linguistics Algorithms”. In: *2006 IEEE Symposium On Visual Analytics And Technology* (Oct. 2006), pp. 151–157. DOI: 10.1109/VAST.2006.261417. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4035760 (cited on pages 75, 76).

[Eis+98] Michael B. Eisen et al. “Cluster analysis and display of genome wide expression patterns”. In: *Proceedings of the National Academy of Sciences* 95.25 (1998), pp. 14863–14868. eprint: http://www.pnas.org/content/95/25/14863.full.pdf+html. URL: http://www.pnas.org/content/95/25/14863.abstract (cited on pages 37, 42, 64, 69).

[Eit+10] Mathias Eitz et al. “An evaluation of descriptors for large-scale image retrieval from sketched feature lines”. In: *Computers & Graphics* 34.5 (2010), pp. 482–498 (cited on page 159).

[Eit+12] Mathias Eitz et al. “Sketch-Based Shape Retrieval”. In: *ACM Transactions on Graphics* 31.4 (2012), 31:1–31:10 (cited on page 157).

[ED07a] G. Ellis and A. Dix. “A Taxonomy of Clutter Reduction for Information Visualisation”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (Nov. 2007), pp. 1216–1223 (cited on page 22).

[ED07b] G. Ellis and A. Dix. “A Taxonomy of Clutter Reduction for Information Visualisation”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1216–1223. ISSN: 1077-2626. DOI: http://doi.ieeecomputersociety.org/10.1109/TVCG.2007.70535 (cited on page 112).

[EDF08a] N. Elmqvist, P. Dragicevic, and J. Fekete. “Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation”. In: *IEEE Transactions on Visualization and Computer Graphics* 14.6 (Nov. 2008), pp. 1539–1148 (cited on page 150).

[EDF08b] Niklas Elmqvist, Pierre Dragicevic, and Jean-Daniel Fekete. “Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation”. In: *IEEE Transactions on Visualization and Computer Graphics* 14.6 (2008), pp. 1141–1148 (cited on page 79).

[EDF08c] Niklas Elmqvist, Pierre Dragicevic, and Jean-Daniel Fekete. “Rolling the dice: multidimensional visual exploration using scatterplot matrix navigation.” In: *IEEE transactions on visualization and computer graphics* 14.6 (2008), pp. 1141–8. ISSN: 1077-2626. DOI: 10.1109/TVCG.2008.153. URL: http://www.ncbi.nlm.nih.gov/pubmed/18989008 (cited on page 74).

[Elm+08] Niklas Elmqvist et al. “ZAME: Interactive Large-Scale Graph Visualization”. In: *2008 IEEE Pacific Visualization Symposium* (Mar. 2008), pp. 215–222. DOI: 10.1109/PACIFICVIS.2008.4475479. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4475479 (cited on pages 34, 35, 37, 49, 71, 75, 81, 91).

[EMM12] EMM. *Europe Media Monitor*. http://emm.newsbrief.eu/. Online; accessed 25-Feb-2012. 2012 (cited on page 82).

[EFN12] Alex Endert, Patrick Fiaux, and Chris North. “Semantic interaction for visual text analytics”. In: *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*. CHI ’12. ACM. New York, NY, USA: ACM, 2012, 473–482. ISBN: 978-1-4503-1015-4. DOI: 10.1145/2207676.2207741. URL: http://doi.acm.org/10.1145/2207676.2207741 (cited on page 159).

[Ene] Energie Baden-Württemberg AG. *Intelligent Grid Project (in German)*. http://www.enbw.com/content/de/der_konzern/enbw_gesellschaften/regionalgesellschaft/aktuell/smartgrid/projekt_freiamt/index.jsp. Online; accessed 26th June 2012 (cited on page 97).

[FPS96a] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. “From data mining to knowledge discovery in databases”. In: *AI magazine* 17.3 (1996), p. 37 (cited on pages 8, 9, 80, 156).

[FPS96b] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. “The KDD Process for Extracting Useful Knowledge from Volumes of Data”. In: *Communications of the ACM* 39.11 (Nov. 1996), pp. 27–34 (cited on page 22).

[Fek04] Jean-Daniel Fekete. “The InfoVis Toolkit”. In: *Proc. of the IEEE Symp. on Info Vis*. INFOVIS ’04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 167–174. ISBN: 0-7803-8779-3 (cited on page 178).

[Fie73] Miroslav Fiedler. “Algebraic connectivity of graphs”. In: *Czechoslovak mathematical journal* 23.2 (1973), pp. 298–305 (cited on page 47).

[FE73] Martin A. Fischler and R.A. Elschlager. “The Representation and Matching of Pictorial Structures”. In: *IEEE TC* 22.1 (1973), pp. 67–92. ISSN: 0018-9340 (cited on page 113).

[Foo99] Jonathan Foote. “Visualizing Music and Audio Using Self-similarity”. In: *Proc. Multimedia*. Orlando, Florida, USA: ACM, 1999, pp. 77–80. ISBN: 1-58113-151-8. DOI: 10.1145/319463.319472. URL: http://doi.acm.org/10.1145/319463.319472 (cited on page 33).

[Fri10] Eszter Friedman. “Epidemic Outbreak Visualizer”. In: (2010) (cited on page 75).

[FT74] J.H. Friedman and J.W. Tukey. “A Projection Pursuit Algorithm for Exploratory Data Analysis”. In: *Computers, IEEE Transactions on* C-23.9 (Sept. 1974), pp. 881–890. ISSN: 0018-9340. DOI: 10.1109/T-C.1974.224051 (cited on page 158).

[Fri02] Michael Friendly. “Corrgrams: Exploratory displays for correlation matrices”. In: *The American Statistician* 56.4 (2002), pp. 316–324 (cited on pages 37, 45, 46, 48).

[FK03] Michael Friendly and Ernest Kwan. “Effect ordering for data displays”. In: *Computational statistics & data analysis* 43.4 (2003), pp. 509–539 (cited on page 37).

[GS62] D. Gale and L. S. Shapley. “College Admissions and the Stability of Marriage”. English. In: *The American Mathematical Monthly* 69.1 (1962), ISSN: 00029890. URL: http://www.jstor.org/stable/2312726 (cited on pages 135, 136).

[Gan+93] Emden R. Gansner et al. “A Technique for Drawing Directed Graphs”. In: *IEEE Trans. Softw. Eng.* 19.3 (Mar. 1993), pp. 214–230. ISSN: 0098-5589. DOI: 10.1109/32.221135. URL: http://dx.doi.org/10.1109/32.221135 (cited on page 53).

[Gel71] Alan E Gelfand. “Rapid seriation methods with archaeological applications”. In: *Hodson, FR, Kendall, DG and Tautu (eds), Mathematics in the Archaeological and Historical Sciences. Edinburgh University Press, Edinburgh* (1971), pp. 186–201 (cited on pages 37, 69).

[Geo71] J Alan George. “Computer implementation of the finite element method”. PhD thesis. DTIC Document, 1971 (cited on pages 37, 54, 55).

[GFC04] Mohammad Ghoniem, J Fekete, and Philippe Castagliola. “A comparison of the readability of graphs using node-link and matrix-based representations”. In: *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*. IEEE. 2004, pp. 17–24 (cited on pages 10, 33, 213).

[GFC05] Mohammad Ghoniem, Jean-Daniel Fekete, and Philippe Castagliola. “On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis”. In: *Information Visualization* 4.2 (July 2005), pp. 114–135. ISSN: 1473-8716. DOI: 10.1057/palgrave.ivs.9500092 (cited on pages 2, 80).

[GPS76] Norman E Gibbs, William G Poole Jr, and Paul K Stockmeyer. “An algorithm for reducing the bandwidth and profile of a sparse matrix”. In: *SIAM Journal on Numerical Analysis* 13.2 (1976), pp. 236–250 (cited on pages 37, 56).

[GLT15] Stefan Gladisch, Martin Luboschik, and Christian Tominski. “Toward Using Matrix Visualizations for Graph Editing”. In: (2015) (cited on pages 72, 76).

[GHS10] Michael Gleicher, David Hatfield, and David Shaffer. “Comparing Epistemic Frames: An Exercise in Visual Comparison”. In: *Eurovis 2010 Poster Proceedings*. June 2010 (cited on page 82).

[Gle+11a] Michael Gleicher et al. “Visual Comparison for Information Visualization”. In: *Information Visualization* 10.4 (Oct. 2011), pp. 289–309 (cited on page 81).

[Gle+11b] Michael Gleicher et al. “Visual comparison for information visualization”. In: *Information Visualization* 10.4 (2011), pp. 289–309 (cited on page 79).

[Goo+05] J.R. Goodall et al. “Preserving the big picture: visual network traffic analysis with TNV”. In: *IEEE Workshop on Visualization for Computer Security, 2005. (VizSEC 05)*. (2005), pp. 47–54. DOI: 10.1109/VIZSEC.2005.1532065. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1532065 (cited on pages 35, 37, 76).

[Gra15] Franz Graf. *JFeatureLib v1.6.3*. Sept. 2015. DOI: 10.5281/zenodo.31793. URL: http://dx.doi.org/10.5281/zenodo.31793 (cited on page 121).

[Gre+15] Robert Gregor et al. “Empirical evaluation of dissimilarity measures for 3D object retrieval with application to multi-feature retrieval”. In: *Content-Based Multimedia Indexing (CBMI), 2015 13th International Workshop on*. IEEE. 2015, pp. 1–6 (cited on page 69).

[GBD09] Martin Greilich, Michael Burch, and Stephan Diehl. “Visualizing the Evolution of Compound Digraphs with TimeArcTrees”. In: *Comput. Graph. Forum* 28.3 (2009), pp. 975–982 (cited on page 82).

[GW72] Gunnar Gruvaeus and Howard Wainer. “TWO ADDITIONS TO HIERARCHICAL CLUSTER ANALYSIS”. In: *British Journal of Mathematical and Statistical Psychology* 25.2 (1972), pp. 200–206. ISSN: 2044-8317. DOI: 10.1111/j.2044-8317.1972.tb00491.x. URL: http://dx.doi.org/10.1111/j.2044-8317.1972.tb00491.x (cited on pages 37, 41, 42).

[GWR09] Zhenyu Guo, Matthew O Ward, and Elke A Rundensteiner. “Model Space Visualization for Multivariate Linear Trend Discovery”. In: (2009), pp. 75–82 (cited on pages 37, 72, 75).

[GI89] Dan Gusfield and Robert W. Irving. *The stable marriage problem: structure and algorithms*. Cambridge, MA, USA: MIT Press, 1989. ISBN: 0-262-07118-5 (cited on page 136).

[HBH14] Michael Hahsler, Christian Buchta, and Kurt Hornik. *Infrastructure for seriation*. R package version 1.0-14. 2014. URL: http://CRAN.R-project.org/ (cited on page 64).

[HHB08] Michael Hahsler, Kurt Hornik, and Christian Buchta. “Getting things in order: An introduction to the R package seriation”. In: *Journal of Statistical Software* 25.3 (Mar. 2008), pp. 1–34. ISSN: 1548-7660 (cited on pages 32, 48, 65, 152).

[Hal+09] Mark Hall et al. “; The WEKA Data Mining Software: An Update; SIGKDD Explorations”. In: 11 (2009) (cited on page 144).

[Ham03] F. van Ham. "Using multilevel call matrices in large software projects". In: *IEEE Symposium on Information Visualization 2003 (IEEE Cat. No.03TH8714)* (2003), pp. 227–232. DOI: 10.1109/INFVIS.2003.1249030. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1249030 (cited on page 34).

[HKP11a] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. 3rd. Elsevier Ltd, Oxford, 2011 (cited on page 114).

[HKP11b] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. 3rd. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN: 0123814790, 9780123814791 (cited on page 186).

[HM02] Ju Han and Kai-Kuang Ma. "Fuzzy color histogram and its use in color image retrieval". In: *Image Processing, IEEE Transactions on* 11.8 (2002), pp. 944–952 (cited on page 121).

[Hao+07a] Ming C. Hao et al. "Intelligent Visual Analytics Queries". In: *2007 IEEE Symposium on Visual Analytics Science and Technology* (Oct. 2007), pp. 91–98. DOI: 10.1109/VAST.2007.4389001. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4389001 (cited on page 74).

[Hao+07b] Ming C. Hao et al. "Intelligent Visual Analytics Queries". In: *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*. 2007, pp. 91–98 (cited on page 157).

[HSD73] Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. "Textural features for image classification". In: *Systems, Man and Cybernetics, IEEE Transactions on* 6 (1973), pp. 610–621 (cited on pages 121, 132, 148).

[HK02] David Harel and Yehuda Koren. "Graph Drawing by High-Dimensional Embedding". In: *Revised Papers from the 10th International Symposium on Graph Drawing*. GD '02. London, UK, UK: Springer-Verlag, 2002, pp. 207–219. ISBN: 3-540-00158-1. URL: http://dl.acm.org/citation.cfm?id=647554.757122 (cited on pages 37, 44, 49, 229).

[Har64] Lawrence H Harper. "Optimal assignments of numbers to vertices". In: *Journal of the Society for Industrial and Applied Mathematics* (1964), pp. 131–135 (cited on page 37).

[Har72] John A Hartigan. "Direct clustering of a data matrix". In: *Journal of the american statistical association* 67.337 (1972), pp. 123–129 (cited on page 37).

[HD12] Christopher G. Healey and Brent M. Dennis. "Interest Driven Navigation in Visualization". In: *IEEE Trans. Vis. Comput. Graph.* 18.10 (2012), pp. 1744–1756 (cited on page 159).

[Hea95] Marti A. Hearst. "TileBars: visualization of term distribution information in full text information access". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. CHI '95. Denver, Colorado, United States: ACM Press/Addison-Wesley Publishing Co., 1995, pp. 59–66. ISBN: 0-201-84705-1 (cited on page 82).

[HP11] J. Heer and A. Perer. “Orion: A system for modeling, transformation and visualization of multidimensional heterogeneous networks”. In: *Information Visualization* (Dec. 2011), pp. 49–58. ISSN: 1473-8716. DOI: 10.1177/1473871612462152. URL: http://ivi.sagepub.com/lookup/doi/10.1177/1473871612462152 (cited on pages 36, 76).

[HP06] Marko Heikkilä and Matti Pietikäinen. “A texture-based method for modeling the background and detecting moving objects.” In: *IEEE transactions on pattern analysis and machine intelligence* 28.4 (2006), pp. 657–62 (cited on pages 121, 130).

[Hei+12] Florian Heimerl et al. “Visual Classifier Training for Text Document Retrieval”. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 18.12 (2012), pp. 2839–2848 (cited on page 191).

[HSW12] Julian Heinrich, John Stasko, and Daniel Weiskopf. “The parallel coordinates matrix”. In: *EuroVis–Short Papers* (2012), pp. 37–41 (cited on pages 35, 74).

[HF07] Nathalie Henry and Jean-Daniel Fekete. “MatLink: Enhanced Matrix Visualization for Analyzing Social Networks”. In: *Proceedings of the 11th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part II*. INTERACT’07. Rio de Janeiro, Brazil: Springer-Verlag, 2007, pp. 288–302. ISBN: 3-540-74799-0, 978-3-540-74799-4. URL: http://dl.acm.org/citation.cfm?id=1778331.1778362 (cited on page 76).

[HF06] Nathalie Henry and Jean-Daniel Fekete. “MatrixExplorer: a Dual-Representation System to Explore Social Networks”. In: *IEEE Transactions on Visualization and Computer Graphics* 12 (2006), pp. 677–684 (cited on pages 33–35, 37, 58, 71, 76, 80, 81).

[HFM07] Nathalie Henry, Jean-Daniel Fekete, and Michael J McGuffin. “NodeTrix: a hybrid visualization of social networks.” In: *IEEE transactions on visualization and computer graphics* 13.6 (2007), pp. 1302–9. ISSN: 1077-2626. DOI: 10.1109/TVCG.2007.70582. URL: http://www.ncbi.nlm.nih.gov/pubmed/17968078 (cited on pages 34, 35, 74, 76, 80).

[Hes+14] Martin Hess et al. “Visual Exploration of Parameter Influence on Phylogenetic Trees”. In: *IEEE Computer Graphics and Applications* 99.PrePrints (2014), p. 1. ISSN: 0272-1716. DOI: http://doi.ieeecomputersociety.org/10.1109/MCG.2014.2 (cited on page 194).

[Hil74] M. O. Hill. “Correspondence Analysis: A Neglected Multivariate Method”. English. In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 23.3 (1974), ISSN: 00359254. URL: http://www.jstor.org/stable/2347127 (cited on page 37).

[Hil79] Mark O Hill. “DECORANA-A FORTRAN program for detrended correspondence analysis and reciprocal averaging.” In: (1979) (cited on page 37).

[HS04] Harry Hochheiser and Ben Shneiderman. “Dynamic query tools for time series data sets: Timebox widgets for interactive exploration”. In: *Information Visualization* 3.1 (2004), pp. 1–18 (cited on pages 157, 160).

[HW08] Danny Holten and Jarke J. van Wijk. “Visual Comparison of Hierarchically Organized Data”. In: *Comput. Graph. Forum* 27.3 (2008), pp. 759–766 (cited on page 82).

[Hop14] Jenny Hope. *Chocolate and red wine can help stave off diabetes: High levels of antioxidants can regulate blood glucose levels*. Online. accessed 29 March 2014. Jan. 2014. URL: http://www.dailymail.co.uk/health/article-2542415/Chocolate-red-wine-help-stave-diabetes-High-levels-antioxidants-regulate-blood-glucose-levels.html (cited on page 206).

[Hou62] P.V.C. Hough. *METHOD AND MEANS FOR RECOGNIZING COMPLEX PATTERNS*. Dec. 1962 (cited on page 121).

[Hua+97] Jing Huang et al. "Image indexing using color correlograms". In: *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. June 1997, pp. 762–768. DOI: 10.1109/CVPR.1997.609412 (cited on page 121).

[Hub74] Lawrence Hubert. "Some Applications of Graph Theory and Related Non-Metric Techniques to Problems of Approximate Seriation The Case of Symmetric Proximity Measures". In: *British Journal of Mathematical and Statistical Psychology* 27.2 (1974), pp. 133–153. ISSN: 2044-8317. DOI: 10.1111/j.2044-8317.1974.tb00534.x. URL: http://dx.doi.org/10.1111/j.2044-8317.1974.tb00534.x (cited on pages 37, 41, 42, 64, 229).

[HG81] Lawrence J Hubert and Reginald G Golledge. "Matrix reorganization and dynamic programming: Applications to paired comparisons and unidimensional seriation". In: *Psychometrika* 46.4 (1981), pp. 429–441 (cited on pages 37, 53).

[HM76] J. W. Hunt and M. D. Mcilroy. *An Algorithm for Differential File Comparison*. Tech. rep. 41. Bell Laboratories Computing Science, July 1976. URL: http://www1.cs.dartmouth.edu/%5C~%7B%7Ddoug/diff.ps (cited on page 102).

[IML13] Jean-François Im, Michael J McGuffin, and Rock Leung. "GPLOM: the generalized plot matrix for visualizing multidimensional multivariate data". In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2606–2614 (cited on pages 33, 36, 213).

[Ing+10a] Stephen Ingram et al. "DimStiller : Workflows for Dimensional Analysis and Reduction". In: (2010), pp. 3–10 (cited on page 76).

[Ing+10b] Stephen Ingram et al. "DimStiller: Workflows for dimensional analysis and reduction". In: *Proceedings of the IEEE Conference on Visual Analytics Science and Technology, IEEE VAST 2010, Salt Lake City, Utah, USA, 24-29 October 2010, part of VisWeek 2010*. 2010, pp. 3–10. DOI: 10.1109/VAST.2010.5652392. URL: http://dx.doi.org/10.1109/VAST.2010.5652392 (cited on page 112).

[Ito15] Masahiko Itoh. "A System for Visual Exploration of Caution Spots from Vehicle Recorder Data". In: 0000011175 (2015) (cited on page 72).

[JZ13] Zbigniew Jerzak and Holger Ziekow. *DEBS Challenge 2013*. http://www.orgs.ttu.edu/debs2013/index.php. 2013 (cited on page 203).

[Jin+08] Ruoming Jin et al. "Overlapping matrix pattern visualization: A hypergraph approach". In: *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE. 2008, pp. 313–322 (cited on pages 37, 62).

[JJ09] S. Johansson and J. Johansson. "Interactive Dimensionality Reduction Through User-defined Combinations of Quality Metrics". In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (Nov. 2009), pp. 993–1000 (cited on page 21).

[Jol86] I T Jolliffe. *Principal Component Analysis*. 1986 (cited on pages 135, 167).

[Kai+15] Sanjay Kairam et al. "Refinery: Visual Exploration of Large, Heterogeneous Networks through Associative Browsing". In: *Comput. Graph. Forum* 34.3 (2015), pp. 301–310. DOI: 10.1111/cgf.12642. URL: http://dx.doi.org/10.1111/cgf.12642 (cited on pages 89, 157).

[Kai11] Sebastian Kaiser. "Biclustering: Methods, Software and Application". PhD thesis. lmu, 2011 (cited on pages 37, 62).

[KL08] Sebastian Kaiser and Friedrich Leisch. *A Toolbox for Bicluster Analysis in R*. 2008. URL: http://nbn-resolving.de/urn/resolver.pl?urn=nbn:de:bvb:19-epub-3293-7 (cited on pages 64, 65).

[KY01] E. Kasutani and A. Yamada. "The MPEG-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval". In: *Image Processing, 2001. Proceedings. 2001 International Conference on*. Vol. 1. 2001, 674–677 vol.1. DOI: 10.1109/ICIP.2001.959135 (cited on page 121).

[Kei00] D.A. Keim. "Designing pixel-oriented visualization techniques: theory and applications". In: *IEEE Transactions on Visualization and Computer Graphics* 6.1 (Jan. 2000), pp. 59–78 (cited on page 21).

[KAK95] Daniel A. Keim, Mihael Ankerst, and Hans-Peter Kriegel. "Recursive Pattern: A Technique for Visualizing Very Large Amounts of Data". In: *IEEE Visualization*. 1995, pp. 279–286 (cited on page 81).

[KO07] Daniel A. Keim and Daniela Oelke. "Literature fingerprinting: A new method for visual literary analysis". In: *IEEE Symposium on Visual Analytics Science and Technology (VAST)*. peer-reviewed (full). n/a, 2007, pp. 115–122 (cited on page 82).

[Kei+10a] Daniel A. Keim et al. "Generalized Scatter Plots". In: *Information Visualization* 9.4 (Dec. 2010), pp. 301–311 (cited on page 21).

[Kei+07] Daniel A. Keim et al. "Intelligent Visual Analytics Queries". In: *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST 2007)*. 2007 (cited on page 159).

[Kei+10b] Daniel A. Keim et al., eds. *Mastering the Information Age – Solving Problems with Visual Analytics*. Eurographics Association, 2010 (cited on page 21).

[Ken63] David G Kendall. "A statistical approach to Flinders petries sequence-dating". In: *Bulletin of the International Statistical Institute* 40.2 (1963), pp. 657–681 (cited on page 37).

[Ker03] Bernard Kerr. "Thread arcs: an email thread visualization". In: *Proc. IEEE Symposium on Information Visualization*. Seattle, Washington: IEEE Computer Society, 2003, pp. 211–218. ISBN: 0-7803-8154-8 (cited on page 82).

[KS12] W Kienreich and C Seifert. "Visual Exploration of Feature-Class Matrices for Classification Problems". In: (2012), pp. 37–41. DOI: 10.2312/PE/EuroVAST/EuroVA12/037-041 (cited on page 74).

[KY05] Dohyun Kim and Bong-Jin Yum. "Collaborative filtering based on iterative principal component analysis". In: *Expert Systems with Applications* 28.4 (2005), pp. 823–830 (cited on page 49).

[Kin70] Ian P King. "An automatic reordering scheme for simultaneous equations derived from network systems". In: *International Journal for Numerical Methods in Engineering* 2.4 (1970), pp. 523–533 (cited on pages 37, 55).

[Kle+16] Paul Klemm et al. "3D Regression Heat Map Analysis of Population Study Data". In: 22.1 (2016), pp. 81–90. ISSN: 1077-2626. DOI: 10.1109/TVCG.2015.2468291 (cited on pages 73, 74, 76).

[Kny01] Andrew V. Knyazev. "Toward the Optimal Preconditioned Eigensolver: Locally Optimal Block Preconditioned Conjugate Gradient Method". In: *SIAM Journal on Scientific Computing* 23.2 (2001), pp. 517–541. DOI: 10.1137/S1064827500366124. URL: http://dx.doi.org/10.1137/S1064827500366124 (cited on page 45).

[Ko+15] Sungahnn Ko et al. "Analyzing high-dimensional multivariate network links with integrated anomaly detection, highlighting and exploration". In: *2014 IEEE Conference on Visual Analytics Science and Technology, VAST 2014 - Proceedings* (2015), pp. 83–92. DOI: 10.1109/VAST.2014.7042484 (cited on page 73).

[Ko+12] S. Ko et al. "MarketAnalyzer: An Interactive Visual Analytics System for Analyzing Competitive Advantage Using Point of Sale Data". In: *Computer Graphics Forum* 31.3pt3 (June 2012), pp. 1245–1254. ISSN: 01677055. DOI: 10.1111/j.1467-8659.2012.03117.x. URL: http://doi.wiley.com/10.1111/j.1467-8659.2012.03117.x (cited on page 76).

[Koh88] Teuvo Kohonen. *Self-organization and associative memory*. 2te. Springer series in information sciences. Berlin: Springer, 1988. ISBN: 3-540-18314-0 (cited on page 189).

[Koh82] Teuvo Kohonen. "Self-organized formation of topologically correct feature maps". In: *Biological Cybernetics* 43.1 (1982), pp. 59–69. ISSN: 1432-0770. DOI: 10.1007/BF00337288. URL: http://dx.doi.org/10.1007/BF00337288 (cited on page 189).

[Koh97] Teuvo Kohonen. *Self-organizing maps*. 2te. Vol. 30. Berlin: Springer, 1997. ISBN: 3-540-62017-6 (cited on page 189).

[KOK05] H. Koike, K. Ohno, and K. Koizumi. "Visualizing cyber attacks using IP matrix". In: *IEEE Workshop on Visualization for Computer Security, 2005. (VizSEC 05).* (2005), pp. 91–98. DOI: 10.1109/VIZSEC.2005.1532070. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1532070 (cited on pages 75, 76).

[Kol+15] S Koldijk et al. "Visual Analytics of Work Behavior Data - Insights on Individual Differences". In: (2015). DOI: 10.2312/eurovisshort.20151129 (cited on pages 73, 75).

[Kor05] Y. Koren. "Drawing Graphs by Eigenvectors: Theory and Practice". In: *Comput. Math. Appl.* 49.11-12 (June 2005), pp. 1867–1888. ISSN: 0898-1221. DOI: 10.1016/j.camwa.2004.08.015. URL: http://dx.doi.org/10.1016/j.camwa.2004.08.015 (cited on page 37).

[KC03] Yehuda Koren and Liran Carmel. “Visualization of Labeled Data Using Linear Transformations”. In: *Proceedings of the IEEE Conference on Visualization*. Washington, DC, USA: IEEE Computer Society, 2003, pp. 121–128 (cited on pages 13, 21).

[KH02a] Yehuda Koren and David Harel. “A Multi-scale Algorithm for the Linear Arrangement Problem”. In: *Revised Papers from the 28th International Workshop on Graph-Theoretic Concepts in Computer Science*. WG ’02. London, UK, UK: Springer-Verlag, 2002, pp. 296–309. ISBN: 3-540-00331-2. URL: http://dl.acm.org/citation.cfm?id=647683.760642 (cited on pages 57, 64).

[KH02b] Yehuda Koren and David Harel. “A Multi-scale Algorithm for the Linear Arrangement Problem”. In: *Revised Papers from the 28th International Workshop on Graph-Theoretic Concepts in Computer Science*. WG ’02. London, UK, UK: Springer-Verlag, 2002, pp. 296–309. ISBN: 3-540-00331-2 (cited on pages 57, 66, 199).

[KC02] Serhiy Kosinov and Terry Caelli. “Inexact Multisubgraph matching using Graph Eigenspace and Clustering Models”. In: *In Proceedings of SSPR/SPR*. Springer, 2002, pp. 133–142 (cited on pages 113, 114, 133, 134, 167).

[Köt+15] P. Köthur et al. “Visual Analytics for Correlation-Based Comparison of Time Series Ensembles”. In: *Computer Graphics Forum* 34.3 (2015), pp. 411–420. ISSN: 01677055. DOI: 10.1111/cgf.12653. URL: http://doi.wiley.com/10.1111/cgf.12653 (cited on page 73).

[Lam11] Walter Marcelo Lamagna. “An integrated visualization on network events VAST 2011 mini challenge 2 award: “Outstanding integrated overview display””. In: *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2011), pp. 319–321. DOI: 10.1109/VAST.2011.6102493. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6102493 (cited on pages 75, 76).

[LS09] Tatiana Von Landesberger and Tobias Schreck. “Visual Analysis of Graphs with Multiple Connected Components”. In: (2009), pp. 155–162 (cited on page 72).

[LGS09] Tatiana von Landesberger, Melanie Görner, and Tobias Schreck. “Visual analysis of graphs with multiple connected components”. In: *Proc. IEEE Symposium on Visual Analytics Science and Technology*. IEEE Computer Society. 2009, pp. 155–162 (cited on pages 82, 114).

[Lan+10] Tatiana von Landesberger et al. “Smart query definition for content-based search in large sets of graphs”. In: *Proc. Int. Symposium on Visual Analytics Science and Technology (EuroVAST)*. Peer-reviewed short paper. Eurographics Association. 2010, pp. 7–12 (cited on page 160).

[Lan+11] Tatiana von Landesberger et al. “Visual analysis of large graphs: State-of-the-art and future research challenges”. In: *Wiley-Blackwell Computer Graphics Forum* (2011) (cited on page 80).

[L+02] Laura Lazzeroni, Art Owen, et al. “Plaid models for gene expression data”. In: *Statistica sinica* 12.1 (2002), pp. 61–86 (cited on pages 37, 60, 62).

[LeC+98] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324 (cited on pages 137, 138).

[Lee+06] Bongshin Lee et al. “Task taxonomy for graph visualization”. In: *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*. ACM. 2006, pp. 1–5 (cited on pages 25, 26, 70).

[LZC11] Yong Jae Lee, C. Lawrence Zitnick, and Michael F. Cohen. “ShadowDraw: Real-time User Guidance for Freehand Drawing”. In: *ACM Transactions on Graphics* 30.4 (2011), 27:1–27:10. DOI: 10.1145/1964921.1964922. URL: http://doi.acm.org/10.1145/1964921.1964922 (cited on pages 157, 160).

[Leh+15] Dirk J Lehmann et al. “Visualnostics: Visual Guidance Pictograms for Analyzing Projections of High-dimensional Data”. In: *Computer Graphics Forum*. Vol. 34. 3. Wiley Online Library. 2015, pp. 291–300 (cited on page 117).

[Lei+15] Jason Leigh et al. “Tell Me What Do You See : Detecting Perceptually-Separable Visual Patterns via Clustering of Image-Space Features in Visualizations”. In: (2015), pp. 3–4 (cited on page 73).

[LK75] Jan K Lenstra and AHG Rinnooy Kan. “Some simple applications of the travelling salesman problem”. In: *Operational Research Quarterly* (1975), pp. 717–733 (cited on pages 37, 57, 58).

[Len74] JK Lenstra. “Technical Note—Clustering a Data Array and the Traveling-Salesman Problem”. In: *Operations Research* 22.2 (1974), pp. 413–414 (cited on pages 37, 58).

[LBK09] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. “Meme-tracking and the dynamics of the news cycle”. In: *KDD ’09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. Paris, France: ACM, 2009, pp. 497–506. ISBN: 978-1-60558-495-9. DOI: http://doi.acm.org/10.1145/1557019.1557077 (cited on page 82).

[LVW84] Joseph YT Leung, Oliver Vornberger, and James D Witthoff. “On some variants of the bandwidth minimization problem”. In: *SIAM Journal on Computing* 13.3 (1984), pp. 650–667 (cited on pages 37, 56).

[Lex+10] Alexander Lex et al. “Comparative Analysis of Multidimensional , Quantitative Data”. In: 16.6 (2010), pp. 1027–1035 (cited on page 34).

[Lex+14] Alexander Lex et al. “UpSet: Visualization of intersecting sets”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1983–1992. ISSN: 10772626. DOI: 10.1109/TVCG.2014.2346248 (cited on pages 37, 76).

[Lex+12] a. Lex et al. “StratomeX: Visual Analysis of Large-Scale Heterogeneous Genomics Data for Cancer Subtype Characterization”. In: *Computer Graphics Forum* 31.3pt3 (June 2012), pp. 1175–1184. ISSN: 01677055. DOI: 10.1111/j.1467-8659.2012.03110.x. URL: http://doi.wiley.com/10.1111/j.1467-8659.2012.03110.x (cited on pages 34–36, 76).

[LZM15] Jie Li, Kang Zhang, and Zhao Peng Meng. “Vismate: Interactive visual analysis of station-based observation data on climate changes”. In: *2014 IEEE Conference on Visual Analytics Science and Technology, VAST 2014 - Proceedings* (2015), pp. 133–142. DOI: 10.1109/VAST.2014.7042489 (cited on pages 72, 75).

[Lic13] M. Lichman. *UCI Machine Learning Repository*. 2013. URL: http://archive.ics.uci.edu/ml (cited on page 151).

[Lii10] Innar Liiv. “Seriation and matrix reordering methods: An historical overview”. In: *Statistical Analysis and Data Mining* 3.2 (2010), pp. 70–91. ISSN: 1932-1872. DOI: 10.1002/sam.10071 (cited on pages 163, 164).

[Liu+03] Li Liu et al. “Robust singular value decomposition analysis of microarray data”. In: *Proceedings of the National Academy of Sciences* 100.23 (2003), pp. 13167–13172 (cited on pages 37, 47, 49).

[LS76] Wai-Hung Liu and Andrew H Sherman. “Comparative analysis of the Cuthill-McKee and the reverse Cuthill-McKee ordering algorithms for sparse matrices”. In: *SIAM Journal on Numerical Analysis* 13.2 (1976), pp. 198–213 (cited on pages 37, 55).

[LKS05] Levon Lloyd, Dimitrios Kechagias, and Steven Skiena. “Lydia: A System for Large-Scale News Analysis”. In: *String Processing and Information Retrieval: 12th International Conference, SPIRE 2005, Buenos Aires, Argentina, November 2-4, 2005: Proceedings*. 2005, pp. 161–166 (cited on page 82).

[Low04] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110 (cited on page 113).

[Loz+13] Manuel Lozano et al. “A hybrid metaheuristic for the cyclic antibandwidth problem”. In: *Knowledge-Based Systems* 54 (2013), pp. 103–113 (cited on pages 37, 56).

[Loz+12] Manuel Lozano et al. “Variable neighborhood search with ejection chains for the antibandwidth problem”. In: *Journal of Heuristics* 18.6 (2012), pp. 919–938 (cited on pages 37, 56).

[Lub+15] M Luboschik et al. “Feature-Driven Visual Analytics of Chaotic Parameter-Dependent Movement”. In: 34.3 (2015). ISSN: 01677055. DOI: 10.1111/cgf.12654 (cited on pages 73, 76).

[LC08] Mathias Lux and Savvas A. Chatzichristofis. “Lire: Lucene Image Retrieval: An Extensible Java CBIR Library”. In: *Proceedings of the 16th ACM International Conference on Multimedia*. MM ’08. Vancouver, British Columbia, Canada: ACM, 2008, pp. 1085–1088. ISBN: 978-1-60558-303-7. DOI: 10.1145/1459359.1459577. URL: http://doi.acm.org/10.1145/1459359.1459577 (cited on page 121).

[LPM10] Mathias Lux, Arthur Pitman, and Oge Marques. “Callisto: Tag Recommendations by Image Content”. In: *WISMA 2010* (2010), p. 87 (cited on page 121).

[Ma+15] Chihua Ma et al. “Visualizing Dynamic Brain Networks Using an Animated”. In: (2015). DOI: 10.2312/eurovisshort.20151128 (cited on pages 73, 75).

[MH08] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9 (Nov. 2008), pp. 2579–2605. URL: http://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf (cited on page 135).

[Mac+03] Alan Maceachren et al. "Exploring High-D Spaces with Multiform Matrices and Small Multiples." In: *IEEE Conference on Information Visualization : an International Conference on Computer Visualization & Graphics, proceedings ... IEEE Conference on Information Visualization* (Jan. 2003), pp. 31–38. DOI: 10.1109/INFVIS.2003.1249006. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3176663%5C&tool=pmcentrez%5C&rendertype=abstract (cited on pages 34, 36, 72, 76).

[Mac86] Jock Mackinlay. "Automating the design of graphical presentations of relational information". In: *Acm Transactions On Graphics (Tog)* 5.2 (1986), pp. 110–141 (cited on page 8).

[Mac67] J. MacQueen. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press, 1967, pp. 281–297. URL: http://projecteuclid.org/euclid.bsmsp/1200512992 (cited on page 147).

[MO04] Sara C Madeira and Arlindo L Oliveira. "Biclustering algorithms for biological data analysis: a survey". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 1.1 (2004), pp. 24–45 (cited on pages 60, 62).

[Maf14] Liviu Octavian Mafteiu-Scai. "The Bandwidths of a Matrix. A Survey of Algorithms". In: *Annals of West University of Timisoara-Mathematics* 52.2 (2014), pp. 183–223 (cited on pages 53, 58).

[MS00] Erkki Mäkinen and Harri Siirtola. "Reordering the reorderable matrix as an algorithmic problem". In: *Theory and Application of Diagrams*. Springer, 2000, pp. 453–468 (cited on pages 37, 53).

[MS05] Erkki Mäkinen and Harri Siirtola. "The barycenter heuristic and the reorderable matrix". In: *Informatica (Slovenia)* 29.3 (2005), pp. 357–364 (cited on page 53).

[Mam+13] Gladys MH Mamani et al. "User-driven Feature Space Transformation". In: *Computer Graphics Forum*. Vol. 32. 3pt3. Wiley Online Library. 2013, pp. 291–299 (cited on page 160).

[MJS07] Gurmeet Singh Manku, Arvind Jain, and Anish Das Sarma. "Detecting near-duplicates for web crawling". In: *WWW '07: Proceedings of the 16th international conference on World Wide Web*. Banff, Alberta, Canada: ACM, 2007, pp. 141–150. ISBN: 978-1-59593-654-7. DOI: http://doi.acm.org/10.1145/1242572.1242592 (cited on page 83).

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008. ISBN: 0521865719, 9780521865715 (cited on page 182).

[MS93] Gary Marchionini and Ben Shneiderman. “3.1 Finding facts vs. browsing knowledge in hypertext systems”. In: *Sparks of innovation in human-computer interaction* (1993), p. 103 (cited on page 156).

[May+11] Thorsten May et al. “Guiding feature subset selection with an interactive visualization”. In: *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2011), pp. 111–120. DOI: 10.1109/VAST.2011.6102448. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6102448 (cited on page 76).

[MSW72] William T. McCormick, Paul J. Schweitzer, and Thomas W. White. “Problem Decomposition and Data Reorganization by a Clustering Technique”. In: *Operations Research* 20.5 (1972), pp. 993–1009. DOI: 10.1287/opre.20.5.993. eprint: http://dx.doi.org/10.1287/opre.20.5.993. URL: http://dx.doi.org/10.1287/opre.20.5.993 (cited on pages 37, 52).

[McC+69] William T McCormick et al. *Identification of Data Structures and Relationships by Matrix Reordering Techniques*. Tech. rep. DTIC Document, 1969 (cited on pages 37, 52).

[McK11] McKinsey Global Institute. *Big data: The next frontier for innovation, competition, and productivity*. Online. May 2011. URL: http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation (cited on page 181).

[McQ68] Louis L McQuitty. “Multiple clusters, types, and dimensions from iterative intercolumnar correlational analysis”. In: *Multivariate Behavioral Research* 3.4 (1968), pp. 465–477 (cited on pages 37, 45).

[Mey+10] M Meyer et al. “Pathline : A Tool For Comparative Functional Genomics”. In: 29.3 (2010) (cited on pages 75, 76).

[MGW11] M.a. Migut, J.C. van Gemert, and M. Worring. “Interactive decision making using dissimilarity to visually represented prototypes”. In: *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)* (Oct. 2011), pp. 141–149. DOI: 10.1109/VAST.2011.6102451. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6102451 (cited on pages 36, 76).

[Mig10] Malgorzata Migut. “Visual Exploration of Classification Models for Risk Assessment”. In: (2010), pp. 11–18 (cited on page 75).

[Mil+97] Nancy Miller et al. “The Need for Metrics in Visual Information Analysis”. In: *Proceedings of the ACM Workshop on New Paradigms in Information Visualization and Manipulation*. New York, NY, USA: ACM, 1997, pp. 24–28 (cited on page 21).

[Mor+15] Dominik Moritz et al. “Perfopticon: Visual Query Analysis for Distributed Databases”. In: *Computer Graphics Forum* 34.3 (2015), pp. 71–80. ISSN: 01677055. DOI: 10.1111/cgf.12619. URL: http://doi.wiley.com/10.1111/cgf.12619 (cited on pages 73, 75, 76).

[Mor]

[MML07] C. Mueller, B. Martin, and a. Lumsdaine. “Interpreting large visual similarity matrices”. In: *2007 6th International Asia-Pacific Symposium on Visualization* (Feb. 2007), pp. 149–152. DOI: 10.1109/APVIS.2007.329290. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4126233 (cited on page 25).

[Mun+03] Tamara Munzner et al. “TreeJuxtaposer: scalable tree comparison using Focus+Context with guaranteed visibility”. In: *ACM Trans. Graph.* 22.3 (July 2003), pp. 453–462. ISSN: 0730-0301 (cited on page 82).

[MK03] TM Murali and Simon Kasif. “Extracting conserved gene expression motifs from gene expression data.” In: *Pacific Symposium on Biocomputing*. Vol. 8. World Scientific. 2003, pp. 77–88 (cited on pages 37, 61).

[Net] NetworkX. *NetworkX*. https://networkx.github.io/. online, last visited November 19, 2015 (cited on page 65).

[Ng+13] Andrew Ng et al. *UFLDL Tutorial - Deep Learning Tutorial*. online. 2013. URL: http://ufldl.stanford.edu/tutorial/ (cited on page 137).

[Nie05] Stefan Niermann. “Optimizing the ordering of tables with evolutionary computation”. In: *The American Statistician* 59.1 (2005) (cited on pages 37, 53).

[Oel+10] Daniela Oelke et al. “Visual Readability Analysis: How to make your writings easier to read”. In: *Proceedings of IEEE Conference on Visual Analytics Science and Technology (VAST ’10)*. 2010, pp. 123–130 (cited on page 82).

[OLS15] Daniel Osei-Kuffuor, Ruipeng Li, and Yousef Saad. “Matrix reordering using multilevel graph coarsening for ilu preconditioning”. In: *SIAM Journal on Scientific Computing* 37.1 (2015), A391–A419 (cited on page 58).

[PK06] Suzanne M Paley and Peter D Karp. “The pathway tools cellular overview diagram and omics viewer”. In: *Nucleic Acids Research* 34.13 (2006), pp. 3771–3778 (cited on page 147).

[Pap+16] C Papadopoulos et al. “VEEVVIE : Visual Explorer for Empirical Visualization , VR and Interaction Experiments”. In: 22.1 (2016), pp. 111–120. ISSN: 1077-2626. DOI: 10.1109/TVCG.2015.2467954 (cited on pages 73, 76).

[PJW00] Dong Kwon Park, Yoon Seok Jeon, and Chee Sun Won. “Efficient Use of Local Edge Histogram Descriptor”. In: *Proceedings of the 2000 ACM Workshops on Multimedia*. MULTIMEDIA ’00. Los Angeles, California, USA: ACM, 2000, pp. 51–54. ISBN: 1-58113-311-1. DOI: 10.1145/357744.357758. URL: http://doi.acm.org/10.1145/357744.357758 (cited on pages 121, 131).

[POM07] Fernando V. Paulovich, Maria Cristina F. Oliveira, and Rosane Minghim. “The Projection Explorer: A flexible tool for projection-based multidimensional visualization”. In: *Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing - SIBGRAPI*. Belo Horizonte, Brazil: IEEE CS Press, 2007, pp. 27–36. ISBN: 0-7695-2996-8. DOI: http://dx.doi.org/10.1109/SIBGRAPI.2007.39 (cited on pages 113, 135, 167).

[Pel98] M. Pelillo. “A unifying framework for relational structure matching”. In: *Pattern Recognition, 1998. Proc.. Fourteenth Int. Conference on*. Vol. 2. 1998, 1316–1319 vol.2 (cited on page 113).

[PS03] Sriram Pemmaraju and Steven Skiena. *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica ®* New York, NY, USA: Cambridge University Press, 2003. ISBN: 0521806860 (cited on page 135).

[PWR04a] W. Peng, M.O. Ward, and E.A Rundensteiner. "Clutter Reduction in Multi-Dimensional Data Visualization Using Dimension Reordering". In: *IEEE Symposium on Information Visualization, 2004. INFOVIS 2004.* 2004, pp. 89–96 (cited on pages 13, 21, 150).

[PWR04b] Wei Peng, Matthew O Ward, and Elke A Rundensteiner. "Clutter reduction in multi-dimensional data visualization using dimension reordering". In: *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on.* IEEE. 2004, pp. 89–96 (cited on page 112).

[PVF13] C. Perin, R. Vuillemot, and J.-D. Fekete. "SoccerStories: A Kick-off for Visual Soccer Analysis". In: *Vis. and Computer Graphics, IEEE Trans. on* 19.12 (2013), pp. 2506–2515. ISSN: 1077-2626 (cited on page 204).

[Per13] Charles Perin. "CinemAviz". In: (2013) (cited on pages 34, 73, 76).

[PDF14] Charles Perin, Pierre Dragicevic, and Jean-Daniel Fekete. "Revisiting Bertin matrices: New Interactions for Crafting Tabular Visualizations". In: *IEEE Transactions on Visualization and Computer Graphics* (Nov. 2014). DOI: 10.1109/TVCG.2014.2346279. URL: https://hal.inria.fr/hal-01023890 (cited on pages 33, 37, 43, 70, 71).

[Pet03] Jordi Petit. "Experiments on the minimum linear arrangement problem". In: *J. Exp. Algorithmics* 8 (Dec. 2003). ISSN: 1084-6654. DOI: 10.1145/996546.996554. URL: http://doi.acm.org/10.1145/996546.996554 (cited on pages 65, 107, 199).

[Pic09] Christian Pich. *MDSJ: Java Library for Multidimensional Scaling (Version 0.2).* online. Available at http://www.inf.uni-konstanz.de/algo/software/mdsj/. 2009 (cited on pages 50, 192).

[PGU12] Alexander Pilh, Alexander Gribov, and Antony Unwin. "Comparing Clusterings Using Bertin ' s Idea". In: 18.12 (2012), pp. 2506–2515 (cited on page 74).

[PBK10] H Piringer, W Berger, and J Krasser. "HyperMoVal : Interactive Visual Validation of Regression Models for Real-Time Simulation". In: 29.3 (2010) (cited on page 36).

[PM14] Petrica Pop and Oliviu Matei. "An Efficient Metaheuristic Approach for Solving a Class of Matrix Optimization Problems". In: *METAHEURISTICS AND ENGINEERING* (2014), p. 17. URL: http://goo.gl/lusslE (cited on pages 37, 56).

[Pre+06] Amela Prelić et al. "A systematic comparison and evaluation of biclustering methods for gene expression data". In: *Bioinformatics* 22.9 (2006), pp. 1122–1129 (cited on pages 37, 62).

[Rai+15] Renata Georgia Raidou et al. "Visual analytics for the exploration of multi-parametric cancer imaging". In: *2014 IEEE Conference on Visual Analytics Science and Technology, VAST 2014 - Proceedings* Figure 2 (2015), pp. 263–264. DOI: 10.1109/VAST.2014.7042521 (cited on page 73).

[RPC15] Fateme Rajabiyazdi, Charles Perin, and Sheelagh Carpendale. “WES t : Visualizing non-Emergency Surgery Waiting Times”. In: (2015), pp. 4–5 (cited on page 76).

[RC94] Ramana Rao and Stuart K Card. “The table lens: merging graphical and symbolic representations in an interactive focus+ context visualization for tabular information”. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 1994, pp. 318–322 (cited on pages 37, 70, 88).

[Ras+09] André Raspaud et al. “Antibandwidth and cyclic antibandwidth of meshes and hypercubes”. In: *Discrete Mathematics* 309.11 (2009), pp. 3541–3552 (cited on page 56).

[Ren+05] Pin Ren et al. “IDGraphs : Intrusion Detection and Analysis Using Histographs”. In: *visual security* (2005), pp. 39–46 (cited on pages 37, 72, 76).

[Rob51] William S Robinson. “A method for chronologically ordering archaeological deposits”. In: *American antiquity* 16.4 (1951), pp. 293–301 (cited on page 37).

[RT92] Joseph Lee Rodgers and Tony D Thompson. “Seriation and multidimensional scaling: A data analysis approach to scaling asymmetric proximity matrices”. In: *Applied psychological measurement* 16.2 (1992), pp. 105–117 (cited on pages 37, 50).

[Ros68] Richard Rosen. “Matrix bandwidth minimization”. In: *Proceedings of the 1968 23rd ACM national conference*. ACM. 1968, pp. 585–595 (cited on pages 37, 54).

[RPD10] Edward Rosten, Reid Porter, and Tom Drummond. “FASTER and better: A machine learning approach to corner detection”. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 32 (2010), pp. 105–119. DOI: 10.1109/TPAMI.2008.275. eprint: arXiv:0810.2434[cs.CV]. URL: http://lanl.arXiv.org/pdf/0810.2434 (cited on page 121).

[Rüd15] Laura von Rüden. “Visual Analytics of Parallel-Performance Data: Automatic Identification of Relevant and Similar Data Subsets”. MA thesis. RWTH Aachen University, Apr. 2015 (cited on page 121).

[Rüd+15b] Laura von Rüden et al. “Separating the Wheat from the Chaff: Identifying Relevant and Similar Performance Data with Visual Analytics”. In: *Proc. of the 2nd Workshop on Visual Performance Analysis (VPA), held in conjunction with the Supercomputing Conference (SC15), Austin, TX, USA*. ACM, 2015, 4:1–4:8. ISBN: 978-1-4503-4013-7. DOI: 10.1145/2835238.2835242. URL: http://dl.acm.org/citation.cfm?id=2835242&CFID=563218440&CFTOKEN=69253459 (cited on page 174).

[RHC99] Yong Rui, Thomas S Huang, and Shih-Fu Chang. “Image retrieval: Current techniques, promising directions, and open issues”. In: *Journal of visual communication and image representation* 10.1 (1999), pp. 39–62 (cited on pages 117, 119, 121).

[Rui+98a] Yong Rui et al. “Relevance feedback: a power tool for interactive content-based image retrieval.” In: *IEEE Trans. Circuits Syst. Video Techn.* 8.5 (1998), pp. 644–655. URL: http://dblp.uni-trier.de/db/journals/tcsv/tcsv8.html#RuiHOM98 (cited on page 158).

[Rui+98b] Yong Rui et al. “Relevance feedback: a power tool for interactive content-based image retrieval”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 8.5 (1998), pp. 644–655 (cited on page 161).

[RN03] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 2nd ed. Pearson Education, 2003. ISBN: 0137903952 (cited on page 190).

[Sac+14] Dominik Sacha et al. “Knowledge Generation Model for Visual Analytics”. In: *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visual Analytics Science and Technology 2014)* 20.12 (Dec. 2014), pp. 1604–1613. DOI: 10.1109/TVCG.2014.2346481 (cited on pages 9, 10, 157).

[SGS10] K.E.A. van de Sande, T. Gevers, and C.G.M. Snoek. “Evaluating Color Descriptors for Object and Scene Recognition”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32.9 (Sept. 2010), pp. 1582–1596. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2009.154 (cited on page 121).

[SF83] Alberto Sanfeliu and King-Sun Fu. “A distance measure between attributed relational graphs for pattern recognition”. In: *Systems, Man and Cybernetics, IEEE Trans. on* 3 (1983), pp. 353–362 (cited on page 113).

[SBS11] Maximilian Scherer, Jürgen Bernard, and Tobias Schreck. “Retrieval and exploratory search in multivariate research data repositories using regressional features”. In: *Proceedings of the ACM/IEEE Conference on Digital Libraries*. ACM. 2011, pp. 363–372. URL: http://dl.acm.org/citation.cfm?id=1998144 (cited on page 112).

[SSK06] J. Schneidewind, M. Sips, and D.A. Keim. “Pixnostics: Towards Measuring the Value of Visualization”. In: *Visual Analytics Science And Technology, 2006 IEEE Symposium On*. Oct. 2006, pp. 199–206. DOI: 10.1109/VAST.2006.261423 (cited on pages 23, 113, 117).

[SSK07] Jörn Schneidewind, Mike Sips, and Daniel A. Keim. “An Automated Approach for the Optimization of Pixel-Based Visualizations”. In: *Information Visualization* 6.1 (Mar. 2007), pp. 75–88 (cited on page 21).

[SFK08] Tobias Schreck, Dieter Fellner, and Daniel Keim. “Towards Automatic Feature Vector Optimization for Multimedia Applications”. In: *Proceedings of the ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2008, pp. 1197–1201 (cited on page 22).

[SMT13] M. Sedlmair, T. Munzner, and M. Tory. “Empirical Guidance on Scatterplot and Dimension Reduction Technique Choices”. In: *Visualization and Computer Graphics, IEEE Transactions on* 19.12 (Dec. 2013), pp. 2634–2643. ISSN: 1077-2626 (cited on page 150).

[Sed+14] Michael Sedlmair et al. “Visual Parameter Space Analysis: A Conceptual Framework”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (Dec. 2014), pp. 2161–2170 (cited on page 151).

[Sed+12] M. Sedlmair et al. “RelEx: Visualization for Actively Changing Overlay Network Specifications”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (Dec. 2012), pp. 2729–2738. ISSN: 1077-2626. DOI: 10.1109/TVCG.2012.255. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6327279 (cited on pages 35, 75, 76).

[SH10] Edward Segel and Jeffrey Heer. "Narrative Visualization: Telling Stories with Data". In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (Nov. 2010), pp. 1139–1148. ISSN: 1077-2626 (cited on page 22).

[SS05] Jinwook Seo and Ben Shneiderman. "A rank-by-feature framework for interactive exploration of multidimensional data". In: *Information Visualization* 4.2 (2005), pp. 96–113 (cited on page 23).

[SS04] Jinwook Seo and Ben Shneiderman. "A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections". In: *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*. IEEE. 2004, pp. 65–72 (cited on pages 23, 113).

[Sha+15] L. Shao et al. "Guiding the Exploration of Scatter Plot Data Using Motif-based Interest Measures". In: *IEEE International Symposium on Big Data Visual Analytics*. 2015 (cited on page 113).

[Sha+11] Hossam Sharara et al. "G-PARE: A visual analytic tool for comparative analysis of uncertain graphs". In: *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)* (Oct. 2011), pp. 61–70. DOI: 10.1109/VAST.2011.6102442. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6102442 (cited on page 76).

[SM07] Zeqian Sheny and Kwan-Liu Ma. "Path visualization for adjacency matrices". In: *Proceedings of the 9th Joint Eurographics / IEEE VGTC conference on Visualization*. EUROVIS'07. Norrköping, Sweden: Eurographics Association, 2007, pp. 83–90. ISBN: 978-3-905673-45-6 (cited on page 82).

[Shn96] Ben Shneiderman. "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations". In: *Proceedings of the 1996 IEEE Symposium on Visual Languages*. VL '96. Washington, DC, USA: IEEE Computer Society, 1996, pp. 336–. ISBN: 0-8186-7508-X. URL: http://dl.acm.org/citation.cfm?id=832277.834354 (cited on pages 21, 89, 101, 104, 181).

[SLL01] Jeremy G Siek, Lie-Quan Lee, and Andrew Lumsdaine. *Boost Graph Library: User Guide and Reference Manual, The*. Pearson Education, 2001 (cited on page 65).

[Sii99] Harri Siirtola. "Interaction with the Reorderable Matrix". In: *Proceedings of the 1999 International Conference on Information Visualisation*. IV '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 272–. ISBN: 0-7695-0210-5. URL: http://dl.acm.org/citation.cfm?id=555607.850368 (cited on page 37).

[Sil12] Sabrina A Silveira. "ADVISE : VISUALIZING THE DYNAMICS OF ENZYME ANNOTATIONS IN". In: (2012), pp. 49–56 (cited on pages 34, 74, 76).

[Sip+12] Mike Sips et al. "A Visual Analytics Approach to Multiscale Exploration of Environmental Time Series." In: *IEEE Trans. Vis. Comput. Graph.* 18.12 (2012), pp. 2899–2907 (cited on pages 35, 81).

[Sip+09a] Mike Sips et al. "Selecting Good Views of High-dimensional Data Using Class Consistency". In: *Proceedings of the Eurographics/IEEE VGTC Conference on Visualization*. Berlin, Germany: Eurographics Association, 2009, pp. 831–838 (cited on page 22).

[Sip+09b] M. Sips et al. “Selecting good views of high-dimensional data using class consistency”. In: *Computer Graphics Forum* 28.3 (2009), pp. 831–838 (cited on page 158).

[Slo86] S. W. Sloan. “An algorithm for profile and wavefront reduction of sparse matrices”. In: *International Journal for Numerical Methods in Engineering* 23.2 (1986), pp. 239–251. ISSN: 1097-0207. DOI: 10.1002/nme.1620230208. URL: http://dx.doi.org/10.1002/nme.1620230208 (cited on pages 37, 55, 57, 59).

[Slo89] SW Sloan. “A FORTRAN program for profile and wavefront reduction”. In: *International Journal for Numerical Methods in Engineering* 28.11 (1989), pp. 2651–2679 (cited on pages 37, 57).

[Sme+00] Arnold WM Smeulders et al. “Content-based image retrieval at the end of the early years”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.12 (2000), pp. 1349–1380 (cited on pages 117, 119).

[SLM96] TG Smith, GD Lange, and WB Marks. “Fractal methods and results in cellular morphology—dimensions, lacunarity and multifractals”. In: *Journal of neuroscience methods* 69.2 (1996), pp. 123–136 (cited on page 121).

[Sno+11] Tristan Mark Snowsill et al. “Refining causality: who copied from whom?” In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD ’11. San Diego, California, USA: ACM, 2011, pp. 466–474. ISBN: 978-1-4503-0813-7 (cited on page 82).

[Son+14] Hyunjoo Song et al. “Comparative Gaze Analysis Framework for Volumetric Medical Images”. In: *IEEE Information Visualization* (2014), pp. 5–6 (cited on page 72).

[Son+12] Hyunjoo Song et al. “DiffMatrix : Matrix-based Interactive Visualization for Comparing Temporal Trends”. In: (2012), pp. 103–107. DOI: 10.2312/PE/EuroVisShort/EuroVisShort2012/103-107 (cited on page 73).

[SG74] Ian Spence and Jed Graef. “The determination of the underlying dimensionality of an empirically obtained matrix of proximities”. In: *Multivariate Behavioral Research* 9.3 (1974), pp. 331–341 (cited on pages 37, 49, 50).

[SBB96] Michael Spenke, Christian Beilken, and Thomas Berlage. “FOCUS: the interactive table for product comparison and selection”. In: *Proceedings of the 9th annual ACM symposium on User interface software and technology*. ACM. 1996, pp. 41–50 (cited on page 70).

[SW68] K Steiglitz and P Weiner. “Some improved algorithms for computer solution of the traveling salesman problem”. In: (1968) (cited on page 58).

[Ste65] Haim Sternin. *Statistical Methods of Time Sequencing*. Tech. rep. DTIC Document, 1965 (cited on pages 37, 45).

[Str+12a] Hendrik Strobelt et al. “HiTSEE KNIME: a visualization tool for hit selection and analysis in high-throughput screening experiments for the KNIME platform”. en. In: *BMC Bioinformatics* 13.8 (Dec. 2012), pp. 1–13. (Visited on 04/09/2014) (cited on page 22).

[Str+12b] Hendrik Strobelt et al. “HiTSEE KNIME: a visualization tool for hit selection and analysis in high-throughput screening experiments for the KNIME platform”. In: *BMC Bioinformatics* 13.Suppl 8 (2012), S4. ISSN: 1471-2105. DOI: 10.1186/1471-2105-13-S8-S4. URL: http://www.biomedcentral.com/1471-2105/13/S8/S4 (cited on page 147).

[TMY78] H. Tamura, S. Mori, and T. Yamawaki. “Textural Features Corresponding to Visual Perception”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 8.6 (June 1978), pp. 460–473. ISSN: 0018-9472. DOI: 10.1109/TSMC.1978.4309999 (cited on page 121).

[TSS05] Amos Tanay, Roded Sharan, and Ron Shamir. “Biclustering algorithms: A survey”. In: *Handbook of computational molecular biology* 9.1-20 (2005), pp. 122–124 (cited on pages 43, 62).

[Tao+06] Dacheng Tao et al. “Asymmetric Bagging and Random Subspace for Support Vector Machines-Based Relevance Feedback in Image Retrieval.” In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 28.7 (Aug. 23, 2006), pp. 1088–1099. URL: http://dblp.uni-trier.de/db/journals/pami/pami28.html#TaoTLW06 (cited on page 158).

[Tat+12a] Aditya Tatu et al. “Subspace search and visualization to make sense of alternative clusterings in high-dimensional data”. In: *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*. IEEE. 2012, pp. 63–72 (cited on page 112).

[Tat+11a] Andrada Tatu et al. “Automated Analytical Methods to Support Visual Exploration of High-Dimensional Data”. In: *IEEE Trans. Vis. Comput. Graph.* 17.5 (2011), pp. 584–597 (cited on page 112).

[Tat+11b] Andrada Tatu et al. “Automated Analytical Methods to Support Visual Exploration of High-Dimensional Data”. In: *IEEE Trans. Vis. Comput. Graph.* 17.5 (2011), pp. 584–597 (cited on page 158).

[Tat+12b] Andrada Tatu et al. “Subspace Search and Visualization to Make Sense of Alternative Clusterings in High-Dimensional Data”. In: *Procedings of IEEE VAST*. IEEE CS Press, 2012, pp. 63–72 (cited on page 158).

[Tat+10] Andrada Tatu et al. “Visual Quality Metrics and Human Perception: An Initial Study on 2D Projections of Large Multidimensional Data”. In: *Proceedings of the International Conference on Advanced Visual Interfaces*. New York, NY, USA: ACM, 2010, pp. 49–56 (cited on pages 21, 23).

[Tat+11c] A. Tatu et al. “Automated Analytical Methods to Support Visual Exploration of High-Dimensional Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.5 (May 2011), pp. 584–597 (cited on page 21).

[Tat+09] A. Tatu et al. “Combining automated analysis and visualization techniques for effective exploration of high-dimensional data”. In: *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*. 2009, pp. 59–66. DOI: 10.1109/VAST.2009.5332628. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5332628 (cited on page 22).

[Tat+12c] A. Tatu et al. “Subspace search and visualization to make sense of alternative clusterings in high-dimensional data”. In: *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*. Oct. 2012, pp. 63–72 (cited on page 23).

[Tor+11] Thomas Torsney-Weir et al. “Tuner: principled parameter finding for image segmentation algorithms using visual response surface exploration.” In: *IEEE transactions on visualization and computer graphics* 17.12 (Dec. 2011), pp. 1892–901. ISSN: 1941-0506. DOI: 10.1109/TVCG.2011.248. URL: http://www.ncbi.nlm.nih.gov/pubmed/22034306 (cited on pages 72, 76).

[TM04] M. Tory and T. Moller. “Rethinking Visualization: A High-Level Taxonomy”. In: *Proceedings of the IEEE Symposium on Information Visualization*. 2004, pp. 151–158 (cited on page 22).

[Tri+15] John A. Triana et al. “VafusQ: A Visual Analytics Application with Data Quality Features to Support the Urban Planning Process”. In: *Visualization in Environmental Sciences 2015 workshop* (2015). DOI: 10.2312/envirvis.20151091 (cited on page 76).

[TG83] Edward R Tufte and PR Graves-Morris. *The visual display of quantitative information*. Vol. 2. Cheshire, CT: Graphics Press, 1983 (cited on page 21).

[TBK05] Heather Turner, Trevor Bailey, and Wojtek Krzanowski. “Improved biclustering of microarray data demonstrated through systematic performance tests”. In: *Computational statistics & data analysis* 48.2 (2005), pp. 235–254 (cited on pages 37, 60).

[Ume88] S. Umeyama. “An eigendecomposition approach to weighted graph matching problems”. In: *Pattern Analysis and Machine Intelligence, IEEE Trans. on* 10.5 (1988), pp. 695–703. ISSN: 0162-8828 (cited on page 113).

[Ung+12] Andrea Unger et al. “A Visual Analysis Concept for the Validation of Geoscientific Simulation Models”. In: 18.12 (2012), pp. 2216–2225 (cited on page 72).

[VP09] Frank Van Ham and Adam Perer. “[[[ERROR FOR PACKAGE inputenc]]]Search, show context, expand on demand[[[ERROR FOR PACKAGE inputenc]]]: Supporting large graph exploration with degree-of-interest”. In: *Visualization and Computer Graphics, IEEE Transactions on* 15.6 (2009), pp. 953–960 (cited on page 89).

[Veh+11] Corinna Vehlow et al. “iHAT: Interactive hierarchical aggregation table”. In: *2011 IEEE Symposium on Biological Data Visualization (BioVis)*. (Oct. 2011), pp. 63–69. DOI: 10.1109/BioVis.2011.6094049. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6094049 (cited on page 75).

[Veh+12] Corinna Vehlow et al. “Uncertainty-Aware Visual Analysis of Biochemical Reaction Networks”. In: (2012), pp. 91–98 (cited on pages 72, 76).

[Via+10] Christophe Viau et al. “The FlowVizMenu and Parallel Scatterplot Matrix : Hybrid Multidimensional Visualizations for Network Exploration”. In: 16.6 (2010), pp. 1100–1108 (cited on pages 34, 36, 37, 76).

[War02] Matthew O. Ward. "A Taxonomy of Glyph Placement Strategies for Multidimensional Data Visualization". In: *Information Visualization* 1.3/4 (2002), pp. 194–210. ISSN: 1473-8716. DOI: 10.1057/palgrave.ivs.9500025. URL: http://dx.doi.org/10.1057/palgrave.ivs.9500025 (cited on page 79).

[WG11] Matthew O. Ward and Zhenyu Guo. "Visual Exploration of Time-Series Data with Shape Space Projections". In: *Comput. Graph. Forum* 30.3 (2011), pp. 701–710 (cited on page 158).

[WR04] M.O. Ward and E.a. Rundensteiner. "Clutter Reduction in Multi-Dimensional Data Visualization Using Dimension Reordering". In: *IEEE Symposium on Information Visualization* (2004), pp. 89–96. DOI: 10.1109/INFVIS.2004.15. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1382895 (cited on page 74).

[War+02] Colin Ware et al. "Cognitive Measurements of Graph Aesthetics". In: *Information Visualization* 1.2 (June 2002), pp. 103–110 (cited on page 21).

[Wat91] David S. Watkins. *Fundamentals of Matrix Computations*. New York, NY, USA: John Wiley & Sons, Inc., 1991. ISBN: 0-471-61414-9 (cited on page 45).

[WL15] Kodzo Webga and Aidong Lu. "Discovery of Rating Fraud with Real-Time Streaming Visual Analytics". In: (2015) (cited on page 76).

[Wei13] Taiyun Wei. *Corrplot: Visualization of a correlation matrix*. R package version 0.73. Oct. 2013. URL: https://github.com/taiyun/corrplot (cited on pages 64, 65).

[Wes12] Marcos Weskamp. *Newsmap*. http://newsmap.jp/. Online; accessed 25-February-2012. 2012 (cited on page 82).

[WR09] Ryen W. White and Resa A. Roth. *Exploratory Search: Beyond the Query-Response Paradigm*. San Rafael, CA: Morgan & Claypool Publishers, 2009. ISBN: 978-1-59829-783-6 (cited on page 79).

[WAG05a] L. Wilkinson, A. Anand, and R. Grossman. "Graph-theoretic scagnostics". In: *Proceedings of the IEEE Symposium on Information Visualization*. Oct. 2005, pp. 157–164 (cited on pages 13, 21).

[Wil05] Leland Wilkinson. *The Grammar of Graphics (Statistics and Computing)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005. ISBN: 0387245448 (cited on pages 11, 25, 37, 42, 47, 51, 53).

[WAG05b] Leland Wilkinson, Anushka Anand, and Robert Grossman. "Graph-Theoretic Scagnostics". In: *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*. IEEE Computer Society, 2005 (cited on pages 158, 186).

[WAG06] Leland Wilkinson, Anushka Anand, and Robert Grossman. "High-dimensional visual analytics: interactive exploration guided by pairwise views of point distributions." In: *IEEE Transactions on Visualization and Computer Graphics* 12.6 (2006), pp. 1363–72. ISSN: 1077-2626. DOI: 10.1109/TVCG.2006.94. URL: http://www.ncbi.nlm.nih.gov/pubmed/17073361 (cited on page 79).

[WAG05c] Leland Wilkinson, Anushka Anand, and Robert L Grossman. "Graph-Theoretic Scagnostics." In: *INFOVIS*. Vol. 5. 2005, p. 21 (cited on pages 113, 117).

[WH97] R.C. Wilson and E.R. Hancock. “Structural matching by discrete relaxation”. In: *Pattern Analysis and Machine Intelligence, IEEE Trans. on* 19.6 (1997), pp. 634–648. ISSN: 0162-8828 (cited on page 113).

[Wis+95] James A. Wise et al. “Visualizing the non-visual: spatial analysis and interaction with information from text documents”. In: *Proceedings of the IEEE Symposium on Information Visualization*. 1995, pp. 51–58 (cited on page 79).

[Won05] Chee Sun Won. “Advances in Multimedia Information Processing - PCM 2004: 5th Pacific Rim Conference on Multimedia, Tokyo, Japan, November 30 - December 3, 2004. Proceedings, Part III”. In: ed. by Kiyoharu Aizawa, Yuichi Nakamura, and Shin’ichi Satoh. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. Chap. Feature Extraction and Evaluation Using Edge Histogram Descriptor in MPEG-7, pp. 583–590. ISBN: 978-3-540-30543-9. DOI: 10.1007/978-3-540-30543-9_73. URL: http://dx.doi.org/10.1007/978-3-540-30543-9_73 (cited on page 131).

[Won+13] Pak Chung Wong et al. “Visual Matrix Clustering of Social Networks”. In: *Computer Graphics and Applications, IEEE* 33.4 (July 2013), pp. 88–96. ISSN: 0272-1716. DOI: 10.1109/MCG.2013.66 (cited on page 59).

[Wu+15] W. Wu et al. “TelCoVis: Visual Exploration of Co-occurrence in Urban Human Mobility Based on Telco Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2015), pp. 1–1. ISSN: 1077-2626. DOI: 10.1109/TVCG.2015.2467194. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7192730 (cited on pages 73, 76).

[WYM12] Yingcai Wu, Guo-Xun Yuan, and Kwan-Liu Ma. “Visualizing Flow of Uncertainty through Analytical Processes”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (Dec. 2012), pp. 2526–2535. ISSN: 1077-2626. DOI: 10.1109/TVCG.2012.285. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6327258 (cited on pages 34, 35).

[WW02] BarendJacobus Wyk and Micha[[[ERROR FOR PACKAGE inputenc]]]lAntonie Wyk. “Non-bayesian Graph Matching without Explicit Compatibility Calculations”. English. In: *Structural, Syntactic, and Statistical Pattern Recognition*. Vol. 2396. Lecture Notes in Computer Science. Springer Heidelberg, 2002, pp. 74–83. ISBN: 978-3-540-44011-6 (cited on page 113).

[Yal+16] M Adil Yal et al. “AggreSet : Rich and Scalable Set Exploration using Visualizations of Element Aggregations”. In: 22.1 (2016), pp. 688–697. ISSN: 1077-2626. DOI: 10.1109/TVCG.2015.2467051 (cited on pages 75, 76).

[Yan+07a] Di Yang et al. “Managing discoveries in the visual analytics process”. In: *SIGKDD Explor. Newsl.* 9.2 (Dec. 2007), pp. 22–29. ISSN: 1931-0145. DOI: 10.1145/1345448.1345453. URL: http://doi.acm.org/10.1145/1345448.1345453 (cited on page 113).

[YF12] Hang Yang and Simon Fong. “Incrementally Optimized Decision Tree for Noisy Big Data”. In: *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*. BigMine ’12. Beijing, China: ACM, 2012, pp. 36–44. ISBN: 978-1-4503-1547-0. DOI: 10.1145/2351316.2351322. URL: http://doi.acm.org/10.1145/2351316.2351322 (cited on page 188).

[Yan+99] Jing Yang et al. “Interactive Hierarchical Dimension Ordering , Spacing and Filtering for Exploration of High Dimensional Datasets”. In: (1999), pp. 105–112 (cited on page 35).

[Yan+03] Jing Yang et al. “Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets”. In: *IEEE Symposium on Information Visualization, 2003. INFOVIS 2003.* Oct. 2003, pp. 105–112 (cited on page 150).

[Yan+07b] Jing Yang et al. “Value and Relation Display: Interactive Visual Exploration of Large Data Sets with Hundreds of Dimensions”. In: *IEEE Trans. Vis. Comput. Graph.* 13.3 (2007), pp. 494–507 (cited on page 158).

[YKR08] Mingqiang Yang, Kidiyo Kpalma, and Joseph Ronsin. *A Survey of Shape Feature Extraction Techniques*. Peng-Yeng Yin, 2008, pp. 43–90 (cited on page 121).

[Yat+14] A. Yates et al. “Visualizing Multidimensional Data with Glyph SPLOMs”. In: *Computer Graphics Forum* 33.3 (2014), pp. 301–310. ISSN: 1467-8659 (cited on page 151).

[Yeg09] B Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009 (cited on page 190).

[You+13] H. Younesy et al. “An Interactive Analysis and Exploration Tool for Epigenomic Data”. In: *Computer Graphics Forum* 32.3pt1 (June 2013), pp. 91–100. ISSN: 01677055. DOI: 10.1111/cgf.12096. URL: http://doi.wiley.com/10.1111/cgf.12096 (cited on pages 35, 73, 76).

[YWB74] Ian T Young, Joseph E Walker, and Jack E Bowie. “An analysis technique for biological shape. I”. In: *Information and control* 25.4 (1974), pp. 357–370 (cited on page 121).

[Yua+13] Xiaoru Yuan et al. “Dimension projection matrix/tree: interactive subspace visual exploration and analysis of high dimensional data.” In: *IEEE transactions on visualization and computer graphics* 19.12 (Dec. 2013), pp. 2625–33. ISSN: 1941-0506. DOI: 10.1109/TVCG.2013.150. URL: http://www.ncbi.nlm.nih.gov/pubmed/24051829 (cited on pages 34, 75).

[Zha08] J. Zhang. *Visualization for Information Retrieval*. Springer, 2008 (cited on page 79).

[ZWS96] Kaizhong Zhang, Jason Tsong-Li Wang, and Dennis Shasha. “On the Editing Distance Between Undirected Acyclic Graphs”. In: *Int. J. Found. Comput. Sci.* 7.1 (1996), pp. 43–58 (cited on page 113).

[Z+11] Paul Zikopoulos, Chris Eaton, et al. *Understanding big data: Analytics for enterprise class hadoop and streaming data*. McGraw-Hill Osborne Media, 2011 (cited on page 79).