Informatics

Visual Analytics for Blind Source Separation in Time and Space

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

eingereicht von

Dipl.-Medieninf. Nikolaus Piccolotto
Matrikelnummer 11937119

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Prof. Dr. Silvia Miksch

Diese Dissertation haben begutachtet:

Michael Sedlmair

Cagatay Turkay

Wien, 17. Mai 2024

Nikolaus Piccolotto

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Informatics

Visual Analytics for Blind Source Separation in Time and Space

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Dipl.-Medieninf. Nikolaus Piccolotto
Registration Number 11937119

to the Faculty of Informatics

at the TU Wien

Advisor: Prof. Dr. Silvia Miksch

The dissertation has been reviewed by:

Michael Sedlmair	Cagatay Turkay

Vienna, 17th May, 2024

Nikolaus Piccolotto

Erklärung zur Verfassung der Arbeit

Dipl.-Medieninf. Nikolaus Piccolotto

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

Wien, 17. Mai 2024

Nikolaus Piccolotto

Acknowledgements

First, I must thank Silvia for giving me the opportunity to research and even more for committing to the decision and finding ways to support me when reviews were terrible or funding scarce. But, as she likes to stress, "it's a team effort," and there were others without whom I would not have come this far.

Thanks to my friends and my partner, who provided much of the desperately needed emotional support and volunteered to proofread my confusingly written drafts.

Thanks to my parents and grandparents, who made the person I am today (figuratively and literally).

Thanks to our cooperation partners Christoph, Klaus and Peter from the computational statistics group, who informed so much of this thesis and always made time for discussions.

Thanks to my CVAST colleagues, who helped me with all the everyday issues and taught me so much about research: Alessio, Christian, Davide, Ignacio, Markus, Michaela, Roger, Theresia, Velitchko, and Victor.

Thanks to all the people who participated in my evaluations. I cannot name you, but you know who you are. There would be no publications without you.

Thanks also to the anonymous reviewers of my papers, without whom there would be no peer review.

Finally, I want to acknowledge that my research was enabled by public funding bodies, like the Austrian Science Fund (FWF) and the EU. Thank you, taxpayers!

Kurzfassung

Immer mehr Daten werden über die Welt gesammelt und wir versuchen, sie sinnvoll zu nutzen. Daten sind oft zeitlich oder räumlich verteilt, d. h. die einzelnen Messungen werden an bestimmten Zeitpunkten oder Orten durchgefüht. Zusätzlich bestehen sie oft aus mehreren Variablen. So wird z. B. die elektrische Aktivität des Herzens durch Elektrokardiogramme untersucht, und die chemische Analyse von Bodenproben kann das Ausmaß der Umweltverschmutzung aufdecken. Bei der ordnungsgemäßen Analyse solcher Daten muss ihr zeitlicher oder räumlicher Charakter berücksichtigt werden, da diese Strukturen von besonderem Interesse sind. Die Analyse mehrerer Variablen beinhaltet die Suche nach niedrigdimensionalen Unterräumen, die wichtige Trends im Datensatz erfassen. Dies wird als Dimensionality Reduction (DR) bezeichnet. Blind Source Separation (BSS) ist ein statistisches Modellierungsframework für DR von zeitlichen oder räumlichen Daten. Die gewonnenen latenten Dimensionen müssen visuell untersucht werden, um etwas über die gemessenen Phänomene zu erfahren, oder sie werden für die Modellierung verwendet. Die größten Herausforderungen für den praktischen Einsatz von BSS sind jedoch der große zu berücksichtigende Parameterraum und die Anzahl der zu analysierenden latenten Dimensionen.

Die Datenvisualisierung ist ein integraler Bestandteil der Datenanalyse, bei der Daten in einer aussagekräftigen grafischen Form dargestellt werden, um die Mustererkennungsfähigkeiten des menschlichen visuellen Systems zu nutzen. Wenn interaktive Visualisierungen mit leistungsstarken Algorithmen zur Datenverwaltung, -analyse und -verarbeitung verbunden werden, spricht man von Visual Analytics (VA). Diese Arbeit schlägt VA Ansätze für BSS vor. Auf etablierten Forschungsmethoden der Visualisierung aufbauend wurden Designstudien im Kontext von BSS für zeitliche und räumliche Daten durchgeführt. Wir betten unsere Arbeit in die breitere Visualisierungsliteratur zur visuellen Parameteranalyse ein. Die Beiträge dieser Arbeit sind i) VA-gestützte Parameteranalyse für Temporal Blind Source Separation (TBSS), einschließlich einer Aufgabenbeschreibung für BSS, ii) VA-gestützte Parameteroptimierung für Spatial Blind Source Separation (SBSS), und iii) VA-gestützte Sensitivitätsanalyse für SBSS. Diese Arbeit stellt folglich einen ersten Schritt zu BSS-gestützter visueller Datenanalyse dar. Die Lösung der besonderen Probleme von BSS zeigte außerdem mögliche zukünftige Forschung in den Bereichen visueller Parameteranalyse, Visualisierung kategorischer Daten, und geovisueller Analyse auf.

Abstract

Ever more data is collected about everything, and we look for ways to make sense of them. Collected data are often temporally or spatially distributed, i.e., associated with points in time or locations in space, and consist of multiple variables. E.g., the heart's electrical activity is investigated in healthcare through electrocardiograms, and chemical analysis of soil samples may uncover the extent of environmental pollution. Proper analysis of such data has to account for their temporal or spatial nature, as these particular structures, e.g., distributions in time or space, are often of special interest. Analyzing multiple variables involves searching for low-dimensional subspaces that still capture major trends in the dataset, which is referred to as Dimensionality Reduction (DR). Blind Source Separation (BSS) is a statistical modeling framework for DR of temporal/spatial data and hence a superior analysis tool compared to temporally/spatially-unaware methods. Obtained latent dimensions are visually inspected to learn about the data-generating phenomena or used for modeling. However, the main challenges inhibiting the use of BSS in practice are a large parameter space to consider and the number of latent dimensions to analyze.

Data visualization is an integral part of data analysis, where data is presented in a meaningful graphical way to harness the pattern recognition abilities of the human visual system. When interactive visualizations are paired with powerful data management, analysis, and processing algorithms, we speak of Visual Analytics (VA). This thesis proposes VA approaches to make BSS usable in practice. Building on established research methods in visualization, we conducted design studies in the context of BSS for temporal and spatial data. We embed our work in the broader visualization literature concerning visual parameter analysis. The advances of this thesis are i) VA-supported parameter analysis for Temporal Blind Source Separation (TBSS), including a task abstraction for BSS, ii) VA-supported parameter optimization for Spatial Blind Source Separation (SBSS), and iii) VA-supported sensitivity analysis for SBSS. Consequently, this thesis presents a first step toward BSS-supported visual data analysis. Solving the particular problems of BSS also uncovered avenues for future research in visual parameter analysis, set visualization, and geovisual analytics.

Contents

CHAPTER 1

Introduction

Many industries and scientific fields collect and analyze multiple variables distributed over time and space. E.g., the electrical activity of the heart is investigated in healthcare through electrocardiograms, fluctuating stock values are of interest to financial analysts, chemical analysis of soil samples may uncover the extent of environmental pollution, and trends of meteorological measurements, such as precipitation and temperature, are relevant to everyone's daily lives and climate research. Specifically, we consider multivariate data associated with time points on a calendar (multivariate time series) or associated with locations in space (multivariate spatial fields). Analysis of such datasets is challenging due to their temporal and spatial characteristics and the multiple variables involved. Because variables are often correlated and measurements are tend to be noisy, multivariate datasets are subjected to Dimensionality Reduction (DR), sometimes called multidimensional projections or low-dimensional embeddings. The idea behind these approaches is that the high-dimensional signal may often be represented well enough by a (transformed) subset of the original dimensions, referred to as latent dimensions. DR approaches thus obtain a lower-dimensional version of the original dataset that still captures high-dimensional structures. Principal Component Analysis (PCA), Multi-Dimensional Scaling (MDS) [Sae+18], or stochastic neighbor embeddings such as t-Distributed Stochastic Neighbor Embedding (t-SNE) [vdMH08] and Uniform Manifold Approximation and Projection (UMAP) [MHM18] are just a few of these widely applied techniques. The resulting latent dimensions may be used for visualization directly or as input to other algorithms [Sed+12]. Whether these DR techniques are linear or nonlinear, stochastic or deterministic, their main drawback for analyzing temporal and spatial data is that they do not incorporate temporal or spatial information.

Blind Source Separation (BSS), on the other hand, is a general multivariate statistical modeling framework [CJ10; YHX14]. Many BSS models were explicitly designed for and are thus suitable for multivariate time series or spatial fields. BSS brings various advantages compared to the aforementioned alternative methods, e.g., the framework is linear and keeps the well-known loading-scores scheme from PCA, which aids interpretation of latent dimensions. BSS also properly accounts for temporal or spatial dependence due to its model-based approach. Ignoring such dependence in DR might be inefficient or miss important structures. Therefore, latent dimensions identified with BSS often correspond to the physical reality of the collected data, which makes it, in theory, a superior analysis tool.

In practice, however, several challenges hinder the effective use of BSS methods, mainly arising from complex tuning parameters to be set. Temporal Blind Source Separation (TBSS), specifically the Generalized Second-Order Blind Identification (gSOBI) method [Mie+20], requires analysts to choose two sets of temporal intervals (lag sets) and a weighting factor between the two. A lag set thus represents a temporal parameter with 2^T possible settings (for a time series of length T). Similarly, Spatial Blind Source Separation (SBSS) parameters have a meaningful spatial extent. The required tuning parameters consist of a kernel, i.e., a distance defining a location's neighborhood, and a regionalization, i.e., a partition of locations into coherent groups [MBN22]. Again it is easy to see that even small datasets induce many possible tuning parameter settings. Usually, in the case of multivariate parameters, this issue is tackled by taking random samples of the parameter space, precomputing the respective outputs, and visually presenting the results or using the data for computational assessments, such as sensitivity analysis [Sed+14]. However, this approach is seen as less productive with BSS. Partly because automatic sampling methods are not straightforward to find for such complex interdependent parameters and, more importantly, because domain knowledge is considered crucial. For instance, a SBSS kernel should be selected to cover the distance within which a latent data-generating process (e.g., precipitation) may be noticeable. Clearly, this is a problem for the human analyst instead of the computer because not only does one need to know which processes could be in the data in principle, but also whether the spatial resolution of the dataset would show it. Despite these challenges, analysts need to perform common parameter space analysis tasks [Sed+14], such as optimization or sensitivity analysis, to successfully employ BSS.

When the analyst overcomes these obstacles, another group of problems arises. Like in multivariate DR approaches, BSS positions data points on latent dimensions (the *scores* in "loading-scores scheme"). Unlike multivariate DR approaches, the data points retain their temporal/spatial order. In the language of PCA: The scores are temporally/spatially distributed. As a consequence, latent dimensions in BSS are read and interpreted as univariate time series or spatial fields. While analysis of such data can be challenging on its own, the main issue arising in BSS is the amount of data to consider. Each execution of BSS yields as many latent dimensions as there are original dimensions (a consequence of BSS model assumptions). These latent dimensions appear in groups, as each is the outcome of a particular parameter setting. Analysts are interested, e.g., in commonly found latent dimensions with high signal (or other combinations of commonality vs. signal) to decide what the "true" underlying processes are. It may also be useful to connect relevant structures found in latent dimensions back to parameter settings, thus further supporting parameter space analysis tasks. Working mainly in the text-based RStudio development environment with the R programming language, all these tasks are cumbersome and time-consuming to carry out.

Given the existing strong focus on visual inspection of BSS results and the mentioned challenges concerning data management, mining, and processing, it seems promising to employ Visual Analytics (VA). VA was defined by Keim et al. as "[combining] automated analysis techniques with interactive visualizations for an effective understanding, reasoning, and decision making on the basis of very large and complex datasets" [Kei+08, p. 157]. The general idea is to automate everything that can be automated and present everything that cannot be by effective interactive visualizations to the human analyst. VA has been already successfully employed to solve many BSS-adjacent problems. E.g., LiveRAC [McL+08] focused on visual analysis of many time series, which is also relevant in TBSS. Attribute Signatures [Tur+14] is a path-based visual exploration idiom for multivariate spatial data, which may be useful for SBSS. Several VA prototypes were developed to support DR, e.g., iPCA [Jeo+09] or t-viSNE [CMK20]. Using various computational models has been simplified by visualization-oriented parameter space analysis, e.g., generators for visual effects [BM10]. Ensemble visualization is a sub-field of visualization that focuses on the analysis of a group of (usually) temporal, spatial, or spatio-temporal objects, such as hurricane trajectories [Liu+15] or precipitation forecasts [Bis+17]. Set visualization [Als+16] can be considered another adjacent visualization discipline, as it focuses on relations between groups of elements.

However, none of the mentioned examples completely capture the peculiarities of BSS-related data and tasks. Ensemble visualization too often focuses on a single group of objects, whereas in BSS, we have to consider many of them. Set visualization does so but only considers simple elements, like strings. VA approaches for DR rarely incorporate time and space the way BSS does. Visual parameter analysis tends to focus on parameters detached from time and space, which is required for TBSS and SBSS. Approaches for temporal and spatial data analysis may be useful for BSS result exploration, but they need to be connected to the former aspects to facilitate the involved BSS analysis tasks.

In the remainder of this chapter we will provide an introduction to visualization/VA (Section 1.1) and BSS (Section 1.1.3), after which we will formulate our research questions (Section 1.2), list our contributions (Section 1.3), and close with an outline of the remainder of the thesis (Section 1.4).

1.1 Background and Methodology

In this section we present historic background on visualization (Section 1.1.1), discuss common models used in visualization research which inform our methodology (Section 1.1.2) and finally intend to give the reader intuition about BSS (Section 1.1.3).

1.1.1 A Brief History of Visualization

Visualization is about getting insights from data. Its history is intertwined with that of statistics, astronomy, and cartography, and goes back as far as the 10th century [Fri08]. Although its use for cognition and inference remains unclear, what is credited as the first line chart [Fun36] depicts the movements of seven celestial bodies over the night sky. If the image was not just meant as a schematic illustration, it is likely that poor observational tools and theory available at the time hindered its accuracy. In the 17th century, quantitative data began to be systematically collected (via government mandates) and became more widespread. The idea of graphical representation of data was established during that time, although being far from ubiquitous as it is today. The following century then saw a variety of new graphical forms applied to new domains. E.g., isolines (contours of equal value) were proposed by Edmund Halley in 1700, thus starting what came to be known later as *thematic mapping*. Later in the 18th century, two pioneers of data graphics introduced new ideas and graphics that are common to this day: Johann Lambert and William Playfair, the latter of which is often seen as the inventor of modern data visualization for his creation of bar charts and pie charts. While the visual means more and more started to resemble what we know today, they were far from obvious at the time. Michael Friendly [Fri08, p. 24] remarks that "[Playfair] devoted several pages of text (...) describing how to read and understand a line graph."

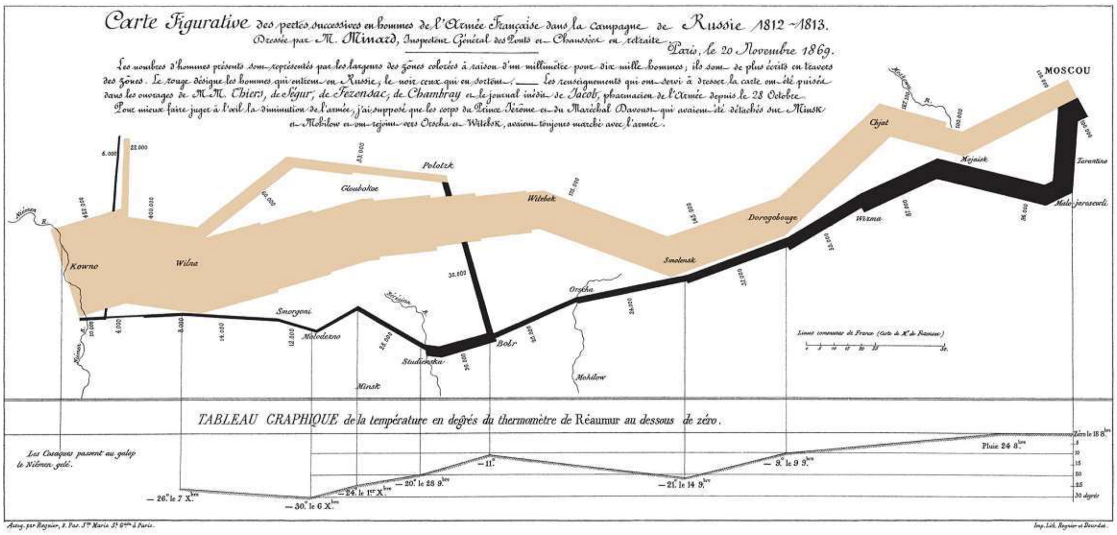


Figure 1.1: Charles Joseph Minard's illustration of Napoleon's losses. Image: [Min70, vue 52]

Nonetheless, several examples can be found from that era where graphics were used to argue, to convince, to making a point. One of the better known, due to the writings of Edward R. Tufte [Tuf01], is Charles Joseph Minard's illustration of Napoleon's losses against Russia in 1812–1813 (Figure 1.1). The image shows the size of Napoleon's army and its movement originating in today's Lithuania. The march to Moscow is depicted as the thick beige line, where thickness encodes the size of the army (initially 422 000 men). The retreating path of 100 000 men is shown as a black line. As shown, crossing the Berezina River was catastrophic and only 10 000 men returned. The map that "may be the best statistical graphic ever drawn" [Tuf01, p. 40] shows several variables: Temperature at various dates (bottom line chart), the size of the army, its location and direction movement.

Another example from that time is the 1854 cholera map by John Snow (Figure 1.2), the legend of which was told in countless "Introduction to Visualization" classes. The story usually goes that in Broad Street, Soho, London, people were dying of cholera. Everyone was puzzled and helpless. But John Snow took the data-driven approach and counted and mapped the reported deaths. From the visualization, it became evident that deaths were concentrated around a water pump. He ripped off the pump's handle, the outbreak eventually subsided, and lives were saved. Except little of that is true, considering the arguments Kenneth Field recently collected [Fie20]: Already doctors at the time suspected that cholera is waterborne, the outbreak was likely in decline anyways, and the map was not used to identify the pattern of the outbreak, among other wrong details. Rather, while no detail of the story was uniquely his idea, John Snow did stay on top of scientific knowledge and bring together recent advances in epidemiology and thematic mapping to craft an argument supported by data and graphics for the transmission of cholera by water [Sno54].

Florence Nightingale, one of the founding figures of medical statistics, is another historic figure that must be mentioned. Born into a wealthy family, she defied societal expectations of her time and became a nurse at the age of thirty-three. In 1854, she served the British army in the Crimean War, where she witnessed the magnitude of unnecessary deaths due to bad sanitary conditions and unavailable medical supplies. Florence Nightingale, together with like-minded supporters, sought to end the suffering. To achieve this, they had to persuade the army's higher-ups, who thought that the excessive death rate are related to physical stress, bad food, or the weather, and thus unavoidable [And22]. Florence Nightingale did so by first of all collecting data and secondly by developing appealing graphics based on that data. This procedure was uncommon at the time, where data was presented mostly as tables. The graphics showed in a very convincing fashion, among other facts, the extreme disparity of the death rate in the peacetime troops vs. the common population, and the proportion of preventable deaths (Figure 1.3). Presented with hard-to-deny facts, reform became inevitable for parliament. The subsequently enacted health codes (and improved sanitary conditions before that, championed by Nightingale) drastically improved the situation. For example, only 5 % of treated soldiers died in May 1855 compared to 42 % in February [Fra02]. Nightingale's work was so influential that she became the first woman elected into the Statistical Society of England in 1858.

The field of visualization really picked up in modern times, i.e., starting in the 1960s. Jacques Bertin wrote his "Sémiologie graphique" in 1965, it was published in 1967 and revised in 1973, then translated to English in 1983 [Ber83]. As one of the milestone books

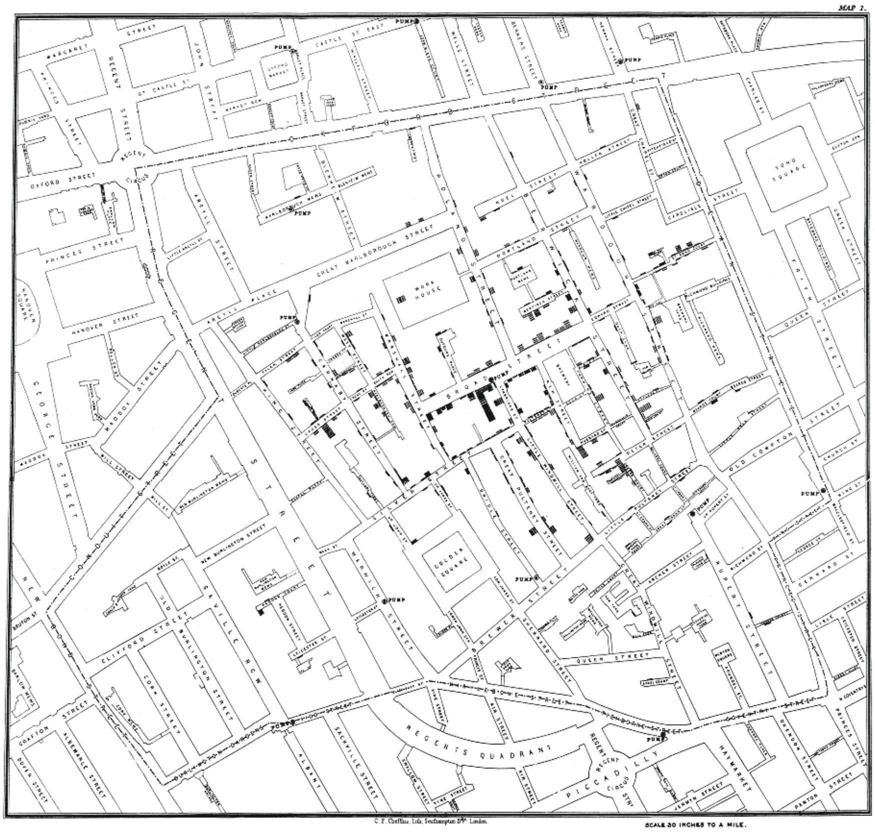
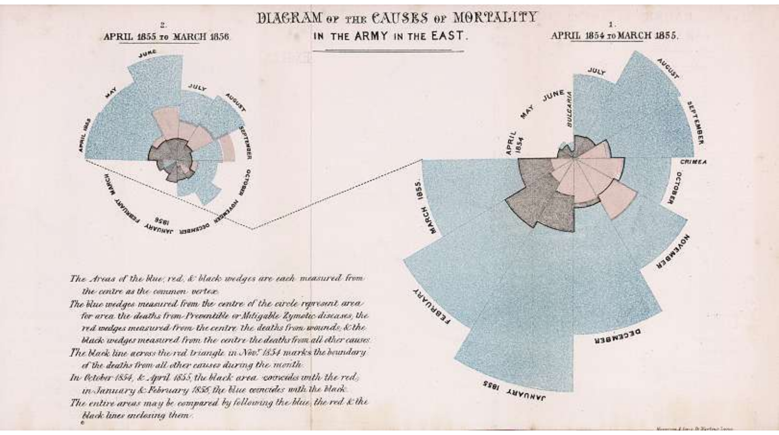


Figure 1.2: John Snow's cholera map. Image: [Sno54, Map 1]

on data visualization, it systematically discusses the role and applicability of different *visual variables*, such as position, shape, color, or length, depending on the data that needs presenting. Another milestone book was published in 1977: "Exploratory Data Analysis" by John W. Tukey [Tuk77], in which the author argues that visual inspection of the data is just as important as statistically supported hypothesis testing. We can only confirm what we suspect, and the only way to obtain such suspicions is by unconstrained exploratory analysis. He further highlights that the strength of visualization is in its potential to emphasize unanticipated features of the data or problems of their quality. Edward R. Tufte initially self-published "The Visual Display of Quantitative Information" in 1983 [Tuf01]. The book condenses the practice of creating data graphics into a couple of guidelines. Established concepts like "chartjunk" and "graphical excellence" originate here. Although Tufte subscribes to a specific minimalist aesthetic of graphic design that is not scientifically grounded in visual perception, his work became highly influential. The 1980s was also the decade of the personal computer, where workstations became smaller and powerful enough to carry out everyday tasks in little time. Researchers quickly realized the potential of *interactive* graphics. Basic geometric interactions, such as rotation or zoom, were established, but also the idea of animating data graphics, or



The Areas of the blue, red, & black wedges are each measured from the centre as the common vertex.
The blue wedges measured from the centre of the circle represent area for area the deaths from Preventible or Mitigable Zymotic diseases, the red wedges measured from the centre the deaths from wounds, & the black wedges measured from the centre the deaths from all other causes.
The black line across the red triangle in Nov.r 1854 marks the boundary of the deaths from all other causes during the month.
In October 1854, & April 1855, the black area coincides with the red, in January & February 1856, the blue coincides with the black.
The entire areas may be compared by following the blue, the red & the black lines enclosing them.

Figure 1.3: Polar area chart showing cause of death in the British army. Blue wedges denote preventable diseases, red wedges depict deaths from wounds, while black wedges show other causes. Image: [Nig59, p. 19]

highlighting data points with *brushing* [CM88]. By 1987, visualization was recognized as a research field within computer science [MDB87], and the first IEEE Visualization conference (today: IEEE VIS) took place in 1990. Leland Wilkinson published "The Grammar of Graphics" in 1999, in which he advocates to abolish the notion of a chart as a rigid fully pre-specified type of graphic. Instead, he highlights the rules that govern statistical graphs (i.e., the grammar), with which we can flexibly define various graphics. His ideas became central to statistical plotting in R [R C23], where they were implemented in the `ggplot` package [Wic16].

Also in 1999, Card, Mackinlay, and Shneiderman wrote that "the foundational period of information visualization is now ending" [CMS99, p. xiii] and the time of intro- and retrospection began. Let us first discuss what visualization *is*. Card et al. [CMS99, pp. 6–7] define visualization as "the use of computer-supported, interactive, visual representations of data to amplify cognition." Colin Ware mentions a somewhat loose definition of "a graphical representation of data or concepts" [War04, p. 2]. Tamara Munzner [Mun14, p. 1] later used "computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively" as a definition. Visualizations are seen as external tools to aid thought and reasoning in a specific task, similar to pen and paper for multiplication or a compass and a map for navigation. Because "the power of the unaided mind is highly overrated" [Nor94, p. 43], visualization brings several advantages compared to carrying out the same task without it [War04, pp. 3–4]. For one, "visualization provides [the] ability to comprehend huge amounts of data." It also "allows the perception of emergent properties that were not anticipated." Consequently, "visualization supports hypothesis generation," which is exactly Tukey's argument for exploratory visual analysis [Tuk77]. Finally, "visualization often enables problems with the data itself to become immediately apparent," e.g., in the form of missing or erroneous data. In the words of John Tukey [Tuk77, p. vi], "the greatest value of a picture is when it forces us to notice what we never expected to see." Great examples that bring these points home are Anscombe's quartet or Datasaurus dozen (Figure 1.4): Scatterplots of two variables with identical summary statistics, but wildly different visual appearances.

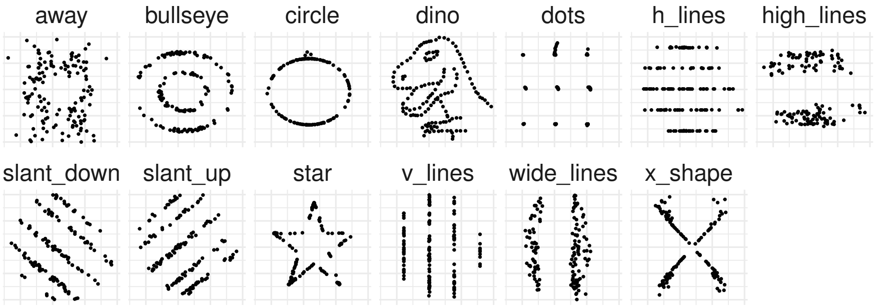


Figure 1.4: Datasaurus' dozen [MF17]. All scatterplots have the same mean, variance and Pearson's correlation. Recreated using the `quartets` R package [DAg23].

1.1.2 Visualization Pipelines, Processes, and Models

Visualizations can be thought of "adjustable mappings from data to visual form to the human perceiver" [CMS99, p. 17]. A widely accepted model for a visualization pipeline, i.e., a series of such mappings, is the *InfoVis pipeline* (Figure 1.5). First, *raw* data has to be translated into neat *data tables*, i.e., relational descriptions of attributes we care about. Those tables are then mapped to *visual structures* by encoding attribute values in graphical properties (e.g., position, size, or color) of visual marks (points, lines, areas, volumes). Finally, the *view*, the particular image, is created from those structures, possibly dependent on *view transformations*, like rotation, zooming, or distortion.

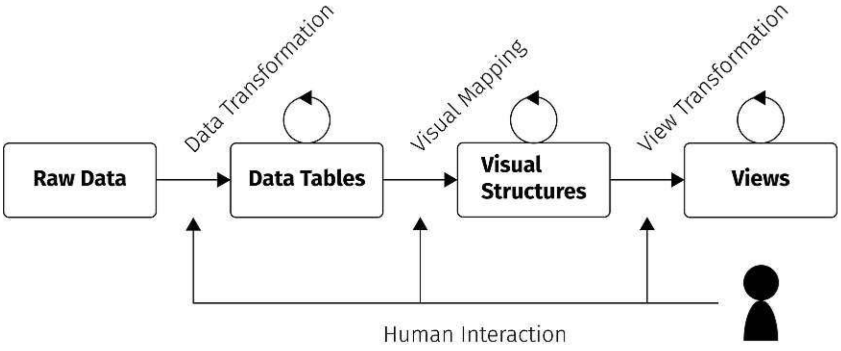


Figure 1.5: InfoVis Pipeline after Card et al. [CMS99, Fig. 1.23]

Interactivity has been recognized as a defining feature of visualizations, as evidenced by the presented definitions and the fact that the InfoVis pipeline accounts for interaction in all pipeline steps. The reason for interacitivity arises from the fact that even though an image is worth a thousand words, it can only show so much while fitting into a typical computer screen. Getting insights from current datasets often involves switching between various points of view (figuratively and literally) and between multiple levels of detail, e.g., between aggregations/summaries and individual data cases. Interactions can modify data transformations (e.g., details-on-demand, brushing, dynamic queries), visual mappings (e.g., pivot tables), and view transformations (e.g., lenses [Tom+17] or geometric navigation like pan and zoom). They are thus a crucial component in scaling visualizations to both the size of the dataset and the number and complexity of questions we have. Insights from data have to be obtained over time; a fact that van Wijk [vWij05] later recognized in his visualization model (Figure 1.6) that was used to reason about the economic (as in gain vs. effort) value of a visualization. In that model, visualizations (V) are built from data (D) and a specification (S), which are comparable to data tables and visual mappings of the InfoVis pipeline. The image (I) is perceived (P) by the user, who learns something and thus gains knowledge (K). That new knowledge leads to new hypotheses, which they test by changing the specification in an interactive exploration (E) process.

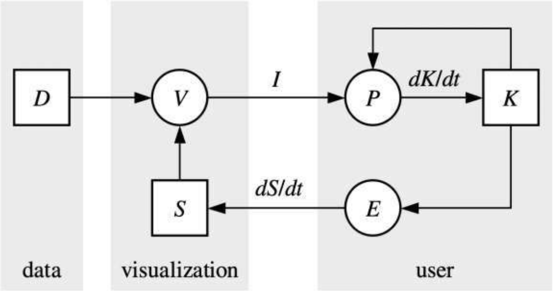


Figure 1.6: A model for visualization by van Wijk. Image: [vWij05, Fig. 1] © 2005 IEEE

As *computer-supported* is a key feature of visualization and as the amount of data to sift through grew ever larger while the power and sophistication of data mining techniques increased, *visual analytics* emerged. Thomas and Cook defined VA as “the science of analytical reasoning facilitated by interactive visual interfaces” [TC05, p. 4]. Keim et al. [Kei+08, p. 157] later specified further that visual analytics “combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data sets.” The goal of VA is to “detect the expected and discover the unexpected,” “provide timely, defensible, and understandable assessments,” “communicate assessment effectively for action,” and “synthesize information and derive insight from massive, dynamic, ambiguous, and often conflicting data” [Kei+08, p. 157]. The key difference to visualization is that VA is an “integral approach to decision-making, combining visualization, human factors and data analysis” [Kei+08, p. 158]. VA is a inter-disciplinary research field combining visualization, data mining, perception, data management, statistics, human-centered computing, and other related fields [Kei+08, Fig. 2–3]. Conceptually, VA builds on the InfoVis pipeline (Figure 1.5) and van Wijk’s model (Figure 1.6) to account for the tight integration of automated analysis methods and interactive visualizations (Figure 1.7). The *model*, a computer-internal representation of the real world built from collected data, appears as its own box. After necessary data transformations to arrive at data tables, the data may either be visualized or used in data mining to build the model. The analyst has to evaluate the model and assess its appropriatenes, then adjust it as necessary. To do so, interactive visualizations are employed to support, e.g., selection of parameters. These are the *model building* and *model visualization* arrows in Figure 1.7, which allow to discard low-quality analysis results at an early stage. Ultimately, knowledge is gained from the interplay between interactive visualizations and automated analysis techniques. Andrienko et al. [And+18] arguably went a step further when they more recently provided a goal-oriented alternative to the process-oriented VA definitions. Following the VA tasks *assess*, *forecast*, and *develop options* by Thomas and Cook [TC05], they define corresponding *descriptive*, *predictive*, or *decision supporting* models. The desired end product (goal) of a VA process is an appropriate model to support either of the three VA tasks.

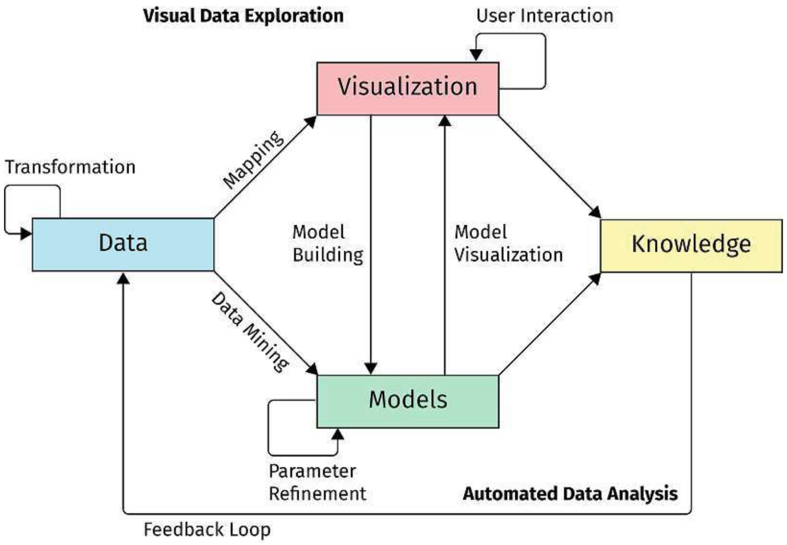


Figure 1.7: VA process after Keim et al. [Kei+10, Fig. 2.3]

As visualization (the field) aims to aid in the analysis of an often very complex dataset, naturally the question arises how such a visualization (the image) should look like. Obviously, the design process should incoroporate the intended target users, the characteristics of the dataset and what users want to know from the dataset. While the previously mentioned models describe the intended high-level VA process/goal and some [And+18] are even a bit instructive how to achieve that, in a specific problem instance the visualization designer remains with many questions. They have to figure out what entities, attributes and relationships in the dataset are important to the intended users. As visualizations are computer-generated and computers need very specific instructions, a precise visual mapping from data to image has to be defined. In addition, data mining algorithms need to be found, assessed, and parameterized. Because visualization designers are, most of the time, not the visualization's target users, the two groups have to collaborate on the design. As such, visualization design is a form of problem-driven research and fits to an interpretivist research paradigm [MD20]. The latter is intuitively understood by considering that another designer will most likely not produce the same visualization, all other parameters and context being equal. Sedlmair et al. [SMM12] formalized their learnings in this problem-driven research into the Design Study Methodology. The authors define a design study as following [SMM12, p. 2432]:

> "A design study is a project in which visualization researchers analyze a specific real-world problem faced by domain experts, design a visualization system that supports solving this problem, validate the design, and reflect about lessons learned in order to refine visualization design guidelines."

Among other contributions, the Design Study Methodology proposes a 9-stage framework to the process (Figure 1.8) and common pitfalls that threaten a successful outcome. For instance, the first three stages (precondition phase) are comprised of *learning* about related visualization literature and about selecting collaborators as well as identifying their roles. Possible pitfalls in these stages include unavailability of real datasets or that visualization is actually not required. The core phase consists of characterizing the expert's problem and formulating a task abstraction (*discover*), *designing* a visualization to solve those problems, and finally *implementing* the design and *deploying* the tool. Several pitfalls may occur in this phase as well, e.g., prototyping designs may take too much time, too early commitment to a particular design, or evaluating designs without their intended users. The final *analysis* phase incorporates *reflecting* on the work and inform future visualization guidelines by *writing* the design study paper. Pitfalls in this phase include, e.g., telling the story chronologically instead of focusing on results or ending the design study prematurely.

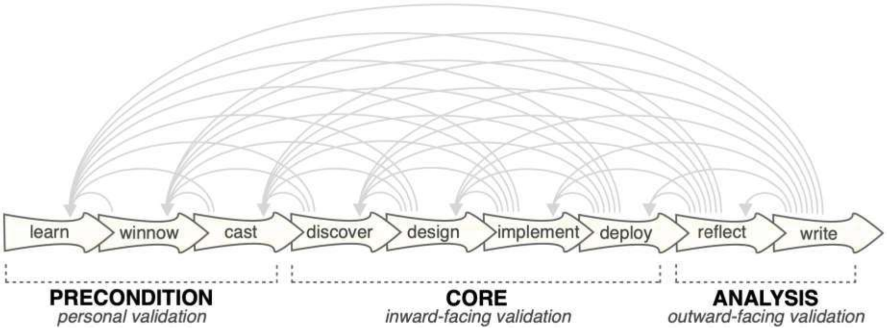


Figure 1.8: The 9-stage framework of the Design Study Methodology. Image: [SMM12, Fig. 2] © 2012 IEEE

Tamara Munzner [Mun09; Mun14] suggests four nested levels that a visualization designer has to tackle (Figure 1.9) in visualization design. The four levels are meant to be visited as necessary, so a strictly sequential order is not required. Each level also suggests different ways to evaluate the visualization.

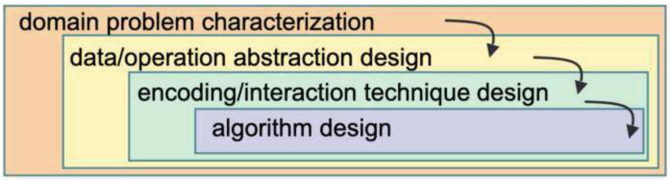

domain problem characterization
data/operation abstraction design
encoding/interaction technique design
algorithm design

Figure 1.9: Munzner's Nested Model of Visualization Design. Image: [Mun09, Fig. 1] © 2009 IEEE

L1, Domain Situation. In the first level, the designer learns about the domain experts' specific but big-picture situation: Who are the visualization's target users, what data do they handle, what do they want to find out from the dataset, how do they currently solve the problem, and so on. These questions can be answered by a variety of ethnographic research methods, but often involve discussions with the experts and observing them work in practice while interrupting for questions [SMM12]. A simple and effective overarching framework to start with is the design triangle by Miksch and Aigner [MA14]. The vertices of the triangle are the data, the users, and their tasks, while edges represent requirements to the resulting visualization (Section 1.1.2). *Expressiveness* means that the visualization show only the necessary information and nothing more. *Effectiveness* mainly relates to "good" visualization design (compare paragraph L3 below), e.g., making use of popout effects and tailoring visualizations to the user's needs. Lastly, *appropriateness* refers to cost vs. gain considerations which can be assessed with van Wijk's economic model [vWij05].

L2, Data/Task Abstraction. Next is the second level, where learnings from the first level are specified into domain-independent abstractions. Many domains work on the same data and have similar tasks, but use different vocabulary. Abstracting from the particular vocabulary suggests related problems and computational solutions, while also increasing the potential for transferability to other contexts of the eventually found visualization. After an initial problem characterization (L1), task taxonomies and typologies become highly useful in this level. These can be general taxonomies for all kinds of visualizations, such as Shneiderman's mantra [Shn96] of *overview first, then details on demand*, the

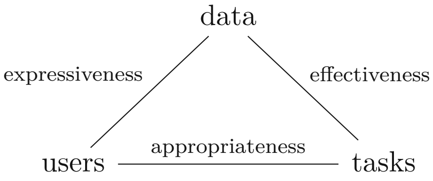


Figure 1.10: The design triangle after Miksch and Aigner. [MA14]

intent-based interaction taxonomy by Yi et al. [YKS07], the goals–tasks bridge by Lam et al. [LTM18], or Brehmer and Munzner's multi-level typology for abstract visualization tasks [BM13]. The latter is especially useful as it incorporates high-level goals (e.g., does the user want to present the data or discover findings?), intermediate-level search tasks and low-level queries while accounting for interactions with visualizations and the data itself. Other task taxonomies focus on specific data types and are helpful if one happens to work with such data. Especially relevant in our case are the task taxonomy for exploratory analysis of spatio-temporal data by Andrienko and Andrienko [AA06] or the taxonomy for interactive cartography by Roth [Rot13]. Further of interest are taxonomies that do not focus on data types but a certain class of analysis tasks. E.g., Sedlmair et al. [Sed+14] propose, among other things, six parameter space analysis tasks: *Optimization*, *sensitivity* analysis, *uncertainty* analysis, fitting, *partitioning*, and *outliers*. Nonato and Aupetit [NA19] discuss the relations between DR, data characteristics and analytic tasks, such as naming a latent dimension or clusters. Understanding user tasks is especially important as it points to their *knowledge gaps*, which in turn is necessary to design appropriate *guidance* [Cen+17].

L3, Visual Encoding and Interaction Idiom. The particular visualization images and interactions are designed in the third level. Based on the identified tasks and required information to accomplish them, the visualization designer chooses and modifies existing visualization and interaction idioms. Munzner [Mun14, p. 12] argues that designers should seek to *satisfy*, not *optimize* a visualization design, as the design space is usually too large. They should initially consider a large number of possible designs and then iteratively filter it down to the winning candidate. Hence, while visualization idioms that the target users are familiar with should be given special attention, the visualization designer must also learn and consider existing visualization/VA approaches for the data at hand, e.g., time-dependent data [Aig+23], geospatial data [He+19; Yos+20], spatio-temporal data [Bac+17], network data [FAM23], etc. Further, knowledge of human perception is critical for an effective design. E.g., it is well known that visual channels, such as position, size, or color hue, differ in accuracy and bias when encoding quantitative, ordinal, or categorical variables [CM84; Mac86; Mun14; McC+22]. Their perception usually also depends on their surroundings, as the human visual system is profoundly relative. E.g., the Ebbinghaus illusion (Figure 1.11) shows that size perception depends on the elements around the target. The human visual system also cannot separate some channels (e.g., red part and blue part of color hue), which should generally be avoided. Preattentive processing ("popout") on the other hand should be sought after to present critical information. In case of multiple visualizations, e.g., coordinated multiple views, general models of VA [Kei+08; vWij05; Sac+14; And+18] that were discussed earlier, can help to build and define the interplay between them.

L4, Algorithm Design In the last level, the designer focuses on the intricate details of the algorithm that constructs the desired visualization. First and foremost, its correctness should be verified, i.e., that it terminates and produces a correct result. This

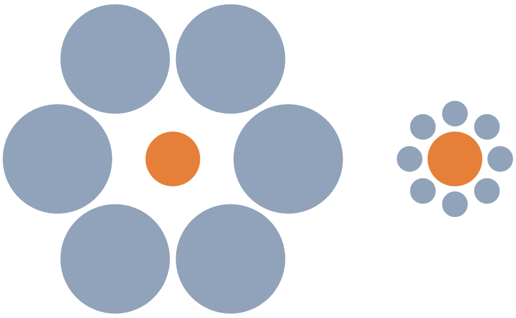


Figure 1.11: Ebbinghaus Illusion. The two center circles are the same size, but appear different due the outer circles' relative sizes.

can be achieved by formal systems (e.g., Hoare rules) or more simply by software testing methods. Next, it is important that the algorithm terminates within time frames suitable for seamless human-computer interaction. Waiting times below 100 ms are referred to as instantaneous, below 1 s as noticable but short enough to keep the user's "train of thought", and below 10 s as short enough to keep the user at their desk [CRM91, Tab. 3]. Response times of algorithms may be evaluated with benchmarks on suitable datasets [Lam+12]. If it turns out that an algorithm is unsuitable for interactive time frames, due to its complexity or the sheer amount of data it must handle, a viable solution may be Progressive VA [Ang+18], where visualizations are rendered progressively.

L1–L4, Validation. To ensure that the VA solution works as intended, validation is required at each of the four levels. The design study pitfalls by Sedlmair et al. [SMM12] can help avoid mistakes in the validation process. Munzner [Mun14, Sec. 4.5] suggests "threats to validity" at each level, specifically *wrong problem* (L1, misunderstood user's needs), *wrong abstraction* (L2, visualization shows the wrong thing), *wrong idiom* (L3, flawed way of visualizing the thing), and *wrong algorithm* (L4, code is too slow). Munzner [Mun14, Sec. 4.6] then proposes to distinguish between downstream (i.e., *after* implementation) and immediate (before implementation) validation, where every validation is supposed to counteract the threat at the respective level. To avoid too slow code (L4), one may take into account the algorithm's computational complexity before implementation (immediate) and test the implementation's efficiency by measuring time/memory consumption (downstream). The implementation's correctness can be validated formally or by software testing approaches. To avoid the wrong idiom (L3) immediately, one can only provide an educated justification that the particular design choices are likely to be useful. On the other hand, many alternatives exist in the downstream direction. The Seven Scenarios by Lam et al. [Lam+12] provide a good taxonomy and summary of validation strategies. The authors distinguish between *understanding data analysis* and *understanding visualizations*. The former is concerned with how a visualization impacts the user's workflow, i.e., whether they can do the same tasks faster or gain previously unattainable insights. More specifically, this category further groups into the following:

- *environment and work practices*, i.e., how are visualizations used today in a particular domain?
- *visual data analysis and reasoning*, i.e., does a visualization support insight and hypothesis generation?
- *communication through visualization*, i.e., how can visualization support communication in the wider sense, such as teaching or presenting?
- *collaborative data analysis*, i.e., does a visualization system support collaborative analysis where multiple people arrive at a joint conclusion?

As such, common validation strategies for *understanding data analysis* are qualitative research methods: Observing users using the tool, usability questionnaires using Likert scales [Sou+22], heuristic expert evaluations [Wal+19], or unstructured interviews. The latter (*understanding visualizations*) cares about the visualization itself, i.e., if it is quicker to produce or better at specific tasks than alternatives. To validate that stage, quality metrics (if available) may be used on the generated images or controlled laboratory experiments may be conducted that measure participants' completion time and error for specified tasks, in addition to the strategies outlined for L4 previously. To ensure that the abstraction is right (L2), one may only do that in the downstream direction. Validation approaches for *understanding data analysis* may be used here. Similarly, according to Munzner [Mun14, p. 77–78], only downstream validation is available to validate that the right problem was solved (L1), e.g., by collecting adoption rates after deployment or field studies.

1.1.3 Blind Source Separation

BSS is a statistical modeling framework that originated in the signal processing community in the 1980s. A summary of its history can be found, e.g., in [CJ10; YHX14]. BSS's goal is to separate source components from a mixed signal, but, as the "blind" in the name indicates, the source components themselves are unobserved. (This thesis refers to them as latent "sources", "components," or "dimensions.") The classic example to motivate BSS is the "cocktail party problem." At a party, where the room is filled with a mix of all simultaneously happening conversations, it is easy for humans to focus their attention on individual speakers and separate their voices from others. This is, essentially, what BSS aims to achieve. Prior knowledge has to be assumed about the properties of the source signals, the mixing process, or how noise is integrated into the models to make the problem tractable. The general data flow of BSS (for the purpose of this thesis) is illustrated in Figure 1.12.

For this thesis, we consider instantaneous linear mixture models for multivariate data. The goal of this section is to provide the reader some intuition about BSS tuning parameters.

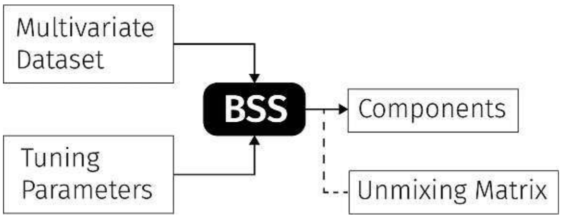


Figure 1.12: A data flow model of BSS. BSS methods considered in this thesis take tuning parameters and the multivariate dataset as input, and provide latent components as output by way of the unmixing matrix.

This section is not a detailed discussion of BSS and all its models, or how they relate to other statistical modeling techniques, which is far better explained elsewhere, e.g., [CJ10; YHX14; NO18; Pan+21; Mue21] and references therein. We will refer to a BSS *model* as specific set of assumptions imposed on latent components.

It turns out that the same problem statement has applications in various domains. Various BSS models were used in speech signal separation, communications, image processing, earth sciences, or biomedical analysis [YHX14]. There are two major purposes for obtained latent components. They may be inspected visually to learn about the measured phenomenon. For instance, Liu et al. [Liu+19] propose a BSS model to de-noise radar measurements of bridges, which can determine their structural integrity. On the other hand, latent components may be the input for other algorithms. In the context of multivariate spatial prediction, Muehlmann et al. [MNY20] showed that combining univariate models of latent components is comparable to and often better than (usually complex) multivariate modeling techniques. Noisy or non-informative sources may be removed in such an application to obtain a dataset with lower dimensionality.

More formally, the basic BSS model is a location-scatter model stating that

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{\mu} \tag{1.1}$$

where $\boldsymbol{x}$ is the p-variate mixed signal, $\boldsymbol{z}$ is the p unknown standardized source signal, $\boldsymbol{\mu}$ is a location vector and $\boldsymbol{A}$ is the $p \times p$ mixing matrix. The goal is to find an unmixing matrix $\boldsymbol{W}$ that can recover $\boldsymbol{z}$, i.e., $\boldsymbol{z} = \boldsymbol{W}(\boldsymbol{x} - \boldsymbol{\mu})$, up to sign and order of $\boldsymbol{z}$. In theory and practice, this procedure often involves the (joint) diagonalization of various so-called scatter matrices [TI06; NR22].

Common in all BSS models is that the statistical properties (like independence or stationarity) of latent components are known due to the framework's model-based approach. Knowing them can help to decide what is appropriate to do subsequently with the obtained sources. For state-of-the-art DR techniques used in visualization, like UMAP or t-SNE, this is not the case, although also arguably not that relevant. Another advantage of BSS models handling temporal or spatial data is that they properly and explicitly model such dependence (cf. Tobler's law [Tob70]) in the assumptions about latent components. This property makes such models preferable over alternatives that do not account for them. Because models discussed in this thesis consider linear mixtures and BSS retains the loading-scores scheme from PCA, latent components are easily interpretable in the sense that individual scores of a source directly relate to input variables.

The downsides or potential drawbacks of BSS models are what motivates this thesis. The recently developed models for temporal and spatial multivariate data take complex tuning parameters that will be presented in the following subsections. Currently, no goodness-of-fit measures or tests exist for BSS that could decide if one parametrization is preferable over another. Consequently, analysts must compare latent components obtained by various parametrizations, which is demanding for several reasons. First of all, due to the tuning parameters' complexity and importance, the analyst has to try out several settings. For p-variate temporally or spatially distributed data, each new BSS method yields p new latent components to consider. They can add up quickly especially when the application domain deals with several dozen variables, as it is common, e.g., in geochemistry. latent components obtained with BSS usually have properties, by model assumptions, that further complicate visualization and data mining. They are defined only up to sign and order, which means that any source can have its sign reversed and still be a valid solution. Since sources are not ordered, functions that derive a number by which sources can be ordered are required. These functions must not use the first two moments, which are equal for all sources.

As mentioned earlier, one arrives at different BSS models depending on various modeling assumptions. When $\boldsymbol{z}$ are assumed to be independent, identically distributed (iid), the model is Independent Component Analysis (ICA): Samples of each source have to be identically distributed, and all pairs of sources have to be independent of each other [CJ10, p. 12], which allows for only one Gaussian source whereas the rest has to be non-Gaussian. ICA is often compared to the well-known PCA. While PCA finds components (sources) that are orthogonal and in the directions of highest variance, ICA obtains independent and not necessarily orthogonal components using higher-order statistics, such as kurtosis.

When samples are ordered in time, it is no longer realistic to assume that subsequent samples are independent of each other, and one arrives at TBSS models (Section 1.1.3). Following Tobler's law [Tob70], a similar argument can be made for spatially distributed data (SBSS, Section 1.1.3), where nearby collected samples are expected to be more alike than two samples collected far away from each other. Models for these two cases will be discussed in the following sections together with brief application examples. A common theme in temporal/spatial BSS is that the particular methods diagonalize multiple covariance matrices, e.g., the sample covariance matrix and the autocovariance matrix at a given lag in the case of Algorithm for Multiple Unknown Signal Extraction (AMUSE). Or in the spatial case, the sample covariance matrix and a so-called local covariance matrix. There is a parallel between all these covariance matrices in that, if we take i, j as one-dimensional indices for time steps or two-dimensional indices for spatial locations, the matrices in question are computed as $\sum_{i,j} f(i,j)\boldsymbol{x}(i)\boldsymbol{x}(j)'$, where $f(i,j)$ is a function providing weights based on the distance between i and j. E.g., the sample covariance is obtained with $f(i,j) = I(i = j)$, the (temporal) autocovariance at lag τ with $f(i,j) = I(|i - j| = \tau)$ and the (spatial) local covariance within a ring (r_1, r_2) with $f(i,j) = I(r_1 \leq |i - j| \leq r_2)$. $I(\cdot)$ denotes the indicator function, i.e., it resolves to 1 if its argument is true and 0 otherwise.

Temporal Blind Source Separation

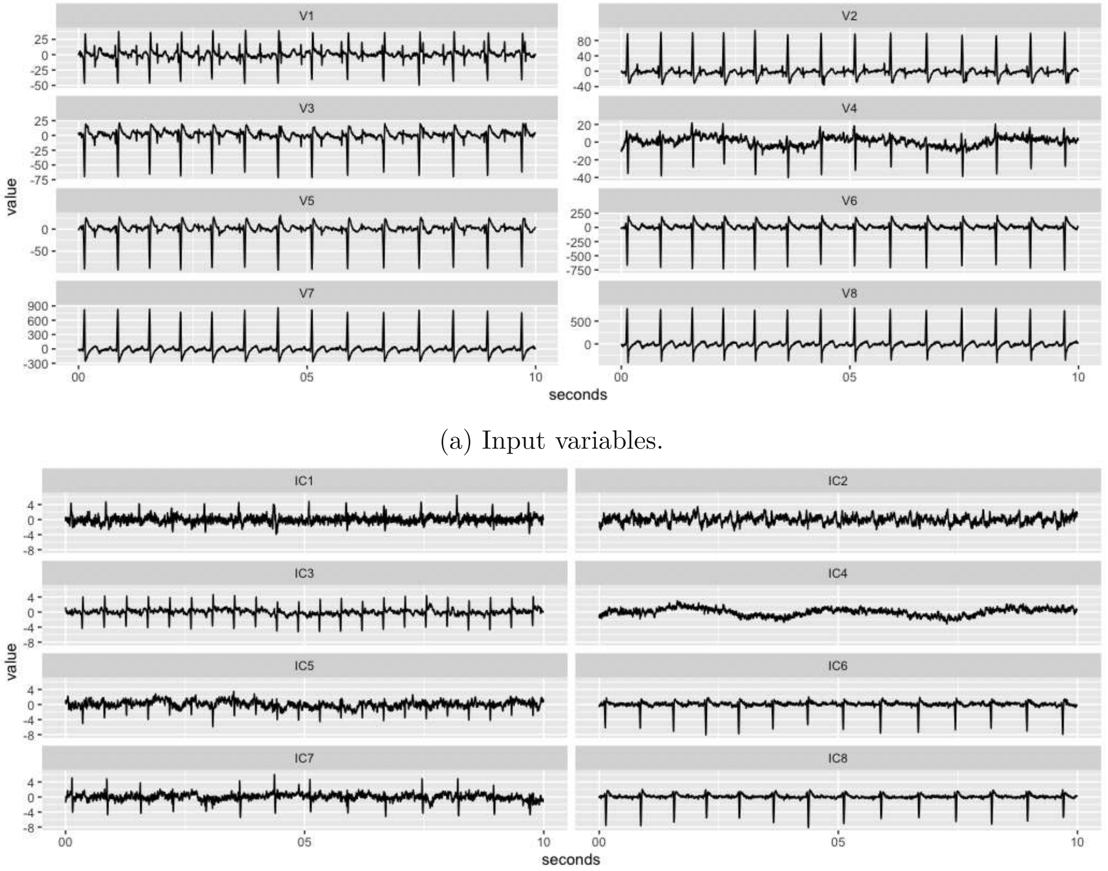
TBSS is used for multivariate time series, i.e, vectors associated with points in time. Applications include, e.g., finance [OKM00], environmental sciences [NFF21], or healthcare [dLdMV00]. An example for the latter is depicted in Figure 1.13. Figure 1.13a shows the raw readings from eight electrocardiogram (ECG) electrodes to measure the heart's electrical activity. Notably, the patient is a pregnant woman, so the ECG data include signals of two hearts. TBSS may be used to filter out the fetus' heartbeat (IC3 in Figure 1.13b), which is a non-invasive way to check if it is healthy, e.g., that the heart does not miss any beats or behaves irregularly in other ways.

Starting from the base BSS model, we add time steps t to signify the temporal order of observations:

$$\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{z}(t) + \boldsymbol{\mu}, t \in \{1, \ldots, T\} \tag{1.2}$$

Just as in time series modeling, one has to make assumptions about the sources' characteristics to make any sort of inference. One fundamental concept is that of *stationarity*. Informally, stationarity means that the behavior of the time series does not change depending on when we start to observe it — it will look roughly the same at intervals of the same length [PW10, p. 6]. A multivariate time series $\boldsymbol{x}(t)$ is said to be weakly or second-order stationary if its first and second moments exist and are constant. When that is the case, the covariance between its variables (cross-covariance) and the covariance with itself (autocovariance) depends only on the temporal distance $|t_1 - t_2|$ (lag) between two time steps.

A popular sub-model of BSS is the second-order separation (SOS) model, where $\boldsymbol{z}(t)$ are assumed to be second-order stationary. Sources can be recovered by assuming distinct serial dependence structures, i.e, $\text{Cov}(\boldsymbol{z}_t, \boldsymbol{z}_{t-\tau})$ is a diagonal matrix for all $t \in \mathbb{Z}$ and $\tau = \pm 1, \pm 2, \ldots$. The first proposed TBSS method of this sort was the AMUSE [Ton+90], which diagonalizes the covariance matrix and an autocovariance matrix given at a lag τ to estimate the unmixing matrix $\hat{W}$. Thus, τ is a tuning parameter that the analyst has to select and the separation performance greatly depends on that choice. Consequently, AMUSE was later generalized to Second-Order Blind Identification (SOBI) [Bel+97], which jointly diagonalizes several autocovariance matrices. By incorporating information at multiple lags $\mathbb{T} = \{\tau_1, \ldots, \tau_k\}$, the individual lag choices become less critical, but the space of possible tuning parameter selections increases from T to 2^T, where T is the length of $\boldsymbol{x}(t)$. BSS has applications in various fields and it is expected that the



(a) Input variables.

(b) Latent components.

Figure 1.13: TBSS example application in healthcare. a) Ten seconds of ECG data by a pregnant woman. The readings include both the mother's and the fetus' heartbeat. The former is much stronger and masks the latter. b) Eight components identified in the ECG. The fetal heart beats faster than that of an adult as visible in IC3.

time series' key characteristics vary between fields. Put bluntly: An ECG reading looks different than the prices at which Apple shares are traded. Especially financial time series, like stock prices, often exhibit sudden spikes and thus are not well characterized by second-order moments, like autocovariance matrices. Most information is in higher-order moments and that property is called "stochastic volatility". This fact was taken into account by Matilainen et al. [Mat+17], who proposed a family of estimators called variant of Second-Order Blind Identification (vSOBI). As a variant of SOBI that diagonalizes scatter matrices of lagged fourth moments, vSOBI shares the tuning parameter space with SOBI, which is again a set of lags $\mathbb{T}$. As SOBI cannot separate sources exhibiting stochastic volatility and vSOBI does not perform well on sources without that property [Mat+17], Miettinen et al. [Mie+20] developed a weighted version between the two, called gSOBI. The weight $b \in [0, 1]$ specifices the weight of the linear part, so gSOBI reduces to SOBI when $b = 1$ and to vSOBI when $b = 0$. Consequently, tuning parameters in addition to the weight b are lag sets for the two models, i.e., $\mathbb{T}_1$ for SOBI and $\mathbb{T}_2$ for vSOBI.

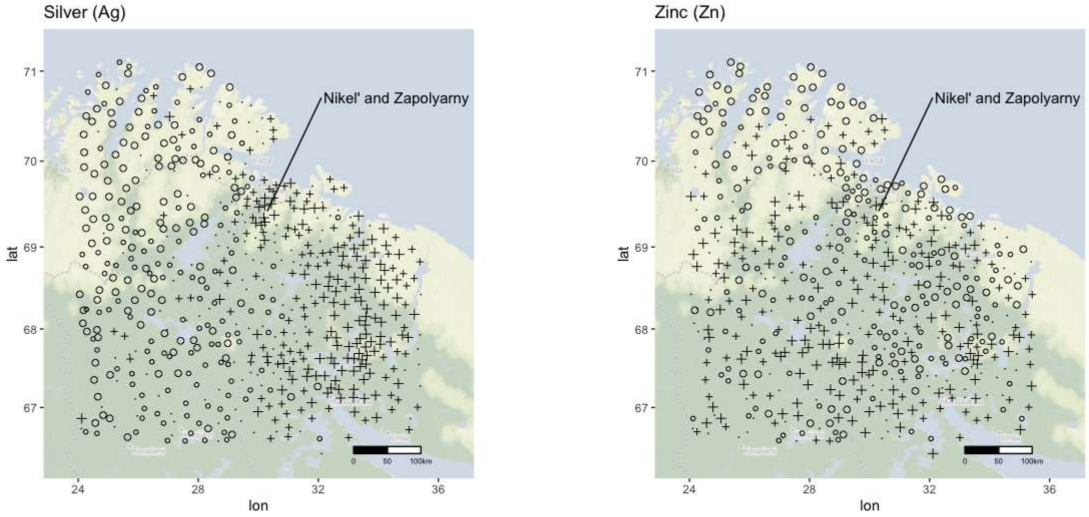
If time series are allowed to have time-shifting first or second moments, one finds themselves in a non-stationary environment. Such models may be, e.g., block-stationary models that assume the temporal dimension can be subdivided into contiguous non-overlapping blocks in which the second-order stationarity assumption again holds. The definition of such blocks then becomes the burden of the analyst, which further enlarges the available tuning parameter space. Such TBSS models are not considered in this thesis, but they are a direct equivalent to non-stationary SBSS models discussed in the next section.

A common theme in the literature of TBSS models is that the particular choice of tuning parameters is crucial for separation performance [TLS05; TMN16; Mie+20; Pan+21]. For gSOBI, Miettinen et al. [Mie+20] propose a default of $b = 0.9$, as the vSOBI part dominates equal weightings, but give no recommendations regarding the lag sets for SOBI and vSOBI. Tang et al. [TLS05, p. 508] investigated SOBI parameter settings for electroencephalogram (EEG) data and discovered that “the ability of SOBI to isolate signals associated with neuronal activations from specific brain regions depends on the appropriate selection of [lags].” While the authors propose a “standard set” of lags, it is unclear how (if) those translate to other fields. TBSS theory does provide guidelines, for instance, that lags are preferred which maximize the difference of diagonal entries in corresponding (cross-)moment matrices of $\boldsymbol{z}$. Informally, it means to choose lags such that the temporal behavior of latent components at the chosen lags is as distinct as possible. However, such guidelines that relate to unobserved sources are unfortunately of limited practical use.

Spatial Blind Source Separation

SBSS deals with the analysis of multivariate spatial data, i.e., vectors associated with locations in space. An example are geochemical surveys, like the Kola project [Rei+98]. Among other samples, moss was collected at ca. 600 spatial locations and the amount of 31 chemical elements in it measured. Figure 1.14 shows plots of the (alphabetically) first and last element in the dataset, which is publicly available as part of the `StatDA` R package [Fil23]. On the other hand, Figure 1.15 shows the first two components obtained with SBSS [Nor+15] and a 50 km ball kernel parameter. Low values (circles) in SBSS component 1 clearly concentrate around Nikel’ and Zapolyarny in Russia and show the nickel-processing industry located there. SBSS component 2, on the other hand, shows an east–west pattern that “separates industrial contamination from the background on a large regional scale.” [Nor+15, p. 766]

Starting again from the base BSS model, we add locations $\boldsymbol{s} \in S \subseteq \mathbb{R}^d$ to indicate position in a d-dimensional Euclidean space:



(a) Silver (Ag).

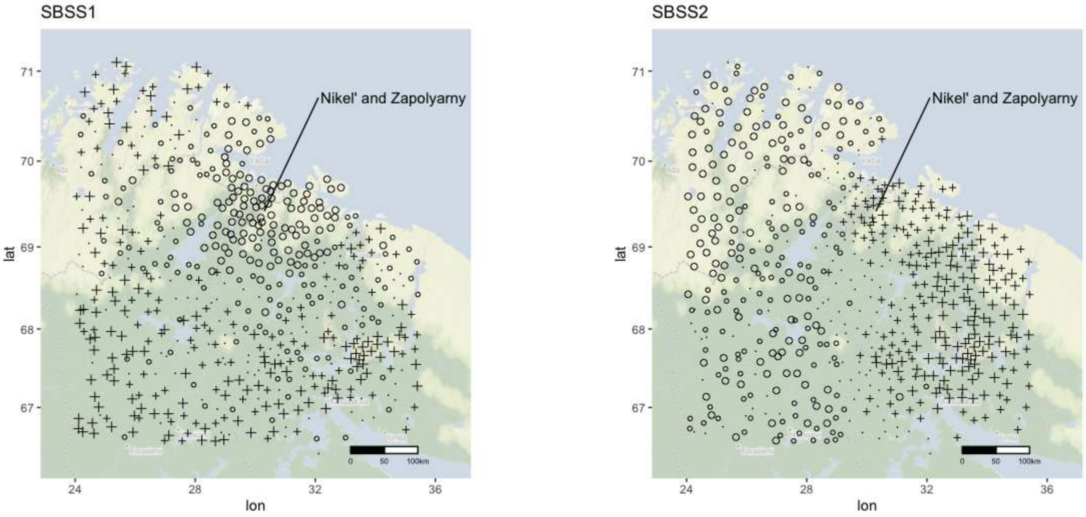
(b) Zinc (Zn).

Figure 1.14: Plots of two out of 31 elements in collected moss samples as part of the Kola project [Rei+98]. Values are binned into 20 % bins (quintiles) encoded by big circles (first 20 %), small circles (second), dots (third), small crosses (fourth) and big crosses (fifth) following geochemical mapping practices [Rei+08].

$$\boldsymbol{x}(\boldsymbol{s}) = \boldsymbol{A}\boldsymbol{z}(\boldsymbol{s}) + \boldsymbol{\mu}, \boldsymbol{s} \in S \tag{1.3}$$

As the model considers spatially distributed vectors, i.e., multiple variables associated with locations, we find ourselves in the realm of (multivariate) geostatistics. A core notion in geostatistics is that of a *random field*. A p-variate random field $\boldsymbol{x}(\boldsymbol{s})$ is a family of random p-variate vectors indexed by the spatial domain $S \subseteq \mathbb{R}^d$. Notably, a random field exists everywhere in S, but we only ever see one realization of it at some sample locations $C \subset S$. One common task of geostatistics is prediction: What measurements can we expect in locations where we did not observe the random field? To model a random field exactly in order to answer that question, all its moments have to be described, which is usually infeasible. A common simplification is then to only consider the first two moments, given by the mean function and the spatial covariance function. The latter is written as $\mathrm{Cov}(\boldsymbol{x}(\boldsymbol{s}_1), \boldsymbol{x}(\boldsymbol{s}_2))$, but, as modeling the random field in terms of all location pairs is also usually infeasible, further simplifying assumptions are that the covariance function depends only on the distance between locations (stationarity) and that it is not dependent on the angle between locations (isotropy). More detailed discussions of spatial covariance functions can be found in [Wac03; GK15].

Given these two assumptions of stationarity and isotropy, the corresponding BSS model is that proposed by Nordhausen et al. [Nor+15]. Its asymptotic behavior was later investigated by Bachoc et al. [Bac+20]. The model can be seen similar in spirit to



(a) SBSS Component 1.

(b) SBSS Component 2.

Figure 1.15: First two components identified in the Kola dataset with SBSS. Visual encoding identical to Figure 1.14.

AMUSE from the TBSS context as it diagonalizes the sample covariance matrix and a so-called local covariance matrix **LCov**. This matrix is computed with the help of kernel functions providing weights for each pair of locations. Bachoc et al. [Bac+20] propose three such spatial kernel functions: A ball (circle) f_{ball} parametrized by its radius r, a ring f_{ring} parametrized by inner and outer radius r_{in}, r_{out}, and a Gauss kernel f_{gauss} which is a "smooth" version of f_{ball}. These are the *kernel* tuning parameters of SBSS. Similar to the temporal lag in AMUSE, the kernel choice and parametrization matters greatly for separation performance. Akin to the improvement by SOBI over AMUSE in the time series context, Bachoc et al. [Bac+20] formulate a stationary SBSS model where multiple local covariance matrices are involved. The tuning parameter space's size then multiplies accordingly, as several kernels can be chosen and parametrized.

The assumption of stationarity might not be realistic for the dataset at hand, especially so when the sample locations are distributed over a wide area. This can be the case, e.g., for geochemical surveys, such as GEMAS [Rei+14]. GEMAS combines chemical soil measurements taken at roughly 2 100 locations across Europe. As such, it includes regions that differ considerably regarding climate, rock formations, soil type, population, or land use. To account for this, Muehlmann et al. [MBN22] adapted the block-stationary model from TBSS to the spatial context. The *blocks* are called "regions" and we get the *regionalization* tuning parameter, i.e., a partition of locations. The stationarity condition is assumed to hold within each region. A local covariance matrix is computed for each region, and they are jointly diagonalized [MBN22, Def. 4]. Again the tuning parameter space's size multiplies by all possible partitions.

Also in the SBSS literature the importance of tuning parameter settings were recognized.

Bachoc et al. [Bac+20] conducted simulation studies and found, somewhat expectedly, that incorporating more than one kernel yields more stable sources, whereas the BSS outcome is more sensitive to the particular choice of a single kernel parameter setting. The simulation studies conducted by Muehlmann et al. [MBN22] suggested that the regionalization tuning parameter has more influence on separation performance than the kernel. Similar as in TBSS, SBSS theory would prefer kernels that distinguish the spatial dependence of latent components as best as possible. Fuzzy guidelines, such as “not too small and not too large” kernels, relate to the fact that too small kernels capture too little spatial dependence structures, while a too large kernel at some point does not account for spatial dependence at all anymore. Even considering all that advice, selecting appropriate SBSS tuning parameters remains a challenging task for analysts.

1.2 Research Questions

BSS is well suited to be tackled with VA approaches as the current process resembles the VA model quite well already (Figure 1.16) as it combines visual data exploration and automated analysis (with BSS). The considered data are multivariate time series or spatial fields (blue box in Figure 1.16). Analysts employ BSS to decompose them into latent components (green box). Latent components need to be visually inspected (red box). By doing so, analysts learn about the components, the data they represent, or the model’s tuning parameters (yellow box). In addition, BSS analysis is highly exploratory [BH19], a kind of analysis that (interactive) visualizations are traditionally expected to support greatly [Tuk77]. The exploration concerns, e.g., the tuning parameters, as their appropriateness to the dataset at hand and their relation to resulting latent components are a-priori unknown. The analyst also does not know what components to expect or how they comprise the input dataset. Features in components have to be identified. Only when a hypothesis as to what they represent is available can the analyst switch to confirmatory analysis. For example, they might visually test if a spatial cluster of high scores in a certain region coincides with a type of sediment by overlaying two maps.

Following these considerations, the main research question handled in this thesis is: **How can we use VA best practices to aid usage of BSS techniques in time and space?** For practical purposes we divide it in the following subquestions:

RQ1 What are the characteristics of effective guidance for temporal and spatial BSS parameter selection?

RQ2 Which VA methods can be utilized to explore ensembles of temporal and spatial BSS components?

RQ3 How can we characterize tasks BSS analysts carry out, especially to explain latent temporal and spatial components?

RQ4 Can we adapt approaches suggested to explain multivariate DR to temporal and spatial latent components?

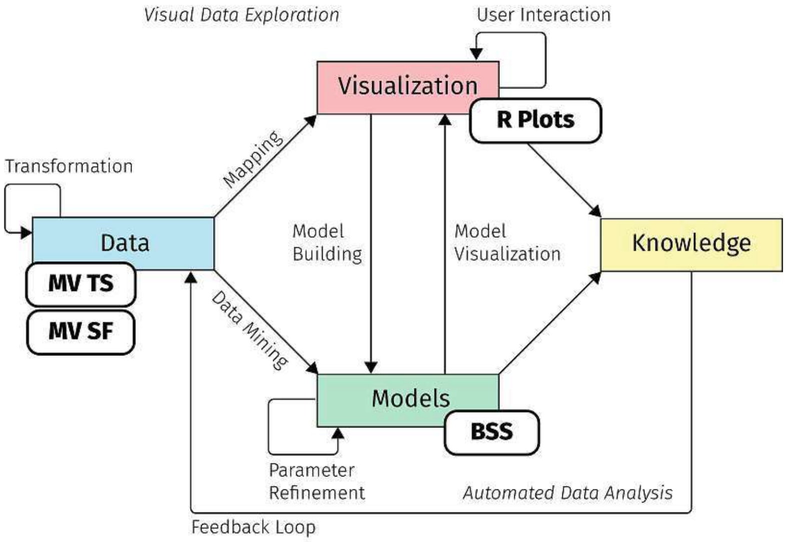


Figure 1.16: BSS in the VA model by Keim et al. (compare Figure 1.7). Multivariate time series or spatial fields (blue box) are decomposed into latent components with BSS (green). These components are plotted with static graphics (red) and visually inspected to gain knowledge (yellow).

Our reasoning for the subquestions is the following.

RQ1. Tuning parameters are crucial for the performance of BSS, but few practical guidelines exist. Analysts need guidance in the tuning parameter selection process, but their intricate tuning parameters or outputs (sets of components) are not currently considered in visualization literature.

RQ2. After several tuning parameter settings have been selected, analysts eventually must compare and decide between latent components. However, they are many and come in sets; cherry-picking components from several BSS methods or parametrizations is not an option. Suitable VA approaches have to be identified and/or developed to support set relation tasks.

RQ3. BSS task characterizations are missing in the visualization literature. Such task descriptions can be helpful for other visualization researchers to identify suitable VA solutions. Investigating especially the process of explaining patterns in latent components could be enlightening for (e.g., spatial) visual data analysis. Relating identified tasks to existing taxonomies might strengthen the latter.

RQ4. In multivariate (non-temporal and non-spatial) DR literature, several ways are proposed to explain a facet of the reduction process. E.g., which area of the DR scatterplot represents which original variable(s) [Sil+15; Soh+22] or where boundaries between categorical variables are located [Esp+23]. Furthermore, a subset of latent components can represent the original data only imperfectly, so visualizing the projection quality also received attention [Sta+16; Jeo+22]. These approaches, however, do not consider temporal or spatial data and they have to be adapted to BSS. They also usually do not indicate, e.g., whether the error is acceptable or how to improve it, which would be helpful in a practical BSS setting.

1.3 Contributions

Our work has been published in peer-reviewed scientific journals, such as Computer Graphics Forum or IEEE Transactions on Visualization and Computer Graphics, and our results have been presented at scientific conferences, like IEEE VIS or EuroVis. The scope of these journals and conferences is visualization and VA. Each of our contributions presents the VA approach, an evaluation thereof including discussion of the results, as well as an outline for future work.

The overarching contributions of our work to the field of VA are:

- Investigating how a latent variable model (BSS) can support the analysis of multivariate temporal and spatial datasets;
- Identifying challenges and gaps analysts encounter when applying BSS to a given dataset;
- Developing and evaluating VA approaches to solve these gaps and challenges, particularly regarding BSS parameter-output relationships (compare Figure 1.17), as well as providing a contextualization to the visualization literature, for:
  - Parameter optimization and result exploration for TBSS (P1);
  - Parameter optimization for SBSS (P2);
  - Sensitivity analysis for SBSS parameters (P3);
- Utilizing dissimilarity information in existing and novel ways to support analytic tasks in TBSS and SBSS;
- Summarizing results and suggesting directions for future research.

This cumulative thesis rests on the following three peer-reviewed publications, which are described in more detail in their respective chapter (Chapters 3–5). They are listed in the following, along with our contributions in the form of a CRediT statement [AOK19].

P1 **Nikolaus Piccolotto**, Markus Bögl, Theresia Gschwandtner, Christoph Muehlmann, Klaus Nordhausen, Peter Filzmoser, Silvia Miksch: *TBSSvis: Visual Analytics for Temporal Blind Source Separation*. Visual Informatics, vol. 6, no. 4, 2022. DOI: 10.1016/j.visinf.2022.10.002.

- Contributed with: Conceptualization, Methodology, Software, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization.

P2 **Nikolaus Piccolotto**, Markus Bögl, Christoph Muehlmann, Klaus Nordhausen, Peter Filzmoser, Silvia Miksch: *Visual Parameter Selection for Spatial Blind Source Separation*. Computer Graphics Forum, vol. 41, no. 3, 2022. DOI: 10.1111/cgf14530.

- Contributed with: Conceptualization, Methodology, Software, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization.

P3 **Nikolaus Piccolotto**, Markus Bögl, Christoph Muehlmann, Klaus Nordhausen, Peter Filzmoser, Johanna Schmidt, Silvia Miksch: *Data Type Agnostic Visual Sensitivity Analysis*. IEEE Transactions on Visualization and Computer Graphics, vol. 30, no. 1, 2024. DOI: 10.1109/TVCG.2023.3327203.

- Contributed with: Conceptualization, Methodology, Software, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization.

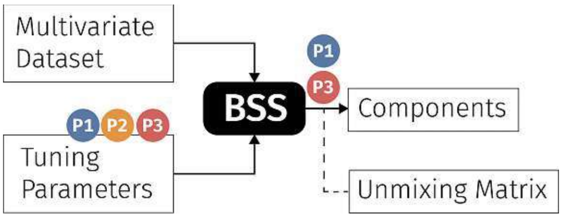


Figure 1.17: Relation of our publications to the BSS data flow model in Figure 1.12.

In addition, we (co-)authored the following papers where we leveraged our particular BSS-centered problem context to contribute to diverse topics such as visual parameter space analysis (P5, P6), ensemble visualization (P6), guidance (P4), or temporal data analysis (P7).

P4 Davide Ceneda, Natalia Andrienko, Gennady Andrienko, Theresia Gschwandtner, Silvia Miksch, **Nikolaus Piccolotto**, Tobias Schreck, Marc Streit, Josef Suschnigg, Christian Tominski: *Guide Me in Analysis: A Framework for Guidance Designers*. Computer Graphics Forum, vol. 39, no. 6, 2020. DOI: 10.1111/cgf.14017.

- Contributed with: Resources.

P5 **Nikolaus Piccolotto**, Markus Bögl, Silvia Miksch: *Visual Parameter Space Exploration in Time and Space*. Computer Graphics Forum, vol. 42, no. 6, 2023. DOI: 10.1111/cgf.14785.

- Contributed with: Conceptualization, Methodology, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization.

P6 **Nikolaus Piccolotto**, Markus Bögl, Silvia Miksch: *Multi-Ensemble Visual Analytics via Fuzzy Sets*. EuroVis Workshop on Visual Analytics (EuroVA), 2023. DOI: 10.2312/eurova.20231092.

- Contributed with: Conceptualization, Methodology, Software, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization.

P7 Claudia Capello, **Nikolaus Piccolotto**, Christoph Muehlmann, Markus Bögl, Peter Filzmoser, Silvia Miksch, Klaus Nordhausen: *Visual Interactive Parameter Selection for Temporal Blind Source Separation*. Accepted for publication in Journal of Data Science, Statistics, and Visualisation.

- Contributed with: Conceptualization, Software, Writing - Original Draft, Writing - Review & Editing, Visualization.

1.4 Outline

This section describes the structure of the remainder of the thesis, which is illustrated in Figure 1.18 as a flow chart.

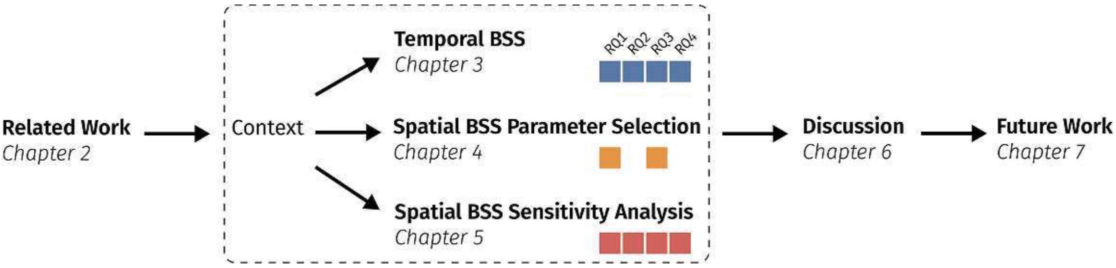


Figure 1.18: Structure of the thesis and relation between publications and research questions.

In Chapter 2, we discuss the state of the art and open challenges in the visualization literature related to BSS and our research questions. In particular, we cover visual parameter space analysis (Visual Parameter Space Analysis (vPSA)), temporal and spatial data visualization (including geovisual analytics), set visualization, ensemble visualization, dimensionality reduction (DR), and guidance. We discuss the relation and limitations of the visualization literature's state of the art to BSS and formulate possible future research directions for vPSA.

In Chapter 3, we present a VA prototype for TBSS supporting exploration of the tuning parameter space and identified latent components. We provide a general, i.e., not specific to temporal characteristics, task abstraction that likely applies to other latent variable models and BSS for other data types. Novel VA approaches include, e.g., a set-aware clustering method that we use to group latent components to provide an overview. The prototype was evaluated with usage scenarios on financial and healthcare data and interviews with five BSS experts.

In Chapter 4, we move on to SBSS and focus on the problem of parameter optimization. We suggest three tasks that a VA solution needs to support based on the user-centered design with our collaborators and a domain expert in geochemistry. We present a VA prototype implementing these tasks. The main features highlighted in the evaluation were the guidance suggestions for the complex regionalization tuning parameter and a direct manipulation interface for both tuning parameters. The evaluation included five visualization experts, two SBSS experts, and the geochemistry expert.

In Chapter 5, we still tackle SBSS but now focus on sensitivity analysis. We aim to use only dissimilarity measures based on the observation that neither SBSS inputs nor outputs neatly conform to multivariate data. We suggest five required tasks and present a VA prototype. We again included five visualization experts and two SBSS experts to evaluate the prototype. We showed the transferability of our prototype by interviewing an expert in microclimate simulations.

In Chapter 6, we discuss and reflect on how the publications presented in chapters 3–5 support our research questions (Section 6.1) and contributions (Section 6.2). Section 6.3 primarily focuses on the limitations of our research and possible gaps in the practical application of our results.

In Chapter 7, we conclude the thesis with an outline of possible directions for future research.

1.5 Bibliography

[Aig+23] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Second Edition. Springer, 2023. ISBN: 978-1-4471-7529-2 (cit. on p. 13).

[AOK19] L. Allen, A. O'Connell, and V. Kiermer. "How Can We Ensure Visibility and Diversity in Research Contributions? How the Contributor Role Taxonomy (CRediT) Is Helping the Shift from Authorship to Contributorship". In: *Learned Publishing* 32.1 (2019), pp. 71–74. DOI: 10.1002/leap.1210 (cit. on p. 25).

[Als+16] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers. "The State-of-the-Art of Set Visualization". In: *Computer Graphics Forum* 35.1 (2016), pp. 234–260. DOI: 10.1111/cgf.12722 (cit. on p. 3).

[And22] R. J. Andrews. *How Florence Nightingale Changed Data Visualization Forever*. Scientific American. 2022. URL: https://www.scientificamerican.com/article/how-florence-nightingale-changed-data-visualization-forever/ (visited on 01/26/2024) (cit. on p. 5).

[And+18] N. Andrienko, T. Lammarsch, G. Andrienko, G. Fuchs, D. Keim, S. Miksch, and A. Rind. “Viewing Visual Analytics as Model Building”. In: *Computer Graphics Forum* 37.6 (2018), pp. 275–299. DOI: 10.1111/cgf.13324 (cit. on pp. 10, 13).

[AA06] N. Andrienko and G. Andrienko. *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer Berlin Heidelberg New York, 2006. ISBN: 978-3-540-25994-7 (cit. on p. 13).

[Ang+18] M. Angelini, G. Santucci, H. Schumann, and H.-J. Schulz. “A Review and Characterization of Progressive Visual Analytics”. In: *Informatics* 5.3 (3 2018), p. 31. DOI: 10.3390/informatics5030031 (cit. on p. 14).

[Bac+17] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, and S. Carpendale. “A Descriptive Framework for Temporal Data Visualizations Based on Generalized Space-Time Cubes”. In: *Computer Graphics Forum* 36.6 (2017), pp. 36–61. DOI: 10.1111/cgf.12804 (cit. on p. 13).

[Bac+20] F. Bachoc, M. G. Genton, K. Nordhausen, A. Ruiz-Gazen, and J. Virta. “Spatial Blind Source Separation”. In: *Biometrika* 107.3 (2020), pp. 627–646. DOI: 10.1093/biomet/asz079 (cit. on pp. 21–23).

[BH19] L. Battle and J. Heer. “Characterizing Exploratory Visual Analysis: A Literature Review and Evaluation of Analytic Provenance in Tableau”. In: *Computer Graphics Forum* 38.3 (2019), pp. 145–159. DOI: 10.1111/cgf.13678 (cit. on p. 23).

[Bel+97] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines. “A Blind Source Separation Technique Using Second-Order Statistics”. In: *IEEE Transactions on Signal Processing* 45.2 (1997), pp. 434–444. DOI: 10.1109/78.554307 (cit. on p. 18).

[Ber83] J. Bertin. *Semiology of graphics*. Madison, Wis: University of Wisconsin Press, 1983. 415 pp. ISBN: 978-0-299-09060-9 (cit. on p. 5).

[Bis+17] A. Biswas, G. Lin, X. Liu, and H.-W. Shen. “Visualization of Time-Varying Weather Ensembles across Multiple Resolutions”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 841–850. DOI: 10.1109/tvcg.2016.2598869 (cit. on p. 3).

[BM13] M. Brehmer and T. Munzner. “A Multi-Level Typology of Abstract Visualization Tasks”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2376–2385. DOI: 10.1109/tvcg.2013.124 (cit. on p. 13).

[BM10] S. Bruckner and T. Möller. “Result-Driven Exploration of Simulation Parameter Spaces for Visual Effects Design”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 1468–1476. DOI: 10.1109/tvcg.2010.190 (cit. on p. 3).

[CMS99] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. 1st Edition. Morgan Kaufmann, 1999. ISBN: 1-55860-533-9 (cit. on pp. 7, 8).

[CRM91] S. K. Card, G. G. Robertson, and J. D. Mackinlay. "The Information Visualizer, an Information Workspace". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New Orleans, Louisiana, United States: ACM Press, 1991, pp. 181–186. ISBN: 978-0-89791-383-6. DOI: 10.1145/108844.108874 (cit. on p. 14).

[Cen+17] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H. Schulz, M. Streit, and C. Tominski. "Characterizing Guidance in Visual Analytics". In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 111–120. DOI: 10.1109/tvcg.2016.2598468 (cit. on p. 13).

[CMK20] A. Chatzimparmpas, R. M. Martins, and A. Kerren. "T-viSNE: Interactive Assessment and Interpretation of t-SNE Projections". In: *IEEE Transactions on Visualization and Computer Graphics* 26.8 (2020), pp. 2696–2714. DOI: 10.1109/tvcg.2020.2986996 (cit. on p. 3).

[CM84] W. S. Cleveland and R. McGill. "Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods". In: *Journal of the American Statistical Association* 79.387 (1984), pp. 531–554 (cit. on p. 13).

[CM88] W. S. Cleveland and M. E. McGill, eds. *Dynamic Graphics for Statistics*. Pacific Grove, Calif: Wadsworth & Brooks/Cole Advanced Books & Software, 1988. 424 pp. ISBN: 978-0-534-09144-6 (cit. on p. 7).

[CJ10] P. Comon and C. Jutten. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Academic Press, 2010 (cit. on pp. 1, 15–17).

[DAg23] L. D'Agostino McGowan. *quartets: Datasets to help teach statistics*. 2023 (cit. on p. 8).

[dLdMV00] L. de Lathauwer, B. de Moor, and J. Vandewalle. "Fetal Electrocardiogram Extraction by Blind Source Subspace Separation". In: *IEEE Transactions on Biomedical Engineering* 47.5 (2000), pp. 567–572. DOI: 10.1109/10.841326 (cit. on p. 18).

[Esp+23] M. Espadoto, G. Appleby, A. Suh, D. Cashman, M. Li, C. Scheidegger, E. W. Anderson, R. Chang, and A. C. Telea. "UnProjection: Leveraging Inverse-Projections for Visual Analytics of High-Dimensional Data". In: *IEEE Transactions on Visualization and Computer Graphics* 29.2 (2023), pp. 1559–1572. DOI: 10.1109/TVCG.2021.3125576 (cit. on p. 25).

[Fie20] K. Field. *Something in the water: the mythology of Snow's map of cholera*. https://www.esri.com/arcgis-blog/products/arcgis-pro/mapping/something-in-the-water-the-mythology-of-snows-map-of-cholera/. Accessed: 2023-08-19. 2020 (cit. on p. 5).

[FAM23] V. Filipov, A. Arleo, and S. Miksch. “Are We There Yet? A Roadmap of Network Visualization from Surveys to Task Taxonomies”. In: *Computer Graphics Forum* 42.6 (2023), e14794. DOI: `10.1111/cgf.14794` (cit. on p. 13).

[Fil23] P. Filzmoser. *StatDA: Statistical Analysis for Environmental Data.* R package version 1.7.11. 2023 (cit. on p. 20).

[Fra02] C. A. Franklin. “MATH ROOTS: The Other Life of Florence Nightingale”. In: *Mathematics Teaching in the Middle School* 7.6 (2002), pp. 337–340 (cit. on p. 5).

[Fri08] M. Friendly. “A Brief History of Data Visualization”. In: *Handbook of Data Visualization.* Ed. by C.-h. Chen, W. Härdle, and A. Unwin. Berlin, Heidelberg: Springer, 2008, pp. 15–56. ISBN: 978-3-540-33037-0. DOI: `10.1007/978-3-540-33037-0_2` (cit. on p. 4).

[Fun36] H. G. Funkhouser. “A Note on a Tenth Century Graph”. In: *Osiris* 1 (1936), pp. 260–262 (cit. on p. 4).

[GK15] M. G. Genton and W. Kleiber. “Cross-Covariance Functions for Multivariate Geostatistics”. In: *Statistical Science* 30.2 (2015), pp. 147–163. DOI: `10.1214/14-sts487` (cit. on p. 21).

[He+19] X. He, Y. Tao, Q. Wang, and H. Lin. “Multivariate Spatial Data Visualization: A Survey”. In: *Journal of Visualization* 22.5 (2019), pp. 897–912. DOI: `10.1007/s12650-019-00584-3` (cit. on p. 13).

[Jeo+22] H. Jeon, H.-K. Ko, J. Jo, Y. Kim, and J. Seo. “Measuring and Explaining the Inter-Cluster Reliability of Multidimensional Projections”. In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2022), pp. 551–561. DOI: `10.1109/tvcg.2021.3114833` (cit. on p. 25).

[Jeo+09] D. H. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang. “iPCA: An Interactive System for PCA-based Visual Analytics”. In: *Computer Graphics Forum* 28.3 (2009), pp. 767–774. DOI: `10.1111/j.1467-8659.2009.01475.x` (cit. on p. 3).

[Kei+08] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. “Visual Analytics: Definition, Process, and Challenges”. In: *Information Visualization.* Ed. by A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North. 4950. Berlin, Heidelberg: Springer, 2008, pp. 154–175. ISBN: 978-3-540-70955-8 978-3-540-70956-5. DOI: `10.1007/978-3-540-70956-5_7` (cit. on pp. 3, 9, 13).

[Kei+10] D. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann. *Mastering the Information Age Solving Problems with Visual Analytics.* Eurographics Association, 2010. ISBN: 978-3-905673-77-7. DOI: `http://diglib.eg.org/handle/10.2312/14803` (cit. on p. 10).

[Lam+12] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. “Empirical Studies in Information Visualization: Seven Scenarios”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.9 (2012), pp. 1520–1536. DOI: `10.1109/tvcg.2011.279` (cit. on p. 14).

[LTM18] H. Lam, M. Tory, and T. Munzner. “Bridging from Goals to Tasks with Design Study Analysis Reports”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2018), pp. 435–445. DOI: `10.1109/tvcg.2017.2744319` (cit. on p. 13).

[Liu+15] L. Liu, M. Mirzargar, R. M. Kirby, R. Whitaker, and D. H. House. “Visualizing Time-Specific Hurricane Predictions, with Uncertainty, from Storm Path Ensembles”. In: *Computer Graphics Forum* 34.3 (2015), pp. 371–380. DOI: `10.1111/cgf.12649` (cit. on p. 3).

[Liu+19] X. Liu, H. Wang, M. Huang, and W. Yang. “An Improved Second-Order Blind Identification (SOBI) Signal De-Noising Method for Dynamic Deflection Measurements of Bridges Using Ground-Based Synthetic Aperture Radar (GBSAR)”. In: *Applied Sciences* 9.17 (2019), p. 3561. DOI: `10.3390/app9173561` (cit. on p. 16).

[Mac86] J. Mackinlay. “Automating the Design of Graphical Presentations of Relational Information”. In: *ACM Transactions on Graphics* 5.2 (1986), pp. 110–141. DOI: `10.1145/22949.22950` (cit. on p. 13).

[MF17] J. Matejka and G. Fitzmaurice. “Same Stats, Different Graphs: Generating Datasets with Varied Appearance and Identical Statistics Through Simulated Annealing”. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA). New York, NY, USA: ACM, 2017, pp. 1290–1294. ISBN: 978-1-4503-4655-9. DOI: `10.1145/3025453.3025912` (cit. on p. 8).

[Mat+17] M. Matilainen, J. Miettinen, K. Nordhausen, H. Oja, and S. Taskinen. “On Independent Component Analysis with Stochastic Volatility Models”. In: *Austrian Journal of Statistics* 46.3-4 (2017), pp. 57–66. DOI: `10.17713/ajs.v46i3-4.671` (cit. on p. 19).

[McC+22] C. M. McColeman, F. Yang, T. F. Brady, and S. Franconeri. “Rethinking the Ranks of Visual Channels”. In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2022), pp. 707–717. DOI: `10.1109/tvcg.2021.3114684` (cit. on p. 13).

[MDB87] B. McCormick, T. DeFanti, and M. Brown. “Visualization in Scientific Computing”. In: *Computer Graphics* 21.6 (1987) (cit. on p. 7).

[MHM18] L. McInnes, J. Healy, and J. Melville. “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction”. In: *arXiv* (2018). DOI: `10.48550/arXiv.1802.0342` (cit. on p. 1).

[McL+08] P. McLachlan, T. Munzner, E. Koutsofios, and S. North. “LiveRAC: Interactive Visual Exploration of System Management Time-series Data”. In: *Proceedings of the 2008 CHI Conference on Human Factors in Computing Systems* (Florence, Italy). New York, NY, USA: ACM, 2008, pp. 1483–1492. ISBN: 978-1-60558-011-1. DOI: 10.1145/1357054.1357286 (cit. on p. 3).

[MD20] M. Meyer and J. Dykes. “Criteria for Rigor in Visualization Design Study”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2020), pp. 87–97. DOI: 10.1109/tvcg.2019.2934539 (cit. on p. 11).

[Mie+20] J. Miettinen, M. Matilainen, K. Nordhausen, and S. Taskinen. “Extracting Conditionally Heteroskedastic Components Using Independent Component Analysis”. In: *Journal of Time Series Analysis* 41.2 (2020), pp. 293–311. DOI: 10.1111/jtsa.12505 (cit. on pp. 2, 19, 20).

[MA14] S. Miksch and W. Aigner. “A Matter of Time: Applying a Data-users-tasks Design Triangle to Visual Analytics of Time-oriented Data”. In: *Computers & Graphics* 38 (2014), pp. 286–290. DOI: 10.1016/j.cag.2013.11.002 (cit. on p. 12).

[Min70] C.-J. Minard. *Tableaux Graphiques et Cartes Figuratives*. Paris, 1870 (cit. on p. 4).

[Mue21] C. Muehlmann. “Advances in Blind Source Separation for Spatial Data”. PhD thesis. Vienna, Austria: TU Wien, 2021, p. 100574. DOI: 10.1016/j.spasta.2021.100574 (cit. on p. 16).

[MBN22] C. Muehlmann, F. Bachoc, and K. Nordhausen. “Blind Source Separation for Non-Stationary Random Fields”. In: *Spatial Statistics* 47 (2022), p. 100574. DOI: 10.1016/j.spasta.2021.100574 (cit. on pp. 2, 22, 23).

[MNY20] C. Muehlmann, K. Nordhausen, and M. Yi. “On Cokriging, Neural Networks, and Spatial Blind Source Separation for Multivariate Spatial Prediction”. In: *IEEE Geoscience and Remote Sensing Letters* 18 (2020), pp. 1931–1935. DOI: 10.1109/lgrs.2020.3011549 (cit. on p. 16).

[Mun09] T. Munzner. “A Nested Model for Visualization Design and Validation”. In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (2009), pp. 921–928. DOI: 10.1109/tvcg.2009.111 (cit. on pp. 11, 12).

[Mun14] T. Munzner. *Visualization Analysis and Design*. CRC Press, 2014. ISBN: 978-1-4987-5971-7 (cit. on pp. 7, 11, 13–15).

[Nig59] F. Nightingale. *A Contribution to the Sanitary History of the British Army during the Late War with Russia*. London: John W. Parker and Son, 1859 (cit. on p. 7).

[NA19] L. G. Nonato and M. Aupetit. "Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Enrichment". In: *IEEE Transactions on Visualization and Computer Graphics* 25.8 (2019), pp. 2650–2673. DOI: 10.1109/tvcg.2018.2846735 (cit. on p. 13).

[NFF21] K. Nordhausen, G. Fischer, and P. Filzmoser. "Blind Source Separation for Compositional Time Series". In: *Mathematical Geosciences* 53.5 (2021), pp. 905–924. DOI: 10.1007/s11004-020-09869-y (cit. on p. 18).

[NO18] K. Nordhausen and H. Oja. "Independent Component Analysis: A Statistical Perspective". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 10.5 (2018), e1440. DOI: 10.1002/wics.1440 (cit. on p. 16).

[Nor+15] K. Nordhausen, H. Oja, P. Filzmoser, and C. Reimann. "Blind Source Separation for Spatial Compositional Data". In: *Mathematical Geosciences* 47.7 (2015), pp. 753–770. DOI: 10.1007/s11004-014-9559-5 (cit. on pp. 20, 21).

[NR22] K. Nordhausen and A. Ruiz-Gazen. "On the Usage of Joint Diagonalization in Multivariate Statistics". In: *Journal of Multivariate Analysis* 188 (2022), p. 104844. DOI: 10.1016/j.jmva.2021.104844 (cit. on p. 16).

[Nor94] D. A. Norman. *Things That Make Us Smart*. Reprint edition. New York: Basic Books, 1994. 304 pp. ISBN: 978-0-201-62695-7 (cit. on p. 7).

[OKM00] E. Oja, K. Kiviluoto, and S. Malaroiu. "Independent Component Analysis for Financial Time Series". In: *Proceedings of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*. New York, NY, USA: IEEE, 2000, pp. 111–116. DOI: 10.1109/asspcc.2000.882456 (cit. on p. 18).

[Pan+21] Y. Pan, M. Matilainen, S. Taskinen, and K. Nordhausen. "A Review of Second-Order Blind Identification Methods". In: *WIREs Computational Statistics* 14.4 (2021), e1550. DOI: 10.1002/wics.1550 (cit. on pp. 16, 20).

[PW10] R. Prado and M. West. *Time Series: Modeling, Computation, and Inference*. 1st Edition. Chapman and Hall/CRC, 2010. ISBN: 978-1-4398-8275-7 (cit. on p. 18).

[R C23] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2023 (cit. on p. 7).

[Rei+14] C. Reimann, M. Birke, A. Demetriades, P. Filzmoser, and P. O'Connor. *Chemistry of Europe's Agricultural Soils. Part A: Methodology and Interpretation of the GEMAS Data Set*. Vol. Heft 102. Reihe B. Hannover: Bundesanstalt für Geowissenschaften und Rohstoffe (BGR), 2014 (cit. on p. 22).

[Rei+98] C. Reimann, M. Äyräs, V. A. Chekushin, I. V. Bogatyrev, R. Boyd, P. de Caritat, R. Dutter, T. E. Finne, J. H. Halleraker, J. Øystein, G. Kashulina, H. Niskavaara, et al. *Environmental Geochemical Atlas of the Central Barents Region*. Trondheim: NGU Norges Geologiske Undersøkelse, 1998. ISBN: 82-7385-176-1 (cit. on pp. 20, 21).

[Rei+08] C. Reimann, P. Filzmoser, R. Garrett, and R. Dutter. *Statistical Data Analysis Explained*. First Edition. John Wiley & Sons, Ltd, 2008. ISBN: 978-0-470-98581-6. DOI: 10.1002/9780470987605 (cit. on p. 21).

[Rot13] R. E. Roth. “An Empirically-Derived Taxonomy of Interaction Primitives for Interactive Cartography and Geovisualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2356–2365. DOI: 10.1109/tvcg.2013.130 (cit. on p. 13).

[Sac+14] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim. “Knowledge Generation Model for Visual Analytics”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1604–1613. DOI: 10.1109/tvcg.2014.2346481 (cit. on p. 13).

[Sae+18] N. Saeed, H. Nam, M. I. U. Haq, and D. B. Muhammad Saqib. “A Survey on Multidimensional Scaling”. In: *ACM Computing Surveys* 51.3 (2018) (cit. on p. 1).

[SMM12] M. Sedlmair, M. Meyer, and T. Munzner. “Design Study Methodology: Reflections from the Trenches and the Stacks”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), pp. 2431–2440. DOI: 10.1109/tvcg.2012.213 (cit. on pp. 11, 12, 14).

[Sed+12] M. Sedlmair, M. Brehmer, S. Ingram, and T. Munzner. *Dimensionality Reduction in the Wild: Gaps and Guidance*. Technical Report Tr-2012-03. Vancouver, BC, Canada: Dept. Comput. Sci., Univ. British Columbia, 2012 (cit. on p. 1).

[Sed+14] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. “Visual Parameter Space Analysis: A Conceptual Framework”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2161–2170. DOI: 10.1109/tvcg.2014.2346321 (cit. on pp. 2, 13).

[Shn96] B. Shneiderman. “The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations”. In: *Proceedings of the 1996 IEEE Symposium on Visual Languages*. 1996, pp. 336–343. DOI: 10.1109/vl.1996.545307 (cit. on p. 12).

[Sil+15] R. R. O. da Silva, P. E. Rauber, R. M. Martins, R. Minghim, and A. C. Telea. “Attribute-Based Visual Explanation of Multidimensional Projections”. In: *EuroVis Workshop on Visual Analytics (EuroVA)* (2015), 5 pages. DOI: 10.2312/eurova.20151100 (cit. on p. 24).

[Sno54] J. Snow. *On the Mode of Communication of Cholera*. Second Edition, much enlarged. John Churchill, New Burlington Street, 1854 (cit. on pp. 5, 6).

[Soh+22] J.-T. Sohns, M. Schmitt, F. Jirasek, H. Hasse, and H. Leitte. “Attribute-Based Explanation of Non-Linear Embeddings of High-Dimensional Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2022), pp. 540–550. DOI: `10.1109/tvcg.2021.3114870` (cit. on p. 24).

[Sou+22] L. South, D. Saffo, O. Vitek, C. Dunne, and M. Borkin. “Effective Use of Likert Scales in Visualization Evaluations: A Systematic Review”. In: *Computer Graphics Forum* 41.3 (2022). DOI: `10.31219/osf.io/6f3zs` (cit. on p. 15).

[Sta+16] J. Stahnke, M. Dörk, B. Müller, and A. Thom. “Probing Projections: Interaction Techniques for Interpreting Arrangements and Errors of Dimensionality Reductions”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 629–638. DOI: `10.1109/tvcg.2015.2467717` (cit. on p. 25).

[TLS05] A. C. Tang, J.-Y. Liu, and M. T. Sutherland. “Recovery of Correlated Neuronal Sources from EEG: The Good and Bad Ways of Using SOBI”. In: *NeuroImage* 28.2 (2005), pp. 507–519. DOI: `10.1016/j.neuroimage.2005.06.062` (cit. on p. 20).

[TMN16] S. Taskinen, J. Miettinen, and K. Nordhausen. “A More Efficient Second Order Blind Identification Method for Separation of Uncorrelated Stationary Time Series”. In: *Statistics & Probability Letters* 116 (2016), pp. 21–26. DOI: `10.1016/j.spl.2016.04.007` (cit. on p. 20).

[TI06] F. Theis and Y. Inouye. “On the Use of Joint Diagonalization in Blind Signal Processing”. In: *International Symposium on Circuits and Systems*. IEEE, 2006. DOI: `10.1109/iscas.2006.1693402` (cit. on p. 16).

[TC05] J. J. Thomas and K. A. Cook. *Illuminating the Path: An R&D Agenda for Visual Analytics*. National Visualization and Analytics Center, Department of Homeland Security, IEEE Computer Society, 2005. ISBN: 0-7695-2323-4 (cit. on pp. 9, 10).

[Tob70] W. R. Tobler. “A Computer Movie Simulating Urban Growth in the Detroit Region”. In: *Economic Geography* 46 (1970), p. 234. DOI: `10.2307/143141` (cit. on pp. 16, 17).

[Tom+17] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann. “Interactive Lenses for Visualization: An Extended Survey”. In: *Computer Graphics Forum* 36.6 (2017), pp. 173–200. DOI: `10.1111/cgf.12871` (cit. on p. 9).

[Ton+90] L. Tong, V. C. Soon, Y. F. Huang, and R. Liu. “AMUSE: A New Blind Identification Algorithm”. In: *International Symposium on Circuits and Systems (ISCAS)*. 1990, pp. 1784–1787. DOI: `10.1109/iscas.1990.111981` (cit. on p. 18).

[Tuf01] E. R. Tufte. *The Visual Display of Quantitative Information*. Second Edition. Graphics Press, 2001. ISBN: 978-1-930824-13-3 (cit. on pp. 4–6).

[Tuk77] J. W. Tukey. *Exploratory Data Analysis*. Reading, Mass: Addison-Wesley Pub. Co, 1977. 688 pp. ISBN: 978-0-201-07616-5 (cit. on pp. 6, 7, 23).

[Tur+14] C. Turkay, A. Slingsby, H. Hauser, J. Wood, and J. Dykes. "Attribute Signatures: Dynamic Visual Summaries for Analyzing Multivariate Geographical Data". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2033–2042. DOI: 10.1109/tvcg.2014.2346265 (cit. on p. 3).

[vdMH08] L. van der Maaten and G. Hinton. "Visualizing Data Using T-SNE". In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605 (cit. on p. 1).

[vWij05] J. J. van Wijk. "The Value of Visualization". In: *VIS 05. IEEE Visualization, 2005*. IEEE, 2005, p. 8. DOI: 10.1109/visual.2005.1532781 (cit. on pp. 9, 12, 13).

[Wac03] H. Wackernagel. *Multivariate geostatistics: an introduction with applications*. Berlin, Heidelberg: Springer Science & Business Media, 2003. DOI: 10.1007/978-3-662-05294-5 (cit. on p. 21).

[Wal+19] E. Wall, M. Agnihotri, L. Matzen, K. Divis, M. Haass, A. Endert, and J. Stasko. "A Heuristic Approach to Value-Driven Evaluation of Visualizations". In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 491–500. DOI: 10.1109/tvcg.2018.2865146 (cit. on p. 15).

[War04] C. Ware. *Information Visualization: Perception for Design*. Second Editon. Morgan Kaufmann, 2004. ISBN: 1-55860-819-2 (cit. on p. 7).

[Wic16] H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN: 978-3-319-24277-4 (cit. on p. 7).

[YKS07] J. S. Yi, Y. ah Kang, and J. Stasko. "Toward a Deeper Understanding of the Role of Interaction in Information Visualization". In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1224–1231. DOI: 10.1109/tvcg.2007.70515 (cit. on p. 13).

[Yos+20] A. Yoshizumi, M. M. Coffer, E. L. Collins, M. D. Gaines, X. Gao, K. Jones, I. R. McGregor, K. A. McQuillan, V. Perin, L. M. Tomkins, T. Worm, and L. Tateosian. "A Review of Geospatial Content in IEEE Visualization Publications". In: *2020 IEEE Visualization Conference (VIS)*. 2020. DOI: 10.1109/vis47514.2020.00017 (cit. on p. 13).

[YHX14] X. Yu, D. Hu, and J. Xu. *Blind Source Separation: Theory and Applications*. Singapore: John Wiley & Sons Singapore Pte. Ltd, 2014. ISBN: 978-1-118-67984-5 (cit. on pp. 1, 15, 16).

CHAPTER 2

Related Work

As touched upon in Chapter 1, the most relevant related fields in visualization to BSS are Visual Parameter Analysis, Temporal & Spatial Data Visualization, Set Visualization, Ensemble Visualization, and Dimensionality Reduction. In the following sections, we will introduce each of them and discuss their relation to BSS in more detail.

2.1 Visual Parameter Space Analysis

Parameter analysis is concerned with understanding the relation between the parameters and outputs of a simulation model. Sedlmair et al. define it as:

> "Parameter Space Analysis (PSA) is the systematic variation of model input parameters, generating outputs for each combination of parameters, and investigating the relation between parameter settings and corresponding outputs." [Sed+14, p. 2162]

Many state-of-the-art simulation models require multiple parameters to be set in order to use them effectively. For example, it is highly interesting to domain experts if settings of some parameters are less critical than of others, or whether some combination of parameter settings leads to unrealistic outputs or even model failure. As it is often a-priori unclear what exactly one is looking for, visualization proved to be very useful, thus leading to vPSA. vPSA refers to PSA, supported by interactive visualizations. Santner et al. [SWN03] distinguish between control, environmental, and model parameters. Environmental parameters describe properties of the real world [LSN04], such as the conductivity of transistors or the bone density of a patient. They are usually not under the analyst's control, which is why parameter analysis tends to focus on control and model parameters. The difference between the two is that model parameters, such as grid sizes, are necessary for numerical purposes of the model but usually hidden during model usage. Nevertheless, model parameters impact the model output and are thus just as interesting to investigate as control parameters [Bis+17; Wan+17]. In the following we refer to both of them as "parameters."

Data Flow Model. Sedlmair et al. [Sed+14] surveyed the field of vPSA and identified a common data flow model (Figure 2.1). The model is viewed as a function mapping some input onto some output. Notable is the "surrogate model," which is a less accurate but faster version of the original model, e.g., a trained neural network [Haz+20] or linear regression [Mat+14]. It is derived from outputs of the original model and its input parameters. While the predicted outputs of the surrogate model may not be entirely accurate, it enables interactive visual analysis when the original model is too computationally expensive. The reduced accuracy is usually not seen as an issue, as uncertainty quantification techniques may express the surrogate model's confidence in its prediction or, as a last resort, one could always revert to the original model.

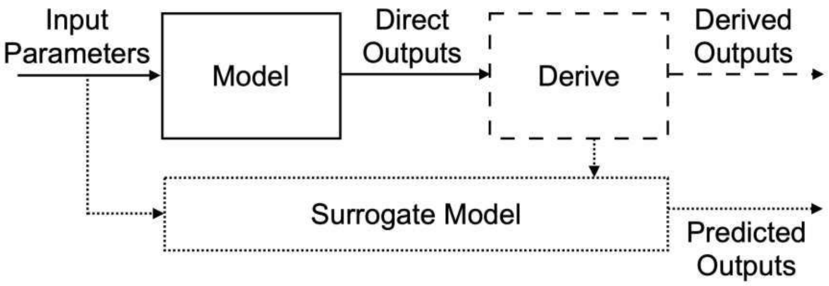


Figure 2.1: Data Flow Model for vPSA. Image: [Sed+14, Fig. 6] © 2014 IEEE

Analytic Tasks. Further, the authors [Sed+14] distilled six common analytic tasks pertaining to parameters and ouputs of a model: Optimization, sensitivity, uncertainty, fitting, outliers, and partitioning. In parameter **optimization**, analysists seek the parameter settings that lead to the "best" output according to some definition. If objective functions can be formulated, numerical optimization, like (integer) linear programming, is applicable. Having a human in the loop can help domain experts consider, maybe even accept, such approaches [Liu+18; Liu+21]. **Sensitivity** analysis asks "what ranges/variations to expect with changes of input" [Sed+14, p. 2166]. Clearly, more attention must be paid during model use to parameters to which the output is highly dependent. If parameters and output are, or can be expressed as, multivariate data, a multitude of numerical sensitivity analysis techniques are available [Ham94; IL15; BP16]. Sedlmair et al. [Sed+14] mention that they found sensitivity analysis to often have cross-cutting concerns, which is also true for BSS: A method is expected to be more likely to be the "true" solution if it comes from a stable parameter subspace. **Uncertainty** analysis is necessary when one seeks to understand the reliability of model outputs: How far off were, e.g., precipitation predictions compared to what was measured? **Fitting**

turns the data flow model upside down, in a way, because it asks about the inverse relationship: Which parameters would yield a given output? This is sometimes referred to as “inverse design” [Cof+13] when parameter settings are optimized by sculpting the desired output directly. **Partitioning** is akin to an overview task, as it asks about all possible model behaviors. Partitioning is usually only achievable with extensive sampling of parameter settings. Finally, **outliers** seeks to find special outputs that do not behave like the others. These may be model failures or extreme but still valid outputs.

Navigation Strategies. Sedlmair et al. [Sed+14] also suggest four parameter space navigation strategies that are commonly employed. **Informed trial-and-error** is the default: A loop in which a parameter setting is manually selected, the output inspected, and the model re-run with modified parameter settings. Large interruption costs between the second and third step are the main problem of the approach. In **local-to-global** navigation, the analyst starts with one output and explores alternative outputs from there step-by-step in a systematic fashion. E.g., by navigating from a biopsy device design (input) with some specific stress on its surface (output) to another design that minimizes stress on a particular part on the surface [Cof+13]. On the other hand, in **global-to-local** approaches, the analyst starts with an overview of all outputs and drills down into details from there. Finally, the analyst influences the simulation as it runs in **steering** approaches. The simulations may calculate water volume during a flood and the analyst’s interventions be barriers placed at specific points in time [Was+10]. Another example are interactive genetic optimization techniques that ask for user preference every couple of hundred generations [Mar13].

While the survey by Sedlmair et al. [Sed+14] is a great resource for visualization designers, it does, aside from parameter space navigation strategies, not discuss how vPSA approaches employ visualizations and interactions to achieve their goal. Further, returning to the focus of this thesis, it does not discuss how time and space are handled. BSS parameters relate to time and space, as do the outputs, which are sets of time series or maps. The accepted principle in visualization literature is that time and space have unique characteristics, e.g., seasonal cycles in time or Tobler’s law in space (“everything is related to everything else, but near things are more related than distant things” [Tob70, p. 236]). Therefore, time and space should be visualized as such and may not be treated as some other numeric variables. The authors [Sed+14] distinguish between inputs/outputs as multivariate/multidimensional and “complex objects” (i.e., something semantically more meaningful than a few numbers), but temporal/spatial data could fall into either category. We will discuss these matters in the next section.

2.1.1 Visualizations & Interactions in Visual Parameter Space Analysis of Temporal and Spatial Data

> This subsection is based on:
>
> Nikolaus Piccolotto, Markus Bögl, Silvia Miksch. *Visual Parameter Space Exploration in Time and Space*. Computer Graphics Forum, vol. 42, no. 6, 2023. DOI: 10.1111/cgf.14785.

In [PBM23b] we surveyed interactive visualizations for vPSA and identified five common themes (Figure 2.2) describing how they work towards solving PSA tasks. Consider a time series segmentation model [Ber+18; EST20]. The model inputs are a multivariate time series, e.g., motion sensor data, and some scalar parameters concering the segmentation process. The model produces a labeled time series, e.g., activities. Analysts may look for a reasonable labeling, i.e., one that is not overly *sensitive* to particular parameter settings. As a first step, analysts must identify interesting parameter settings to investigate (**Finding Parameter Settings**). In this case, the vPSA system computes segmentations for a uniformly random sampling of the parameter space. The obtained parameter/output pairs are then visualized to support the intended analysis (**Input/Output Visualization**). E.g., parameters and outputs may be shown in a tabular visualization using grayscale color for parameter values and color hue for labels. Others may depict derived data, like how much changes in a parameter correlate with changes in a label's occurrence. The analyst then interacts with the visualizations according to current information needs (**Data Case Organization**), e.g., by zooming into a temporal interval of interest, sorting the table by a column, or defining new derived attributes. In doing so, the analyst formulates hypotheses from gained insights [Sac+14], e.g., what a reasonable parameter subspace would be, and acts upon them to verify. This verification may entail changing how the model itself behaves (**(Surrogate) Model Tuning**) or repeating the analysis on a smaller parameter subspace. The analyst keeps track of sensible candidates via bookmarking or saving the parameter settings to a file (**Provenance**).

Terminology and Glyphs. A *model* transforms some input to some output. It can be an existing algorithm, a faster but less accurate "surrogate" to some existing algorithm (usually the case in connection with simulations), or a set of building blocks that perform a specific task, like a processing pipeline. We distinguish between three types of *data cases*: Static inputs (often called *input data*), dynamic inputs (*parameters*), and *output* of a model. The difference between static and dynamic inputs is that the latter take on varying settings to complete a parameter space analysis task [Sed+14], while the other remains static throughout the analysis. We further distinguish between three data characteristics: Spatial (S), temporal (T), and abstract (A) data. Spatial data refers to spatial primitives, like points or volumes. Temporal data refers to temporal primitives, like instants or intervals, and abstract data to tensors. The three characteristics are denoted by glyphs comprised of three hexagons: S, T, and A. Spatially and temporally varying data arises by combining the three characteristics, e.g., a multivariate time series has both

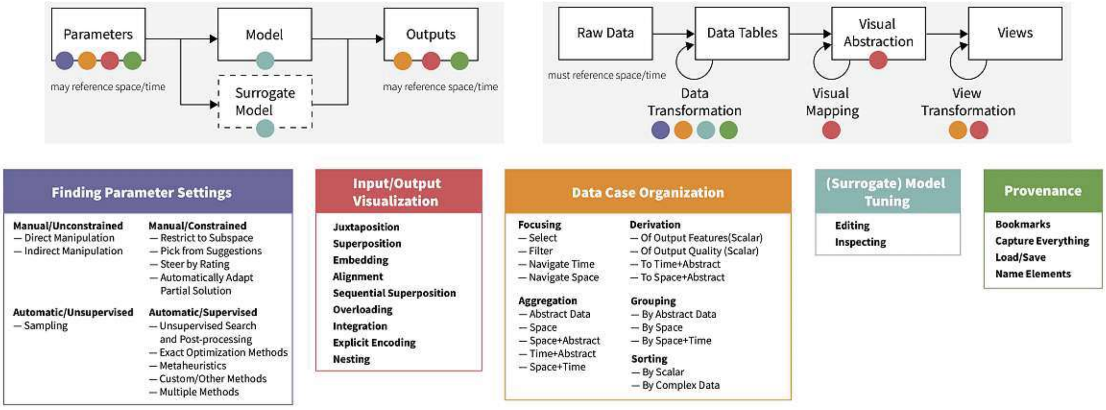


Figure 2.2: Major themes in vPSA approaches and their relation to the Data Flow Model by Sedlmair et al. [Sed+14] (top left) and the InfoVis pipeline by Card et al. [CMS99] (top right).

temporal and abstract features (AT) as an associated vector of variables exists for each time instant. Abstract, spatial, and temporal characteristics amount to seven possible combinations.

Method. We chose three seed papers for our literature search: Another survey of visual parameter analysis [Sed+14], a survey of data processing pipelines [vLFR17], and a popular example of visual parameter analysis [Tor+11]. We build a pool of candidate papers by conducting forward and backward search starting from the seed papers. In the end, we had 526 papers to consider. 103 papers we considered too old (published before 2010) or duplicates of others. Finally, we excluded 322 more papers as they were not about PSA, did not feature interactive visualizations, or did not consider temporal or spatial parameters or outputs. From the remaining 101 papers, we randomly selected 57 to carry out Reflexive Thematic Analysis (RTA) [BC06; BC19; BC21a; BC21b]. RTA is a method to develop themes from qualitative datasets, such as interviews, videos, or research papers. In contrast to codebook or coding reliability approaches, RTA embraces that the researcher *develops* themes from the dataset and that they do not exist independently. RTA [BC06; BC19] devises quality control steps in its process, which we followed. The remaining 44 papers were used as a "test set," like in a machine learning context, to verify the applicability of developed themes. As our themes are rather general, we encountered no issues in that process.

In the following sections we briefly introduce and present the mentioned themes.

Finding Parameter Settings

This theme considers interactions that lead to new (i.e., not previously analyzed) parameter settings and outputs added to the underlying data table. We distinguish broadly how those parameter settings are obtained: Manually, either constrained to a particular parameter subspace or not, and automatically, either supervised or unsupervised. Figure 2.3 shows a S parameter (polygon) and illustrates the Finding Parameter Settings sub-themes. We can imagine an algorithm that evaluates the roundness of the shape as our model. With Manual/Unconstrained, the parameter may be edited at will, thus taking any setting. As a result, any shape is possible. With Manual/Constrained, the parameter is restricted to a subspace, in this case, a ring: The currently edited vertex may be moved anywhere inside the subspace. Automatic techniques obtain parameter settings without or with little user interaction. Unsupervised approaches, like random sampling, traverse the parameter space independent of the output. Consequently, they may obtain very un-round shapes. On the other hand, output quality (roundness) guides supervised approaches' parameter space traversal. In our example, they may, e.g., only visit convex shapes. Regarding parameter space analysis tasks, we find that Manual/Constrained and Automatic/Supervised are commonly used to support *optimization* tasks, while the other two sub-themes do not have a clear preference.

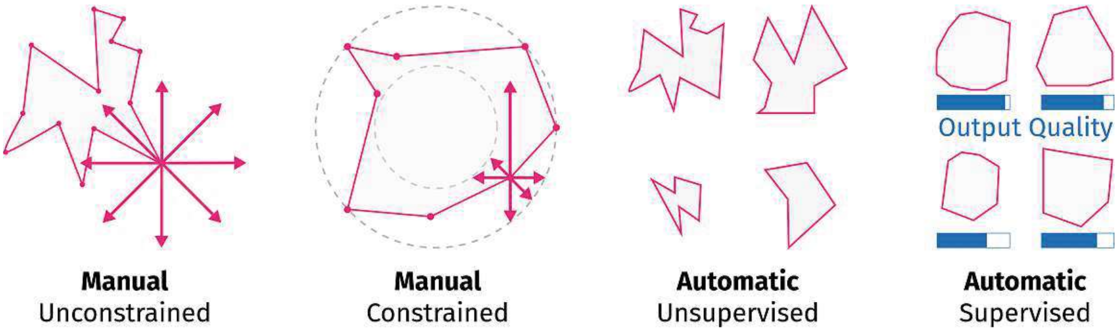


Figure 2.3: Finding Parameter Settings sub-themes.

Manual / Unconstrained. We classified papers to support unconstrained manual input when the user can enter any parameter setting supported by the model. Regarding how manual interactions with parameter spaces work, we can distinguish between indirect and direct manipulation. Direct manipulation, as defined by Shneiderman [Shn83], is characterized by i) continuous representations of objects of interest, ii) physical actions instead of textual commands, and iii) rapid, incremental, and reversible actions. An example of direct manipulation of an abstract parameter can be found in interactive PCPs [MW20], while indirect manipulation would constitute every input method using form controls [Rup+14]. Direct manipulation of a S parameter would be to directly edit the spatial representation, e.g., by growing/shrinking parts of a biopsy device with drag and drop (Figure 2.4). Indirect manipulation of such a parameter may happen through sliders for a parametric representation of it [Sch+17]. While it is widely agreed that direct manipulation is superior to indirect manipulation, the latter can still be very effective if the system is interactive enough [KP10; He+20]. Contrary to the indirect manipulation approaches, which work with abstract parameters, parameters often have a temporal/spatial component in the direct manipulation group. They are manipulated in any way that makes sense for the application domain: Wing shapes are drawn [Ume+14], as are walls [Ste+17] or shadows [Lin+13], time windows resized [Bor+17], and furniture is moved/rotated [Mer+11], through mouse operations on the visual representations. Novel input methods and modalities were explored sometimes, too. Kazi et al. [Kaz+17] explored how generative modeling can be used to support the design stage. Within their system, DreamSketch, the designer sketches a design problem, such as a load-bearing wall mount, using pen and tablet. The system then finds optimal solutions for varying combinations of design variables, which can be browsed within the sketch. Mohiuddi and Woodbury [MW20] explored a direct manipulation paradigm for a parametric representation of a S parameter (building design in architecture). Referring to Balling [Bal99], they argue that "designers prefer direct engagement and manual exploration" over automated sampling [MW20, LBW289, Page 4]. Hence, they propose novel interaction techniques for PCPs, such as sketching polylines, parallel editing, and quick generation of alternatives with operators, such as a cartesian product.

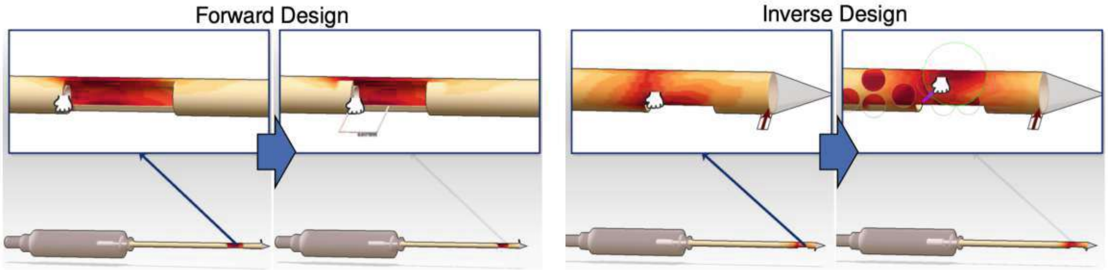


Figure 2.4: Forward and inverse design with direct manipulation of a canule (S parameter); stress on surface (AS output) is shown embedded to the design. Image: [Cof+13, Fig. 1] © 2013 IEEE

Manual / Constrained. We classified papers as supporting constrained manual input when entering a parameter setting is still manual, but the system does not allow the user to enter or develop arbitrary parameter settings, even though the model would support them. The system often expects the available parameter subspace to lead to higher-quality outputs. However, the restriction may also be a UI design decision to grapple with high-dimensional parameter spaces. As with all input modes, this can be optional and in addition to other modes available in the system. We further distinguish four approaches. **Restrict to Subspace** occurs when the system allows free selection only in a continuous parameter subspace. E.g., Brunhart-Lupo et al. [Bru+16] restricted A parameter selection to two dimensions with a "Parallel Planes" visualization in virtual reality. With **Pick from Suggestions**, the system suggests discrete parameter settings, i.e., points in the available subspace. These suggestions can be accepted, usually replacing the current setting. Suggestions were used in interior design [Mer+11; Wal+20], shelf design [UIM12], image processing [KSI14], robot design [Des+19], or graphic design [OAH15; Day+20]. **Steer by Rating** works by shrinking the available subspace step by step until it is so small that it can be considered a point, i.e., the desired solution. E.g., Koyama et al. proposed repeatedly searching along lines [Koy+17] and on planes [KSG20] via selection from galleries. Finally, with **Automatically Adapt Partial Solution**, the user provides the parameter subspace via a partial solution, and the system adapts it according to some objective. Liu et al. [Liu+21] recommend this strategy as part of their design guidelines for interactive optimization systems. Apart from their work, we found it, e.g., in systems using sliders to select parameter settings, where the user may lock slider values and let the system automatically set free sliders [KSI14; Yum+15; Des+19].

Input/Output Visualization

Many PSA tasks ask about a relation between parameters and output. Thus, an important high-level goal in vPSA is to reconcile and compare the parameter and output spaces of the model and this theme explores how it can be supported with visualizations (Figure 2.5).

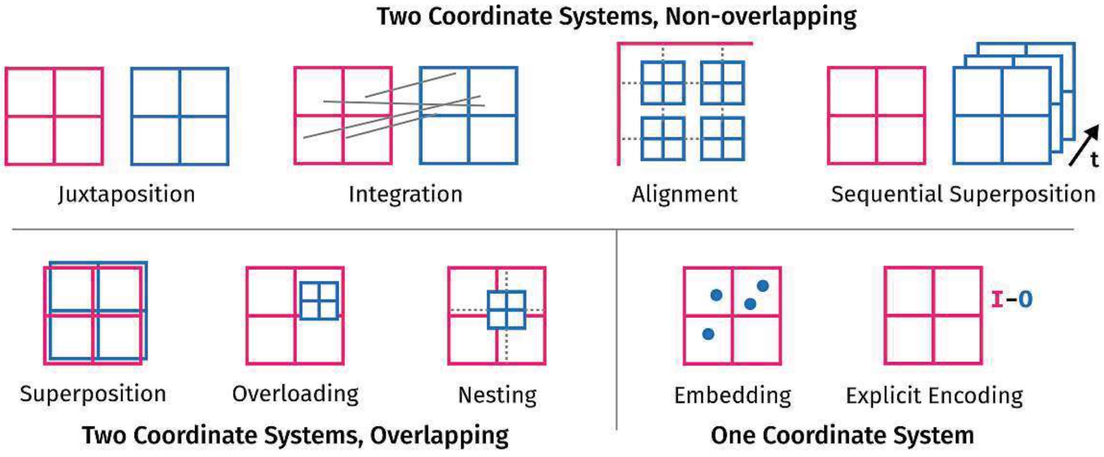


Figure 2.5: Input/Output Visualization sub-themes.

Juxtaposition. Juxtaposition refers to separate input and output visualizations, which are shown side-by-side, and any layout to do so is possible [JE12]. It allows specialized visualizations of the respective data type, e.g., a parallel coordinates plot for the A parameter and a gallery of the resulting S 3D models [AE20], or 2D embeddings of A parameters and AT time series [Orb+19]. Involved views are often conceptually linked through the Gestalt principles of common fate (when the analyst manipulates one view, the other changes immediately as well), or similarity (selected data cases highlighted in the same fashion everywhere). Because respective visualizations can be positioned anywhere and little shared visual cues are necessary, this strategy is flexible and can be applied to any data type combination. An example is by Zaman et al. [Zam+15], who propose a user interface for a geometry generator, i.e., the A parameter is a graph of parameterized drawing operations, and the S output is a vector image. Juxtaposing the graph editor and the output allows specialized visualizations for both. The desired vector image (*optimization* task) is created via indirect manipulation.

Superposition. Input and output visualizations are overlaid onto each other with Superposition: They occupy the same display area and share their coordinate system. While this allows detailed comparisons, the disadvantage of this strategy is that it only works with visual marks of the same domain, e.g., lines depicting time series in the same interval or trajectories referencing the same geographical space. An example is found in brachytherapy, where doctors place radiation seeds, which are injected into the patient's body to control tumors, on a matrix grid. By superpositioning seed amount and location (AS parameter), organs at risk (static S input), and radiation dose (AS output), doctors can *optimize* radiation dose.

Embedding. We refer to Embedding in the sense of "making an integral part of something." There is only one visualization and one coordinate system. Input and output are combined into the same visualization via mapping to visual channels. Hence, Embedding may technically be considered not a composition of two visualizations but rather the combination into one. Examples include scatterplots that show a parameter on one axis and a (possibly derived) output on the other [FMH16], parameters and outputs as axes in a parallel coordinates plot [Ste+13], color-coding output quality in a tilemap of two parameters [Ami+10] or on 3D shapes [Dor+15]. Embedding can support, e.g., *sensitivity* analysis when parameter and output are combined into visualizations that are suited to this type, e.g., scatterplots [Mat+18], parallel coordinates plots [Ste+13], or a combination of the latter with cobweb charts [Rai+14]. *Uncertainty* analysis can be carried out when multiple outputs are aggregated prior to Embedding. In the context of flood simulations, this was useful to visualize, e.g., the highest water level associated with any TS parameter setting (such as breach location) at any time step. From the embedded visualization, it can be seen which areas are flooded or not (colored or gray) and how badly (green-red colormap).

Alignment. Alignment refers to situations where inputs and outputs are visualized in separate visualizations. Hence their visualizations' coordinate systems are separate and do not overlap. In contrast to Juxtaposition and Integration, the visualizations cannot be rearranged at will. Examples of Alignment include spreadsheet-like visualizations (data for a row is horizontally aligned) or grid-like visualizations. Visualizations in the Alignment theme have similarities to pixel-oriented visualizations [Kei00] in that the individual visualizations can be, but are not necessarily, quite simple. The image that emerges by aligning many of those visualizations is more than the sum of its parts. We found Alignment to support diverse parameter space analysis tasks. E.g., when temporal outputs are sorted vertically by parameter settings, dependencies and correlations between parameter settings and output can be highlighted (*sensitivity analysis*). Of course, the exact sorting order must be flexible and changeable by the analyst.

Sequential Superposition. With Sequential Superposition, input and output visualizations have separate coordinate systems. They do not occupy the same display area, but the output visualization shows a single output that is rapidly exchanged over time after user interaction in the input visualization. While this theme could be seen as Juxtaposition, we argue that the high level of interactivity makes this approach qualitatively different. The user controls the emerging movie, enabling trial & error, probing, and "what if" analysis. In other words, by quickly experimenting with varying parameter settings and observing the model output, vPSA becomes possible. The controls are very often juxtaposed sliders, but more sophisticated visualizations are possible [Ume+14; Sch+18]. Sequential Superposition enabled mainly *optimization* and *sensitivity* tasks. Rapid exploration of the output space allows for quickly finding relevant parameter subspaces, which can be further refined. On the other hand, the influence on the output can be determined by varying one parameter and observing the output while keeping other parameters fixed. He et al. [He+20] developed a surrogate model for a computationally expensive ocean simulation by training a neural network to produce the desired visualization image directly. Analysts can freely change simulation, visual mapping, and view parameters on the left while the respective volume visualization is shown on the right. Another example in the same fashion, but without sliders, can be seen in the work by Umetani et al. [Ume+14], where a direct manipulation wing design interface is used instead.

Overloading. With Overloading, input and output visualizations overlap in the display area, but their coordinate systems differ. The position of the overlaid coordinate system is irrelevant, i.e., positions, distances, and sizes in one visualization do not directly translate to the other. An example is overlaying glyphs [RSG21]. While the space depicted in the overlaid graphics is the same as in the selected region of interest underneath, the offset and repetition make the approach different from Superposition. Malik et al. [MHG10], who show detected edges in scanned images (AS output), obtained various scanning configurations (A parameter). Seeing multiple of those in the same view enables both *optimization* when the analyst can pick the setting with the "best" edges, and *sensitivity*, as the analyst can investigate the impact of a few settings of one parameter on the detected edges in the selected region of interest.

Integration. Integration refers to Juxtaposition, i.e., separate non-overlapping input and output visualizations, but with explicit links between marks of the two visualizations [JE12]. Only Weissenböck et al. [Wei+16] and Yumer et al. [Yum+15] used this approach. In the former case, a trapezoid connects the respective A parameter and A derived feature ranges of histograms. Thus the trapezoidal annotation's shape hints at the *sensitivity* of the parameter. The integrating links connect S outputs to a point in the AS parameter space of the latter example, thus enabling *partitioning.*

Explicit Encoding. Explicit Encoding refers to only one coordinate system and visualization showing the difference between inputs and outputs with the Explicit Encoding idiom [Gle+11]. As specialized comparison visualizations were not that common in the papers we surveyed, this category also remains somewhat small. Explicit Encoding was mostly used with time series processing, highlighting where original (input) and output time series differ. In that context, the idiom usually supports an *optimization* task.

Nesting. Nesting means that input and output have separate visualizations and coordinate systems, they overlap in the display area, and the positioning of the overlaid coordinate system matters. The overlaid coordinate systems are nested into the marks of the "host" visualization. Hence, like Overloading, but position matters. Like Embedding, but marks are complete visualizations with their own coordinate system. Like Alignment, but there is a proper host visualization and not only imagined coordinate axes. Working with time series segmentations, Eichner et al. [EST20] added small correlation matrices into the marks of a visualization depicting different A● parameter ranges. In doing so, it becomes visible which ranges of a given parameter influence which features in the output, e.g., the number of segments with a particular label (*sensitivity* analysis).

Data Case Organization

Many systems work with multiple parameter/output pairs with temporal/spatial characteristics. A clear challenge to effective data analysis is the amount and complexity of the involved data. Hence, vPSA systems use varying strategies to reduce the amount and complexity of the data the analyst has to reason about. We found five strategies to achieve that (Figure 2.6), which, considering they resemble buildings blocks of an SQL `SELECT` statement, can be seen as basic querying operations. Their outcome may be visualized directly, or combined with each other to arrive at sophisticated concepts. E.g., we could obtain the accuracy of a model in a given spatial region of the output by i) filtering reference and output data to the spatial region (focusing), ii) computing the difference between reference and outputs (derivation), iii) computing the average of differences (aggregation). If this process is repeated for multiple regions, regions may be ranked (sorting) or clustered (grouping) by accuracy, thus supporting, e.g., *uncertainty* analysis. Other important scalars obtained by combining these operations are sensitivity indices, of which several [Ham94; Bor07; GI12] exist.

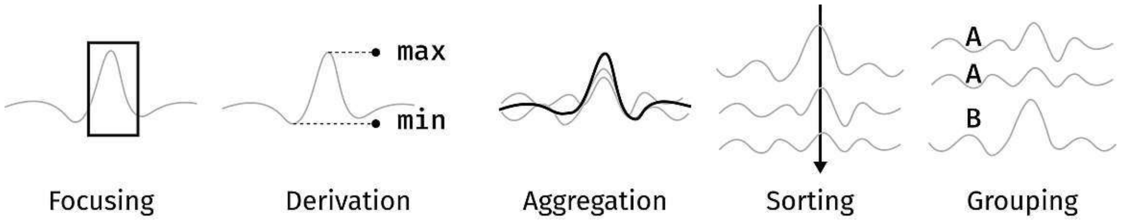


Figure 2.6: Themes for Data Case Organization.

Focusing. This theme collects interactions where the analyst focuses on a subset of data cases through *selection*/brushing (item-based) or *filtering* (attribute-based) or on a region/interval of interest through *navigation* in time or space. In other words, they decide to either look at fewer data cases or less information about a single data case (or both). By selection, individual data cases are marked as interesting. When relevant abstract attribute ranges are defined, it is referred to as filtering or attribute-based selection. Finally, space and time often need to be navigated independently of attribute values. Overview+detail visualizations [CKB09] can be used to maintain the broader context of the current focal region. Focusing on subsets of data cases or time/space is, on the one hand, necessary because display resolution and size are limited. On the other hand, a typical parameter space analysis process requires Focusing interactions. Input/output visualizations display parameters and outputs while highlighting relations relevant to the required parameter space analysis task, e.g., optimization or sensitivity analysis. To go from such *findings* to *insights* and *knowledge* [Sac+14], analysts have to, e.g., inspect relevant data in more detail or find related data cases. **Selection** is often performed by clicking on a data case in a specific visualization, which could, e.g., be a ranking [Was+14; Sor+16] or a time-varying vector field [Sag+17]. Selecting multiple data cases can be achieved by grouping them first and then allowing selection on the group representatives [BM10; Beh+14; FMH16], or by classical multiple selection tools, like a lasso [Was+10]. In systems with multiple linked views, this functionality is provided by brushing and linking. The inverse operation to selection is available in some works, where the user can exclude data cases from the analysis [OBJ16; Yañ+17; Swe+20]. Picking out individual data cases is cumbersome or infeasible when there are many. In such a case, a solution is to define a **filter** on their attributes. This approach is ubiquitous with systems that employ multiple linked views. An often-used example [Mat+10; Mat+13; Mat+14; Mat+17; Cib+17] of those is ComVis [Mat+08], which allows flexible brushing and linking in any view. Such systems allow analysts to filter in either parameter or output space and see the effect on the other. Parallel coordinates and related visualizations are especially common for this task [Ste+13; Cof+13; Beh+14; Dor+15; Orb+19; RPI19; AE20], possibly after feature derivation, but so are histograms and scatterplots. In a multiple-linked view system, InfoVis can be combined with spatial/temporal data. E.g., Ribičić et al. [Rib+13] use them to present derived features from spatio-temporal flood simulations. After the analyst selects data cases by brushing, related frames from multiple simulations are highlighted in a World Lines view [Was+10]. Analysts are provided sculpturing-inspired tools that allow them to filter 3D models based on spatial features in the DreamLens system [Mat+18]. E.g., the "chisel" tool defines a line in 3D and excludes any mesh that intersects that line. With temporal data, it is natural that analysts focus on a subset of the **time axis** because temporal data may span a long interval or have high resolution. This task is often solved by zooming into a smaller contiguous interval [Ber+19]. When there is additionally a spatial dimension in the data, it may be possible to either look at a summary of all temporal data in space (and vice versa) or to inspect single time steps in more detail [Bis+17]. The latter can be simplified by segmenting the time series and showing representatives [BM10; Bry+15]. We can look at the dimensionality of the part of interest to further categorize focusing in **space** beyond geometric view transformations such as pan/zoom or rotation. There are points, lines, surfaces, areas, and volumes. Points of interest occur, e.g., in particle simulations [GT16; Sag+17], where analysts may place seed points for particles and inspect their trajectories, but also in lighting design, where designers place glare probes in a room [Wal+20]. Schultz et al. [SK13] filter vertices of a 3D mesh by any existing or derived scalar value at a vertex by selecting thresholds in a density plot. Areas of interest, of course, naturally appear with two-dimensional spatial data. E.g., in image segmentation, Pretorius et al. [PZR15] allow to brush a subset of reference images so that analysts may focus on known problematic regions. Areas in 3D are surfaces and classified into usage types (e.g., work, leisure) in the context of lighting design [Sor+16; Wal+20] to verify legally prescribed light conditions. For Hazarika et al. [Haz+20], the space is a circle (an idealized yeast cell), and hence the interesting part is a line around it. Analysts may select a portion of that circle by brushing and querying for parameter settings that maximize/minimize the yeast simulation response there. Axis-aligned cubes of interest are used by Amirkhanov et al. [Ami+10] to mark features in a 3D scan.

Derivation. We refer to Derivation when new, simpler information is generated from a single data case. Usually, this data case is the output and we call the result a feature. We classify information that does not pertain to a single element but a population thereof (e.g., central elements, distributions) as Aggregation. Derived features are often scalars that quantify something of interest, such as how well an output matches a "ground truth" reference. Derived features may also preserve the spatial/temporal dimension. E.g., when boundaries of homogeneous regions in an image are of interest, those might be found with an edge detection algorithm. We distinguish between the type of the derived data (abstract, time, space) and what it measures (output quality or output feature). Derived features in the **Scalars Quantifying Output Features** category quantify domain-specific features in the output and produce one or more scalar values (A). These features are various. From the visual appearance of 3D models (S output) [Mat+18] to how far sandbags (ATS parameter) were swept by a flood (TS output) from their initial position [Rib+13]. Energy use can be derived from a building design (S output) [AE20], and the amount or length of labels from a time series segmentation (Figure 2.7). Well-known summary statistics are also used, like minimum/maximum value of a time series [Mat+10; Mat+14]. The other group of scalars (**Scalars Quantifying Output Quality**) quantifies the output quality. If no inherent quality metric exists, e.g., the number of intersecting triangles of a 3D mesh, outputs can be compared to a reference ("ground truth"). The latter can come, e.g., from human experts [Tor+11], from actual physical measurements, like the arrival time and speed on earth of a coronal mass ejection [Boc+15], or from government regulations, like lighting conditions in a work environment [Wal+20]. The former group of scalars depends on the application domain, and proper derivation functions have been identified for image segmentations [FMH16], porosity analysis in materials [Wei+16], or 3D meshes [Beh+14]. When the aforementioned scalars are derived per time step of a parameter/output with temporal characteristics, one derives AT data (**To Time+Abstract Data**). They fall into the same two categories, i.e., they quantify either output quality or characteristics. Uncertainty in time was quantified by Biswas et al. [Bis+17] to show how a spatio-temporal model is influenced by grid size (a model parameter) and by Bernard et al. [Ber+19] to highlight which parts of a multivariate time series change significantly by a preprocessing algorithm. Similarly, Röhlig et al. [Röh+15] and Luboschik et al. [Lub+12] show the fit to a reference over time. Many features that indicate spatial and group behavior in the context of chaotic movement patterns in biology were plotted over time in another work by Luboschik et al. [Lub+15]. Derived features may also preserve the spatial dimension, producing AS data (**To Space+Abstract Data**). Malik et al. [MHG10] used edge detection to highlight differences between many 3D X-ray computed tomography images. Sagristà et al. [SJS20] detect ridges in a finite-time Lyapunov exponent field. Obermaier et al. [OBJ16] derive metrics about temporal and spatial trend characteristics.

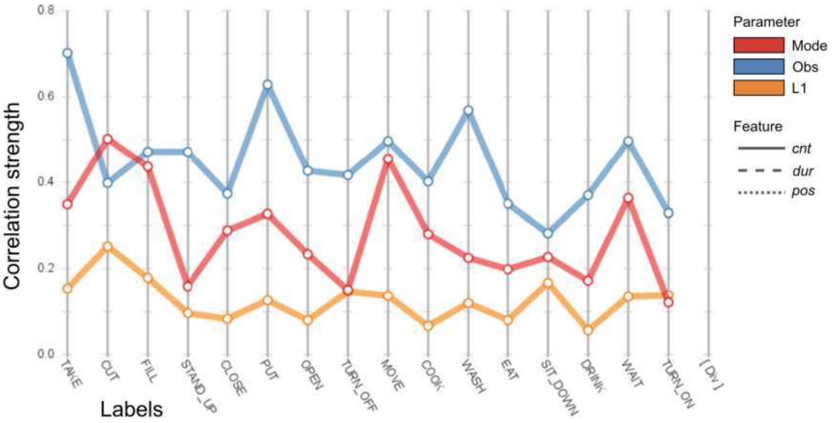


Figure 2.7: Example for Derivation. Parallel Coordinates Plot showing correlations (Y position) between a A parameter (line) and the number of segments with a given label (axes), a derived feature from the AT output of a time series segmentation model. It is visible that the `Obs` parameter influences the number of labeled segments most (*sensitivity*). Image: [EST20, Fig. 4] License: CC-BY

Aggregation. Multiple data cases are aggregated in one way or another to reveal information related to statistical distributions, e.g., central items, outliers, or frequency of items. Data characteristics of data cases are retained, i.e., aggregating many time series yields a time series, and aggregating scalars yields a scalar. Classic examples for A data are summary statistics, like mean or standard deviation, histograms, box plots. Distributions in time and/or space are also often of interest. Naturally, as this section is about summarizing spatial and temporal data, overlap with approaches used in ensemble visualization [Wan+19] is expected. Focusing on the common behavior of multiple elements while preserving data characteristics sets this sub-theme apart from Derivation. We distinguish between characteristics of aggregated data. **Abstract data** often arises as part of feature derivation. Matković et al. [Mat+10] summarize many time series (AT) by showing a histogram of a A user-defined feature (minimum, average, or maximum value). Sagristà et al. [SJS20] summarize a finite-time Lyapunov exponent (FTLE) field by counting ridges (A feature), which are then aggregated by summary statistics. Unger et al. [Ung+12] use the average goodness-of-fit of a geoscientific simulation model to uncertain ground truth to validate the model. In **Space**, Landesberger et al. [vLan+13] show a 2D distribution plot of 3D meshes (S static input) so that the analyst may choose between a gaussian and non-gaussian distribution (a parameter of the 3D segmentation algorithm). To summarize stochastic 3D packings of molecules (S), Schwarzl et al. [Sch+19] use a density plot from an orthogonal direction. In **Space+Abstract**, Beham et al. [Beh+14], as well as Fröhler et al. [FMH16], aggregate multiple image segmentations (AS output) to a single visualization image by highlighting where segmentations disagree. Cibulski et al. [Cib+17] summarize a set of surfaces (AS output) with 3D boxplots. Raidou et al. [Rai+16] show uncertain regions of tumor treatment by showing the variability of recommended radiation dosage from multiple parametrizations of a tumor control probability model (AS output). Malik et al. [MHG10] perform edge detection on scanned images, yielding a AS feature, then align histograms to the side of a scan that shows how many images have an edge in that row/column but not others. In **Time+Abstract**, Ribés et al. [RPI19] find quantile time series by density analysis in principal component space of many AT simulation outputs. Bernard et al. [Ber+16] highlight the uncertain parts of multiple time series segmentations (AT) by showing the probability of class labels over time with line graphs. In **Space+Time**, Rojo et al. [RGG18] employ density volumes and isosurfaces to show the distribution of particle trajectories (TS output) in time and space By separating the density volumes further using color, the influence of the particle size (A parameter) becomes visible (*partitioning*, *sensitivity* tasks). Sagristà et al. [Sag+17] use phase-space FTLE maps to show the variance of particle trajectories (TS output) depending on the initial position or initial velocity. To analyze many flood simulations (TS output), Ribičić et al. [Rib+13] propose an aggregation pipeline that involves extraction, grouping, aggregation, and embedding.

Sorting. Another approach to reducing the amount of data cases is to rank them according to some logic. As position in space is the most accurate visual variable, sorting parameters/outputs allows organizing complex data quickly and aids understanding as the analyst only needs to inspect the top few results. When sorted data cases are presented as visual objects, e.g., glyphs, complex patterns may become apparent. We distinguish sorting by scalars (one-dimensional A data) and complex (i.e., everything else) attributes.

Producing a 1D ordering of objects is known as seriation [Lii10]. The simplest case is a 1D seriation of a **scalar**, which we can sort. Arrangements along a single dimension include lists, rankings, and so on, but also spreadsheets sorted by one column. In the context of flood simulations, Waser et al. used this technique to sort parallel universes based on a derived AT simulation state [Was+10]. Approaches making use of design galleries also often allow sorting those based on a user-defined criterion, like the value of a derived feature [Dan+15; Mat+18]. In interactive optimization, Liu et al. [Liu+21] recommend sorting obtained solutions by the objective function's value. Sorting by more **complex** data than scalars was also often used to organize data cases. E.g., a user-defined weighted sum of a multidimensional A attribute produces a scalar again. Waser et al.

ranked protection plans (ATS parameter) by a weighted sum of cost, protection, and construction time [Was+14]. Sorger et al. [Sor+16] ranked lighted scenes (AS output) by how close lighting conditions are to legally prescribed values on surfaces of interest. With spreadsheet-based visualizations, parameters and outputs (or derived features) are represented as columns and data cases as rows. Users may then sort all rows or subsets by one or more columns. As temporal/spatial data make up one column, different similarity metrics and sorting algorithms, e.g., optimal leaf ordering [HHB08], can be used to obtain an ordering. Spreadsheet-based approaches have been mostly used with AT outputs and A parameters [Lub+12; Lub+14; Röh+15; Ber+18; EST20], but also with derived features from spatial data [PZR15], or spatio-temporal data [Lub+15].

Grouping. Separating data cases into coherent groups is another way to organize a large body of data. This task can be achieved automatically through clustering algorithms if similarity information of data cases is available. It may also make sense to let the user decide on the particular groups, which are then formed based on the current analysis goal. E.g., authors often used Grouping to *partition* the output space and, by visualizing parameter settings per cluster, showing their *sensitivity*. We distinguish further by which characteristics data cases are grouped.

A hierarchy of A (**Abstract**) parameter settings is used in Paramorama [Pre+11], allowing analysts to quickly step through relevant subspaces (subtrees). The parameter space of Poco et al. is a binary vector (A). Hence, analysts may group the AT outputs by whether a Boolean parameter is set or not [Poc+14]. Different groups of data cases appear automatically in the work by Bao et al. [Bao+13] as the dimensionality of underlying parameter subspaces makes it necessary to present data cases separately. When A features are derived from outputs, grouping by abstract data may also involve outputs [OBJ16; Bis+17]. Notable is Schulz et al. [Sch+18] who, in parametric engineering design, achieved a partitioning of solutions in performance space (stress, mass, heat) where groups contain Pareto-optimal designs. Abuzuraiq & Erhan [AE20] use hierarchical clustering both on the A parameter space and on the resulting 3D shape (**Space**, S) of the building in the context of generative architecture. Similarly, Beham et al. [Beh+14] group 3D meshes (S output of a generative model) and display emerging clusters in a A parameter space visualization (PCP). Fröhler et al. [FMH16] group image segmentations (AS output) hierarchically and visualize their disagreement. Clusters may be selected, which updates linked A parameter visualizations. In parametric design, Woodbury et al. [Woo+17] allows analysts to group S data cases (3D models) into collections, which may be automatically expanded by combining A parameter settings. Ribičić et al. [Rib+13], in their proposed pipeline to visualize data from multiple TS flood simulations (**Space+Time**), group data in a domain-specific way and distinguish between objects (buildings), fields (water) and instances (sandbags). Information about group members is subsequently aggregated and visualized, e.g., by Embedding. Working in visual effects design, Bruckner et al. [BM10] group TS outputs into coherent temporal segments based on frame similarity. The segments are then depicted in a timeline.

(Surrogate) Model Tuning

In some vPSA systems found in the literature, it is possible to interact with the (actual or surrogate) model itself. In some applications, this is necessary because building a suitable model is part of the parameter *optimization* task. An example is pipelines, a common concept in image [Wei+16] or time series processing [Ber+19]. The analyst needs to find appropriate parameter settings and choose the required steps (e.g., outlier removal or smoothing), their order, and which algorithm to use. We can distinguish the operation performed on the model: Editing and inspecting. The former alters the model, while the latter collects and presents its internal information.

Editing. Editing refers to the previous example of building a pipeline or a surrogate model inside the system as part of the exploration process. The latter was done in two works by Matković et al. [Mat+14; Mat+17], where the analyst defines a regression model on a data subset. This model was then further used to estimate and sample a A parameter subspace [Mat+14]. As for pipelines, we found examples for image processing [vLan+13; Wei+16] and time series processing [Ber+12; Ber+19]. Matković et al. [Mat+10], in the context of an electronic unit injector simulation, allow the analyst to build a schematic model of the individual involved components. Bryan et al. [Bry+15] support analysts in defining a suitable emulator for a complex simulation with ATS output. Finally, Dang et al. [Dan+15] assist an analyst in defining a probability density function for a shape grammar, in which, after user interactions, they automatically update probabilities of individual rules and the set of rules themselves.

Inspecting. Inspecting, on the other hand, exposes the internals of the surrogate model to the user. While inspecting model internals can be required to build a proper surrogate, it was sometimes also used on its own. Matković et al. [Mat+17] show regression coefficients of a user-developed surrogate model to quantify relationships between A parameters and A features derived from AT output. Hazarika et al. [Haz+20] visualize weight matrices of the neural network surrogate model to validate that it learned domain-aligned logic.

Provenance

The term "provenance" [Xu+20] in the visualization literature roughly refers to tracking either how data was generated/modified or how the user interacted with the system. Capturing and using user interactions is useful, e.g., for an analyst to recall the analysis process. Within vPSA, we can more specifically distinguish between the following approaches:

- analysts mark single data cases that appear in a dedicated list (bookmarks) [Tor+11; Dor+15; OAH15; FMH16; Swe+20; Day+20];

- systems that capture every intermediate result [Bög+13; Bög+14; Zam+15; Bor+17; Wal+20; Liu+21];
- load/save functionality to recover past work [WDR11; Ber+12];
- giving names to individual data cases [Yañ+17].

Thus, this theme refers mainly to accessing relevant data cases later. While other reasons for collecting and using provenance data can be found in the context of vPSA, they were rather few. In only one instance [SK13] was interaction history not used for bookmarking but for replicating useful parameter settings on other datasets. Data provenance was not used at all, which is maybe not surprising given that investigated data often come from simulations and their heritage is thus well known.

Open Challenges and Future Work

We contextualize our directions on future work for the field with those by Sedlmair et al. [Sed+14], who looked at vPSA from a more system-centric perspective. Their identified research gaps pertained to data acquisition, data analysis, and cognition. Data acquisition is about the ability to obtain interesting parameter/output pairs within the vPSA system. The data analysis gap refers to "opening the black box" specifically for the derivation/prediction steps in their data flow model. The cognition gap is about how to facilitate the search for and navigation between A ⬢ parameters. Other mentioned future work topics were scalability, guidance, provenance, collaboration, and evaluation.

Regarding guidance, Ceneda et al. [Cen+17; Cen+18; CGM19] defined it as a computer-assisted process that resolves a knowledge gap of the analyst in an interactive VA session. It received lots of attention in recent years [Col+18; Spe+20]. Among other facets of guidance the authors introduced the guidance *degree*, which is ordered from weak to strong as *orienting*, *directing* and *prescribing* guidance. The knowledge gap in vPSA usually relates to parameters (*data* domain), i.e., which settings cause the most certain/optimal/sensitive/outlying outputs, so it should not be surprising that many of our themes are associated with certain characteristics of guidance and vice versa. A few examples: *Orienting* guidance often involves visual clues. Hence, it can be found in our Input/Output Visualization and Data Case Organization themes. The Manual/Constrained sub-theme is related to *directing* guidance when the system presents options to choose from and *prescribing* guidance when it automatically adapts solutions or prohibits selection outside of certain parameter subspaces. The domain of the knowledge gap is mostly the *data* (parameters/output pairs). Some works [Koy+17; KGS19; KSG20], which break the parameter selection problem down to simpler sub-tasks, can also be seen to provide a solution in the *tasks* domain. The guidance input is usually the *data*, but examples exist for others, e.g., *domain knowledge* [Was+14] or *user knowledge* [Pre+11]. Our increased understanding of guidance since the survey by Sedlmair et al. [Sed+14] shows us that it has been there since the beginning [JM00; Tor+11], albeit sometimes in subtle ways. Thus, the question for the future is less about how to provide guidance for vPSA, as we have provided many examples in this survey. Rather, it is about fine-tuning the guidance process and making it more flexible, e.g., combining multiple guidance inputs, timing guidance correctly [Cen+21], switching between guidance degrees [Pér+22] and means to show the answer, and so on.

However, in our view, other topics (scalability, provenance, collaboration, evaluation) are for the most part still current, even though our perspective is different, as we focus on the user interface. We will list our topics for future work in vPSA first and afterwards relate them to those by Sedlmair et al. [Sed+14].

Parameter Space Tasks in Time and Space. We collected 101 papers supporting various vPSA tasks for models where *either or both* parameters and outputs have a temporal/spatial reference. A complete table of papers, including referenced space/time characteristics, can be found in the supplemental material of the published paper [PBM23b]. Slicing this dataset in different ways, we find chunks smaller than others and thus indicative of gaps in the literature. Table 2.1 shows a contingency table of parameter space analysis tasks and data characteristics of the parameters. The row margins show that most papers discuss A (63/101) or S parameters (27). At the same time, we found only a few papers for the remaining space, time, and abstract combinations. Naturally, some parameter space tasks remain unsupported for these combinations (9 cells highlighted in red). For 20 other combinations, there are only a few examples in the literature. We highlight the relevant cells of Table 2.1 with three or fewer examples in light orange. Hence, future work should investigate the tasks uncertainty analysis, partitioning, outliers, and fitting for AT , AS , and ATS parameters. More generally, vPSA systems for other than A or S parameters seem rare enough to warrant future explorations.

Data Volume. The larger collections of data we saw were about a few thousand parameter settings and relatively small associated data, e.g., 3D models of a monitor stand. Our survey gives relatively few answers how to enable vPSA for data-intensive models, where the output of a single run is on the order of gigabytes of data. He et al. [He+20] suggest a possible approach, in which the surrogate model skips the output and learns the visualization image directly. Producing partial results during model execution (Progressive Visual Analytics [Ang+18]) might be another viable strategy to build interactive visualizations for data-intensive models.

Data Variety. Most of the models in our survey take one or a few parameters and produce a single output. We did not see data structures such as graphs, sets, hierarchies, or even multiple outputs a lot. This may be due to simplifications introduced by visualization designers or an actual property of many models. In any case it is an open question how to enable vPSA for such inputs/outputs.

Data Quality, Data Provenance, and Uncertainty. Many models take complex input parameters, such as time series. These input parameters may need to fulfill some

	PSA Task	Optimization	Sensitivity	Uncertainty	Partitioning	Outliers	Fitting
Parameter	*Total*	*68*	*59*	*25*	*25*	*19*	*13*
A	*63*	37	45	22	17	14	11
S	*27*	18	12	3	8	5	2
TS	*6*	2	3	1	3	1	2
AS	*6*	6	2	0	2	0	0
T	*5*	4	2	3	2	2	1
ATS	*5*	4	3	1	0	1	0
AT	*2*	1	1	0	0	0	0

Table 2.1: Contingency table of parameter space tasks [Sed+14] (columns) and parameter type (rows), where A = abstract, S = space, and T = time. Red color highlights task/parameter combinations that were not tackled by any paper in our survey. Light orange highlights combinations tackled by 1–3 papers. Note that a VA system may support multiple tasks and a model may require multiple parameters.

properties, e.g., the time series being free of holes (no missing values). It may also be the case that the original input did not have these properties and was preprocessed somehow to this end. Few works consider the uncertainty introduced by such preprocessing steps, or uncertainty that may have existed in the input from the beginning. This is an important future research direction towards reliable and trustworthy insights with vPSA.

Analytic Provenance. The Provenance theme in our survey is about quickly accessing individual data cases, as that is the part of provenance-related interactions that was mostly exposed to users. Xu et al. [Xu+20] reviewed provenance in visualization and identified several ends to which provenance data was used. We saw in our survey approaches for *model steering* [Mar13; Koy+17; KGS19; KSG20] and *replication* [SK13], but others, like *adaptive systems* or *understanding user* are less explored. In which ways analytic provenance can be leveraged for vPSA is, therefore, an interesting research direction for the future.

Composite Visualizations. We classified visualizations that show model inputs and outputs. The majority used Juxtaposition, which speaks to the flexibility of the approach. Some composition approaches were used seldomly or rarely, e.g., Integration, Nesting, Overloading. This suggests that the design space of composite visualizations in vPSA is not fully explored yet and future work in this direction might uncover useful visualization idioms.

Data Organization Approaches. Sorting and Grouping are the least popular subthemes in that category. That is somewhat surprising because these two approaches are part of the basic organization activities we do in everyday life. E.g., when organizing a bookshelf, we often group by book owner and sort by author. While related tasks are different—quick retrieval (bookshelf) vs. pattern perception (parameter analysis)—vPSA by flexible grouping and sorting of data cases should be explored more, given how intuitive the two actions are.

Advanced Interaction Design. Woodbury & Mohiuddin [Woo+17; MW20] suggest that designers prefer to pursue multiple design alternatives in parallel and to quickly explore alternatives. We only found one system besides theirs that really allowed that [Zam+15], where users edited graphs of drawing operations for a 2D pattern. How vPSA users can work simultaneously on other complex models and how to quickly come up with suitable alternatives of complex parameters is another promising research direction for the future. In a biological simulation context [Haz+20] it was suggested that this interaction paradigm may be useful not only for designers. A so far not taken direction could be grammars, which encode rules how to construct complex objects from simpler parts [ARB07; Guo+14; Zha+20]. Additionally, most surveyed works employed the established WIMP paradigm (windows, icons, menus, pointers). Exploring vPSA with alternative paradigms, like in virtual reality [Bru+16], or input devices, such as tablets [Kaz+17], encompasses another direction for future research.

Collaborative Aspects. Most surveyed papers were intended for a single user working on one machine. Collaborative aspects were seldomly considered in the proposed systems. A part of Visdom [Kon+14] is dedicated to justifying decisions to avoid flood damages, e.g., where to put barriers, so that officials may explain those to the public. How people can work together in a vPSA setting, is still mostly untouched territory.

Opening the Black Box. Many papers in our survey saw their model as a black box and focused more on parameter/output relations instead of how the internals work. The many successful applications show that this approach works in general. It is especially advantageous, e.g., when intermediate steps inside the model are not important or not well understood by analysts. In other cases, it may lead to better outcomes or deeper insights into how the model works. Future work should determine when and if the additional effort of the "opening" process (e.g., in terms of visualization design) is warranted. A few papers we surveyed considered a pipeline of processing steps, which could be viewed as opening up a model. Aside from that, vPSA designers may draw inspiration on how to open black box models from a large body of research about using VA to interpret machine learning models [Cha+20].

Model Comparison. Most works investigate a single model. It is, however, not difficult to imagine that alternative models exist, e.g., different segmentation pipelines [Wei+16], models with different assumptions [Rai+14], or different formulations of the same physical reality [HG18]. In our survey we found only few works that focus on the specific task of model comparison, e.g., finding respective parameter subspaces that lead to comparable results. More research in this regard could help domain experts choose models based on other considerations than exactness of the output.

Supporting Larger Data Processing Pipelines. Most of the models in systems we surveyed deal with a single step of a more extensive data processing pipeline. Even, e.g., time series preprocessing, which is in itself a pipeline, is only at the beginning of a more holistic task. The larger pipeline also consists of several interdependent steps. Every step incurs choices regarding parameter settings or algorithms, influencing subsequent steps. Systems we found focused either on single pipeline steps and ignored the bigger picture or focused on the whole constructed pipeline and glossed over details. We believe the spectrum between the two extremes is worth exploring more.

Evaluation Practices. Ultimately, we are all interested in what part of our visualization designs worked and what did not, which is why we evaluate our designs. vPSA fits mainly in the "Visual Data Analysis and Reasoning" scenario by Lam et al. [Lam+12]. Proposed evaluation practices include case studies, interviews, or controlled experiments. All of these involve human participants. However, half of the surveyed papers where we could infer that information reported no human participants (median 0.5, mean 4.39, standard deviation 7.83). This number is to be taken with a grain of salt, as our survey includes papers from various journals and conferences. Interactive visualizations for vPSA were not always the main contribution of the paper. Nevertheless, it suggests a certain imbalance between how vPSA systems *should* be evaluated and how it is done in practice. Future work should put more emphasis on appropriate evaluation practices of suggested designs and approaches to strengthen the body of knowledge of our community.

Relation to Blind Source Separation

Referring to the themes presented in the previous sections, one can observe that, currently, BSS analysts identify parameter settings by a **Manual / Constrained** idiom with **Indirect Manipulation**. Navigation between parameter settings happens by **trial and error** [Sed+14]. There is no surrogate model and no systematic provenance.

It becomes apparent that BSS tackles several of the mentioned open challenges. Inputs/outputs appear in sets and analysts are interested both in latent dimensions as well as the unmixing matrix (**Data Variety**). BSS approaches considered in this thesis have temporal/spatial parameters to set, but the support of PSA tasks is unevenly distributed in the vPSA literature (**Parameter Space Tasks in Time and Space**). At the same time, some popular themes are unlikely to work for BSS. E.g., extensive random sampling, as it is often employed for multivariate parameter spaces (**Automatic / Unsupervised** theme) is not considered practical for BSS. BSS parameters have to make sense for the domain BSS is applied in, the dataset at hand, and for what one desires to find in the output. Another example is **Derivation**, with which complex outputs are often simplified. While latent dimensions are expected to show underlying processes, such as climate for SBSS in geochemical applications, there is usually no reference or ground truth to which it would make sense to compare them computationally.

2.2 Temporal and Spatial Data Visualization

Due to the focus of this thesis, which covers temporal and spatial, but not spatio-temporal BSS, and the volume of related literature, we consider temporal and spatial data visualization separately. For an overview of spatio-temporal data visualizations, see, e.g., [Bac+17].

2.2.1 Temporal Data

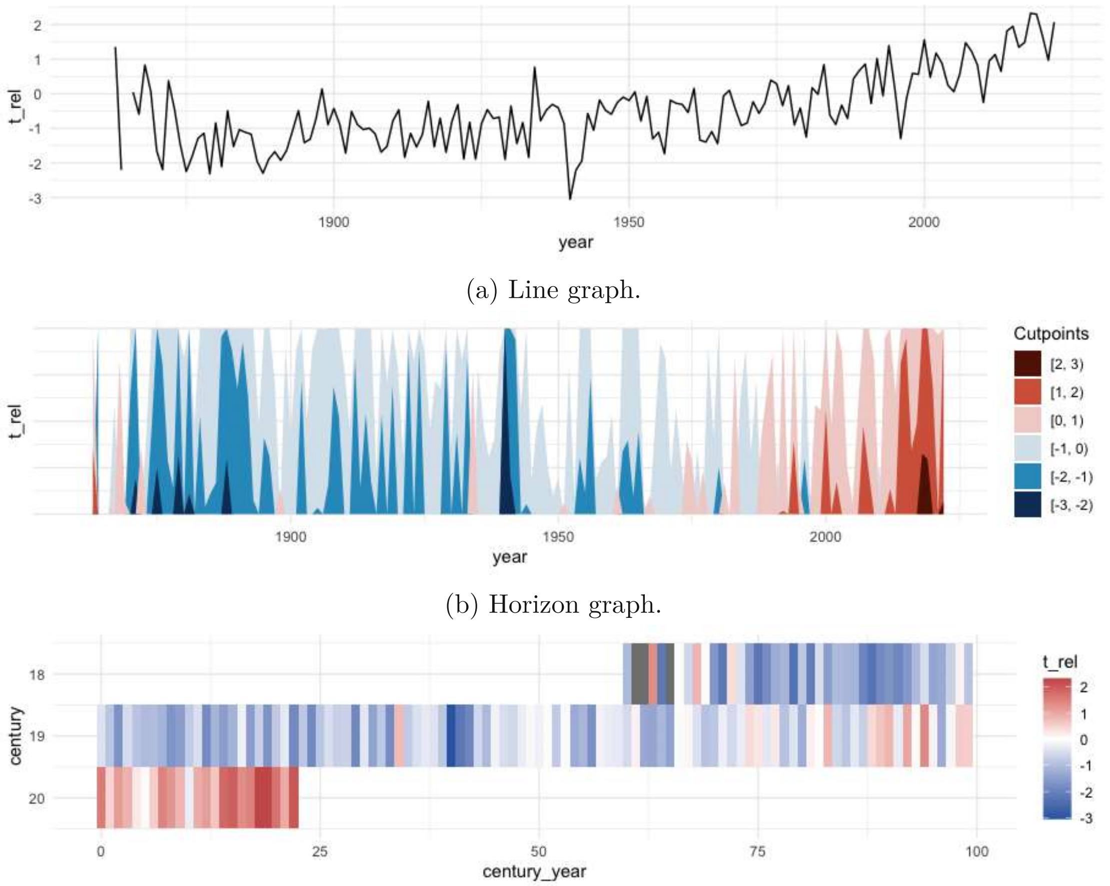
Time is a complex concept. Aigner et al., in their book about visualization of time-oriented data [Aig+11], describe various possible characteristics that a specific time model can have. There is the **scale** of the temporal domain, which can be *ordinal*, *discrete*, or *continuous*. Only the relative order between recorded events is known in an ordinal scale, but no information is available about how much time passed in between. If each point on the time scale is equidistant from its neighbors, we are looking at a discrete time scale. They are usually made from smallest time units (e.g., milliseconds) and it thus impossible to represent a finer granularity. Continuous scales, on the other hand, allow arbitrary precision. The **arrangement** of the time domain is often considered *linear*, i.e., proceeding steadily from the past to the future, but in some circumstances a *cyclic* arrangement of time (e.g., seasons) may prove more useful. Time **granularities** [Dyr+00] allow to group time into conceptual units, such as minutes or business days. In a calendar system, these are (mostly) hierarchically ordered on a lattice [Aig+11, Fig. 3.16]. Relationships between time **primitives** (instants, anchored, and unanchored intervals) across granularities can be peculiar. A time point on *day* granularity is indeterminate on lower granularities, such as *minutes*, as the particular minute is just not known. Equality or generally order relationships change, too, across granularities. For instance, December 31st and January 1st of the following year are always on *different* days, often in the *same* week, and always in *different* years. Further complicating the issue are irregularities betweeen physical time, i.e., the time that is constantly passing, and clock time, i.e., the current time on the clock. These differences regularly appear as leap years, leap seconds, or daylight-saving time. To account for these, the R package `lubridate` [GW11] for time calculus distinguishes between *periods*, which account for clock time changes independent of irregularities, *durations*, which track passing of physical time and are thus not always equal to periods, and *intervals*, which are defined by two instants and thus correspond to neither durations nor periods. Furthermore, time primitives may be subject to **uncertainty**. Common models for temporal uncertainty [Gsc+16] can be statistical, i.e., the uncertainty follows a statistical distribution (e.g., a normal distribution), or bounded, i.e., any value within bounds is possible and no information about its probability is available.

For the purpose of this thesis, we consider only visualizations that use a discrete time model without uncertainty. Additional data are associated to time instants in the form of vectors. Thus, both *univariate* time series are relevant (one visualization per variable) as well as *multivariate* time series (all variables in one visualization).

We picked three visualization idioms for univariate time series fitting to the mentioned constraints to discuss them in more detail (Figure 2.8). We chose them for their generality, i.e., the reader is likely to be familiar with at least one of them. Of course, specialized visualizations for more specific tasks exist [Aig+11], e.g., comparing values in all subintervals of the time series [Kei+06]. The most basic of our three candidates is a line graph (Figure 2.8a), where time is on the X axis and the value on Y, while values are connected by straight line segments. The slope of a segment thus encodes the delta between two subsequent measurements and the area between the line and 0 the sum of values. A variation of that visualization idiom is the horizon graph ([Sai+05], Figure 2.8b), which is especially suited to smaller sizes. The Y axis has a fixed height and values that exceed it are plotted again from the bottom (for positive values) in different color. It behaves like the spaceship of the 1979 Atari game "Asteroids", where space wraps around the screen and the ship reappears at the bottom when it left the screen on top. Negative values are often plotted reversed to save space, like pictured, in which case the Y axis encodes the absolute value and color shows the direction. Finally, we can show two discretized time granules in a tile map[1] (Figure 2.8c), where the mark's color (pictured) or size encodes the value. The famous "Warming Stripes" visualization can also be considered a tile map, although with just one granule (year) and tiles of extreme aspect ratio. While all three visualizations show the same data, they are suited to different tasks [Gog+19]: For instance, patterns in color bands (a continuous variant of a 1D tile map) are perceived differently when they are shifted in time or value, whereas line charts suffer less from this problem. Generally, while all three examples use a linear time domain, they could be plotted on a circular domain. Di Bartolomeo et al. [dBar+20] compared the effect of timeline shape (spiral, circle, horizontal and vertical line) on task performance. The authors suggest that circular time domains should be avoided in visualizations unless specifically requested.

When it comes to multiple variables, we can show one univariate visualization for each variable side by side (Juxtaposition [Gle+11]) or try adding more variables to each visualizations, like in a line chart with one line per variable (Superposition). More sophisticated approaches exist. LiveRAC [McL+08] is a visualization approach for many time series in the context of system administration where several metrics (e.g., CPU load, free memory, network traffic, etc.) are tracked for multiple hosts. Time series are arranged in a flexible grid and visualization idioms change between line charts, spark lines and colored blocks depending on the current cell size. Fuchs et al. [Fuc+13] conducted a comparative study where time series where shown as four glyphs designs. Line graph glyphs performed best when comparing two values, while star and clock glyphs were

[1] While a colored matrix is sometimes also referred to as *heatmap*, we avoid this terminology here and reserve a heatmap for filled areas of isocontours.



(a) Line graph.

(b) Horizon graph.

(c) Tile map.

Figure 2.8: A “Warming Stripes” dataset in different visualization idioms: Line graph (a), horizon graph (b), and tile map (c). Refer to Section 2.2.1 for a discussion. They show yearly deviations from the 1970–2000 mean of daily air temperature in Vienna (Hohe Warte station, Klima ID 105), 1860–2022. Data obtained from ZAMG.

best when comparing two time instants. In subsequent work, glyphs seem common when the task is *outlier detection*. Cao et al. [Cao+18] suggested the Z-Glyph to indicate multivariate outliers, where each glyph shows the whole time series. Suschnigg et al. [Sus+21] designed a glyph that encodes anomaly scores from several algorithms. In contrast, those glyphs represent a single time step.

The connected scatterplot is a technique to visualize relative changes over time in two time series. It works exactly like a scatterplot, except that dots are connected by lines in temporal order. Haroz et al. [HKF16] investigated the type of errors users make in comparison to a dual-axis line chart and found that the most common mistake was about the direction of time. Javed et al. [JME10] conducted a controlled experiment in which they compared superpositioned line charts, a braided graph (filled superpositioned line charts where segments are z-ordered by value), and small multiples of horizon graphs and filled line charts. The investigated tasks were finding time series with maximum value at an instant, finding the time series with the highest slope from start to end, and deciding whether time series A at instant i_A has a higher value than time series B at instant $i_B \neq i_A$. They found shared-space visualizations better for the first task, split-space better for the second, and small multiples of line charts to be a robust choice over all three tasks. Tominski et al. [TAS04] suggested the TimeWheel visualization, where time is its own axis in the center and the other variables are ordered circular around it. The visual encoding is similar to that of a Parallel Coordinates Plot (PCP), i.e., lines connect the time instant in the center and the value on one other axis. In contrast to a PCP, there are no polylines but straight lines for each pair of (time,variable) axes. While they have been picked up again later by Claessen and van Wijk [CV11] in larger work on flexible axes, no controlled task-based studies on their effectiveness were conducted so far. Gruendl et al. [Gru+16] integrated time into a PCP differently, where time passes on the Z axis, i.e., depth, in a perspective-distorted line chart between two PCP axes. ThermalPlot [Sti+16] visualizes many time series by positioning them in a scatterplot where the X axis encodes the current value and the Y axis the delta to some other point in time. Time series are thus naturally grouped (via position) into positive/nevative value/trend (Figure 2.9). Overview+detail visualizations [CKB09] support using the scatterplot.

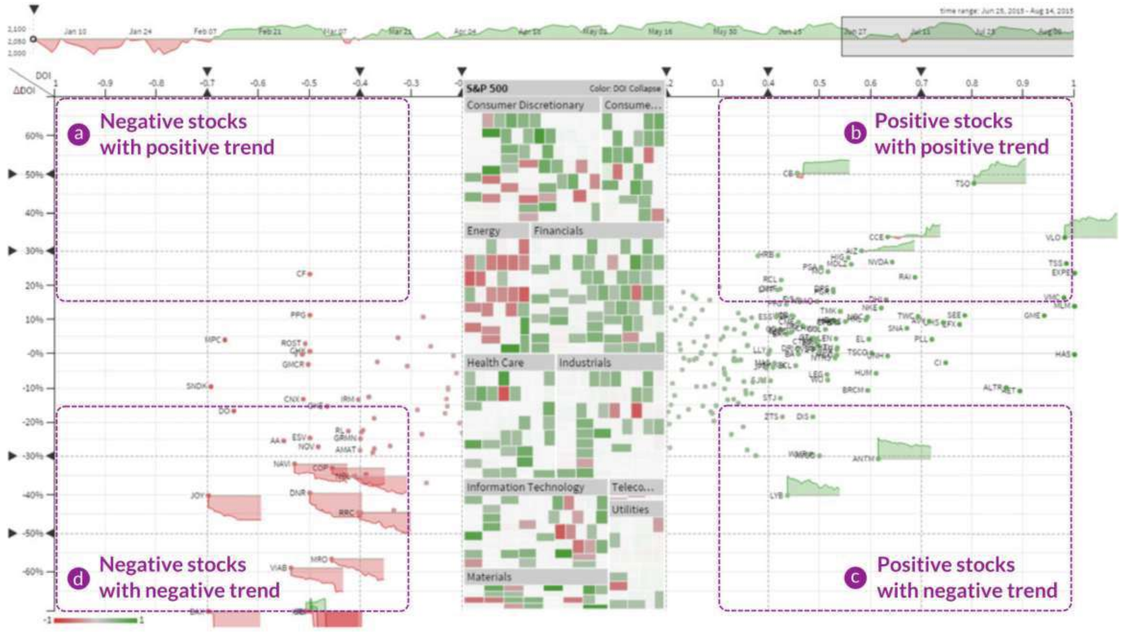


Figure 2.9: ThermalPlot. Stock prices of companies are shown as line charts and arranged by current value (X position) and delta to a reference (Y position). Image: [Sti+16, Fig. 10] © 2016 IEEE

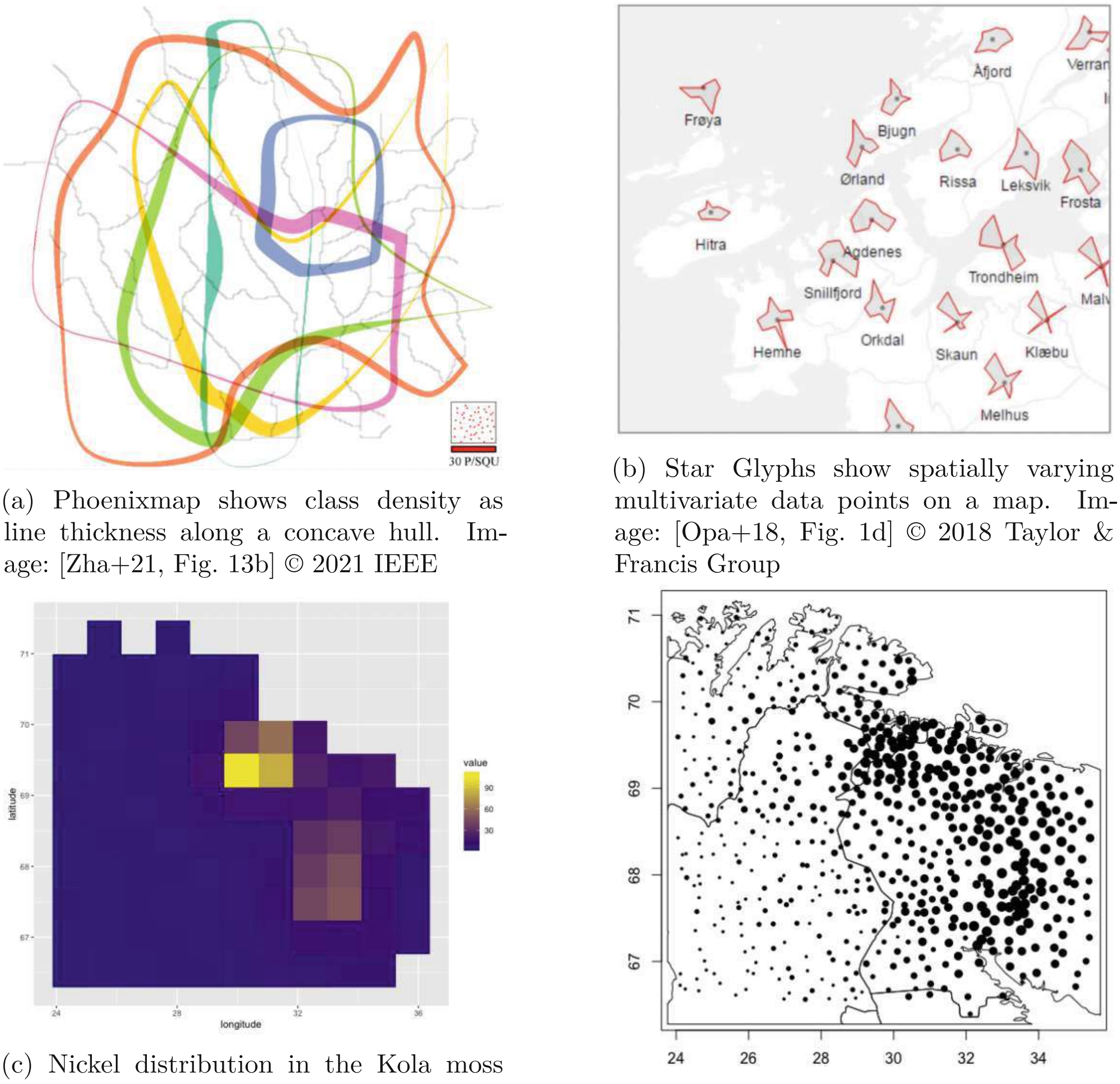
2.2.2 Spatial Data

Several spatial primitives (points, lines, areas, volumes) exist and they can appear in combination with many data types, such as networks or multivariate data. Within our research context, however, we only consider uni-/multivariate data associated with irregular (i.e., non-gridded) points in the plane. That means while, e.g., cartograms [NK16; Nus+18; TML18; Nic+22] or grid maps [MSS21] are well-known visualization idioms, we do not discuss them here as they usually depict areas instead of points. Fundamental considerations about thematic mapping and map design, such as projections and computational aspects, can be found in textbooks [Den96; Mue14; And+20; KO21].

Geovisual Analytics. VA specifically concerned with the analysis of geographic data is known in the literature as geovisual analytics or geovisualization. Andrienko et al. [And+07] proposed a research agenda in 2007. Among the identified research problems was the need to develop geovisual analytics methods and tools that scale well to datasets with more than a few dimensions [And+07, Sec. 5.3.1], where we think Spatial BSS fits best. This issue seems to persist over time from a community and expert perspective [Çöl+17]. Geovisual analytics is closely tied to spatial statistics [Cre93; Wac03] supported by interactive visualizations. From a historical perspective, Haslett et al. [Has+91] and Gahegan et al. (GeoVISTA Studio) [Gah+02] were early examples of using interactive statistical graphics for the analysis of spatial data. Later on, researchers focused on supporting specific statistical methods. Examples include, e.g., Kulldorff's spatial scan statistic (Chen et al. [Che+08]), geographically weighted regression (Demšar et al. [DFC08]), geographically weighted discriminant analysis (Foley and Demšar [FD13]) or moving-window Kriging models (Demšar and Harris [DH11]). Perhaps better known in the visualization community is the work by members of the giCentre at the City University London. E.g., Dykes and Brunsdon [DB07] introduced several geographically-weighted visualizations, such as scalograms, maps, and boxplots. Wood and Dykes [WD08] suggested spatially-ordered treemaps. As a final example, Goodwin et al. [Goo+16] used local regression coefficients to guide the analysis of a spatial dataset on multiple scales.

Regarding the visualization of spatial point data we can broadly distinguish three visualization approaches (Figure 2.10).

Focus on Value and Preserve Location. This group of approaches preserves the point's location exactly and displays the associated value by some visual channels of the point's mark. In the univariate case, the marks are often circles and the channels their size or color (Figure 2.10d). For two variables, two channels might be combined. Elmer [Elm13] compared eight bivariate map designs across some tasks from which the bivariate choropleth design (a choropleth map using a two-dimensional color scale) emerged as the most accurate. Glyphs [War02; Bor+13] are commonly employed to depict more than two variables in maps. Opach et al. [Opa+18] compared star and polyline glyphs (Figure 2.10b) in a map view and in a spatially-unordered grid. They



(a) Phoenixmap shows class density as line thickness along a concave hull. Image: [Zha+21, Fig. 13b] © 2021 IEEE

(b) Star Glyphs show spatially varying multivariate data points on a map. Image: [Opa+18, Fig. 1d] © 2018 Taylor & Francis Group

(c) Nickel distribution in the Kola moss dataset (Section 1.1.3). Each rectangle shows the median value of contained points.

(d) Figure 2.10c with exact locations and five symbol sizes encoding value.

Figure 2.10: Examples of visualization approaches for point data. Approaches can be grouped into distorting value (c,d), distorting locations (a,c), neither (b) or both (c).

found that maps are to be preferred over the grid and star glyphs over polyline glyphs (most of the time) given their investigated tasks. Finally, even though the data itself are points, the visualization does not need to be: Attribute Signatures [Tur+14] is a collection of interactive visualizations in which the user defines a geometric path in a map, and small multiples of variables show each variable's value along the path as a line graph.

Focus on Value and Distort Location. The other group also displays values associated to points, but in a way that the point's exact position is lost. Visualization designers may go down this path when point marks would be otherwise too heavily occluded. The two strategies we identify here are **aggregation** and **displacement**. Aggregation usually distorts both value and location, while displacement distorts only the latter. For aggregation, isocontours (or, if colored, heatmaps) are one of the most widely known visualization techniques. Individual locations are not shown at all and they are replaced by curves denoting constant value. The contours themselves can be computed, e.g., with Marching Squares (gridded data), Meandering Triangles (non-gridded data) or kernel density estimation (non-gridded data). Jankowai and Hotz [JH20] generalized isocontours to multivariate data. A simpler aggregation technique is to overlay a regular tessellation onto the spatial domain, i.e., a grid of squares or hexagons, and aggregate values per tile (Figure 2.10c). This approach is easily extended to multivariate data [ZP04]. The main downside of it is, though, that the spatial distribution of locations often does not follow the tesselation's regularity, leading to some tiles comprised of many points and others of very few. Statistical summaries computed per tile can become less convincing as a consequence. There is also a modifiable areal unit problem (MAUP) as the visualization image depends on the tesselation's offset and tile size. Another angle to that approach is to learn the to-be-aggregated areas from the data itself. This idea runs under many names in the literature and has been called regionalization, districting, or zonation, althouth the former term seems to have established itself. A variety of algorithms exist [DRS07; Ayd+21] for different versions of the problem which is NP-hard even in its simplest case, but sophisticated heuristics were suggested as well [WRK21]. Displacement techniques try to avoid the MAUP altogether as they still show visual marks per point but displace them as little as possible to avoid occlusion and enable readability. Pixel maps [Kei00] use very small marks per point, which are moved to the nearest free location. As pixels are placed (and free spots filled) in the order of rows in the dataset, there is no guarantee about the quality of a particular pixel map. Point grid maps [Zho+17] also regularize point locations onto a grid but retain relative directions. The resulting map is overlap-free but often sparse. Meulemans [Meu19] suggests a linear programming approach to overlap removal of diamond-shaped symbols. Cutura et al. [Cut+21] use a pixel-maps-like approach (move to nearest free cell) on a space-filling curve. Finally, two papers consider both approaches. Opach et al. [Opa+19] compared various versions of aggregation and displacement for clutter reduction in zoomable maps. Perhaps surprisingly, their "control case" of no employed strategy was not much different in terms of accuracy or preference compared to the alternatives. McNabb and Laramee [ML19] suggest to show aggregated symbols if there is not enough space for individual symbols and the latter do not deviate too much from the former. A tree structure designed for that purpose, which has to be computed for each dataset once in the beginning, holds the information which symbols to show at what zoom level.

Focus on Density. The last group is common with categorical data as the focus is less on the point's associated value and more on the density and distribution of a category.

Similar considerations like to the other techniques apply, e.g., overplotting is an issue here, too. Splatterplots [MG13] solve the problem by replacing dense point regions with contour areas. BinSq [CM17] rasters the spatial domain with a quadtree and aligns points on its cells. It also samples points so that the displayed relative frequency approximates the actual frequency. Micro diagrams [GB20] show, e.g., pie charts on a raster of the spatial domain. Phoenixmap [Zha+21] encloses each category with a concave hull and encodes the spatially varying density as line thickness (Figure 2.10a). Jo et al. [Jo+19] devise a declarative rendering model for multiclass density maps. Their pipeline consists of binning, preprocessing, styling, rebinning, assembly, and finally rendering. Several common visualization approaches, such as hatching, weaving, or color-blending, are supported. It remains to be noted that, when considering spatially distributed categories, one ventures into the territory of set visualization, which will be discussed next.

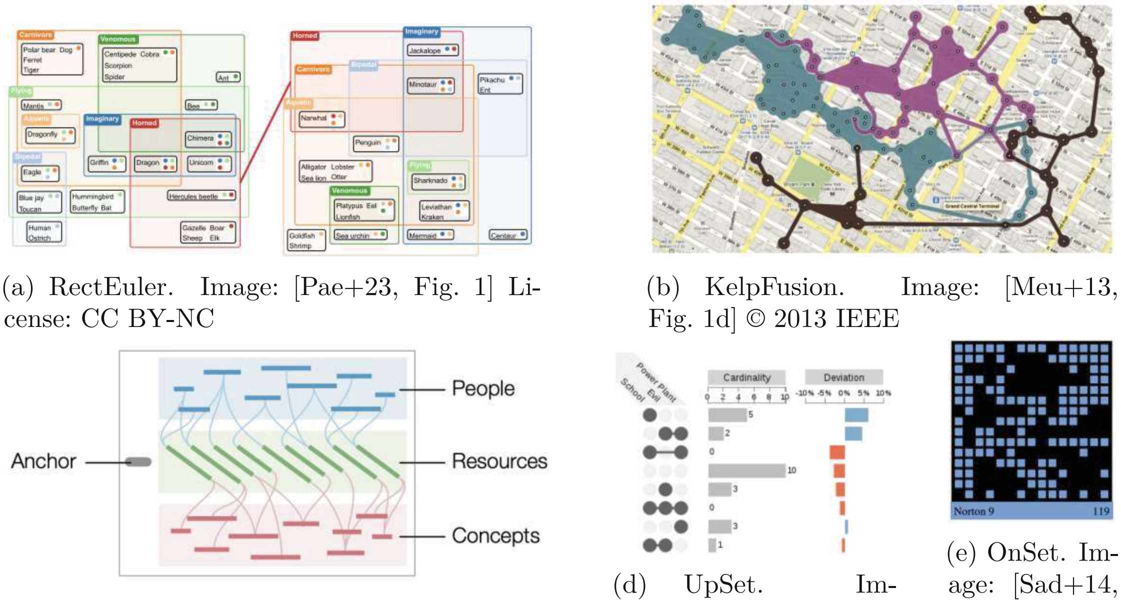
Relation to Blind Source Separation. The BSS approaches tackled in this thesis take multivariate time series (or spatial fields) as input and produce latent dimensions, which are a set of univariate time series (spatial fields). We can expect that BSS inputs and outputs need to be displayed at some point, which will be achieved with the presented visualization techniques. As analysts see input and latent dimensions mainly as separate variables, we will prefer combined univariate visualizations, such as small multiples. A property of BSS that seems not prominently discussed in the literature are the group structures (sets). A set of latent dimensions “belongs together;” they must be selected as a whole for downstream tasks and not mixed and matched. Similarly, some tasks are less interesting to analysts. An example would be comparing latent dimensions within a set, as these are by model assumption uncorrelated. Any visualization designs we use thus have to take care to always show these group structures.

2.3 Set Visualization

Sets are unordered collections of arbitrary elements. They often appear in data analysis as *categorical* variables whereof each data point may take on multiple values: E.g., the genres (e.g., action, romance, or drama) in a movie dataset would constitute sets. In biology, one could think of organisms’ habitat, like dry, forest, or Europe, as sets. Many more examples exist. Set relations, such as union, intersection, and difference can be calculated with set algebra. Sets are relevant to BSS because multiple runs with varying parameter settings produce a collection of latent dimensions. Analysts are interested in relations of these collections. E.g., asking what all possible latent dimensions are is akin to a *union* among the collections, while the search for stable latent dimensions that appear in many collections requires *intersection* operations.

Alsallakh et al. [Als+16] surveyed the field of set visualization and collected both visualization techniques and analysis tasks. The latter are divided into **element-related** tasks, such as finding a set containing a specific element, **set-related** tasks, such as determining which of two sets is bigger, and **attribute-related** tasks, where questions revolve around data distributions within and across sets, e.g., if the average box office income is higher for action movies than for horror movies. Regarding visualization techniques (Figure 2.11), the authors differentiate between various categories. **Euler/Venn diagrams** may be the best-known approach, where elements are enclosed by a closed curve and the element's position is arbitrary. It is often desired that Euler/Venn diagram techniques are well-matched (the relations between curves precisely reflect set relations) and well-formed (e.g., curves are simple, intersecting curves must cross, at most one curve per set, etc.), although it is not always possible to achieve both. Recent examples of this group include spEULER [Keh+22] and RectEuler (Figure 2.11a). The next group of set visualizations concerns **overlay** techniques. Here, the set information is secondary and overlaid onto another, primary visualization. A straightforward example are spatially distributed categories, where each each element has a defined spatial position and some categories, e.g., restaurants. This data type is also referred to as a "spatial hypergraph" [Bek+22], because sets link spatially positioned nodes via hyperedges. Techniques can be, among others, curve-based [Zha+21], like Euler/Venn diagrams, line-based [Alp+11], where a line connects elements of the same set, or a hybrid technique that employs both contours and lines, such as KelpFusion (Figure 2.11b). **Node-link techniques**, on the other hand, encode sets and elements as different types of nodes and membership as links between them. While the visual encoding is simple, these techniques often suffer from scalability issues as both sets and elements are encoded as the same primitive, competing for space, and many links usually entail many crossings. PivotPaths (Figure 2.11c) tackled these problems with layout, interactions and details-on-demand. **Matrix-based techniques** take their inspiration from adjacency matrices in graph visualization. Each row is a set, each column an element (or the other way around) and a filled cell indicates membership. These techniques may literally look like a matrix, as is the case for OnSet (Figure 2.11e), but linear diagrams [RSC15; WDN23] are also examples of that category. Finally, there are **aggregation-based techniques** where the number of elements is too large for individual visual marks and frequency representations are used instead. UpSet (Figure 2.11d) is an example for this technique, where each set intersection shown in a separate row and their size by a bar chart. PowerSet [AR17] and RainBio [LT20] are more recent examples.

Relation to Blind Source Separation. It becomes apparent that time and space are not well represented in set visualization. Regarding time, some works consider dynamic, i.e., time-varying sets [CPC09; AB20; Aga23] or categorical variables with temporal extent [Ngu+16]. GROUPSET [LV22] is an interesting recent idea where set visualizations are applied to time series. The value of each series at each time step is binned, so time steps become set elements and bins become sets. An UpSet-like technique is used to present the data. However, we are not aware of set visualizations that consider whole time series as elements, as would be the case of latent temporal dimensions in TBSS. Regarding space, current research in that direction seems to focus on spatial hypergraphs. Space in other forms than a position attribute for nodes seems less explored. E.g., latent spatial dimensions of SBSS would entail set elements that are maps along with vectors of loadings.



(a) RectEuler. Image: [Pae+23, Fig. 1] License: CC BY-NC

(b) KelpFusion. Image: [Meu+13, Fig. 1d] © 2013 IEEE

(c) PivotPaths. Image: [Dör+12, Fig. 3] © 2012 IEEE

(d) UpSet. Image: [Lex+14, Fig. 7b] © 2014 IEEE

(e) OnSet. Image: [Sad+14, Fig. 4] © 2014 IEEE

Figure 2.11: Examples of set visualization techniques: a) Euler/Venn diagrams, b) Overlay, c) Node-Link, d) Aggregation-based and e) Matrix.

Set elements are rarely seen as more complex data than, e.g., strings. While Alsallakh et al. [Als+16] do mention attribute-based tasks in their survey, these attributes are seen as numerical variables, such as the year a movie was released. It is also notable that to properly account for set relations among elements where it is not useful or desired to match them for 100 % equality, *fuzzy sets* are needed [PBM23a]. There are only a few visualization proposals for fuzzy sets [PP10a; PP10b; Zhu+18] and the challenges for uncertainty-aware set visualizations were recently recognized by established researchers [Tom+23].

2.4 Ensemble Visualization

Wang et al. [Wan+19] define ensemble data in their survey as “data that contains a collection of outputs generated from computer simulation models.” Examples would be hurricane trajectories [Liu+17] or precipitation forecasts [Bis+17]. More generally, ensemble data can refer to a collection of sufficiently complex elements [Xu+19]. Kehrer and Hauser [KH13], in their survey of multifaceted scientific data visualization, highlight the repeated simulations with perturbed parameter settings and describe ensemble data as *multirun data*. Other facets of scientific data include *spatio-temporal*, *multivariate*, *multimodal* (different acquisition modalities) and *multimodel* (multiple coupled simulation models).

Ensemble visualization is of interest to BSS as the latent dimensions obtained by a given parameter setting form an ensemble. Multiple runs of the algorithm produce multiple collections, which can be considered multiple sets (Section 2.3) or multiple ensembles (this section). Wang et al. [Wan+19] identified five orthogonal dimensions in ensemble data: Variable, location, time, member and ensemble. The first three refer to temporally and spatially distributed data, possibly of multiple variables. *Member* expresses to which member of an ensemble the data belongs, and *ensemble* to which ensemble, if there are multiple. The authors [Wan+19] further identify a common visualization pipeline for ensemble data (Figure 2.12).

Ensemble Data ⟶ (Aggregation) ⟶ Visualization ⟶ (Composition)

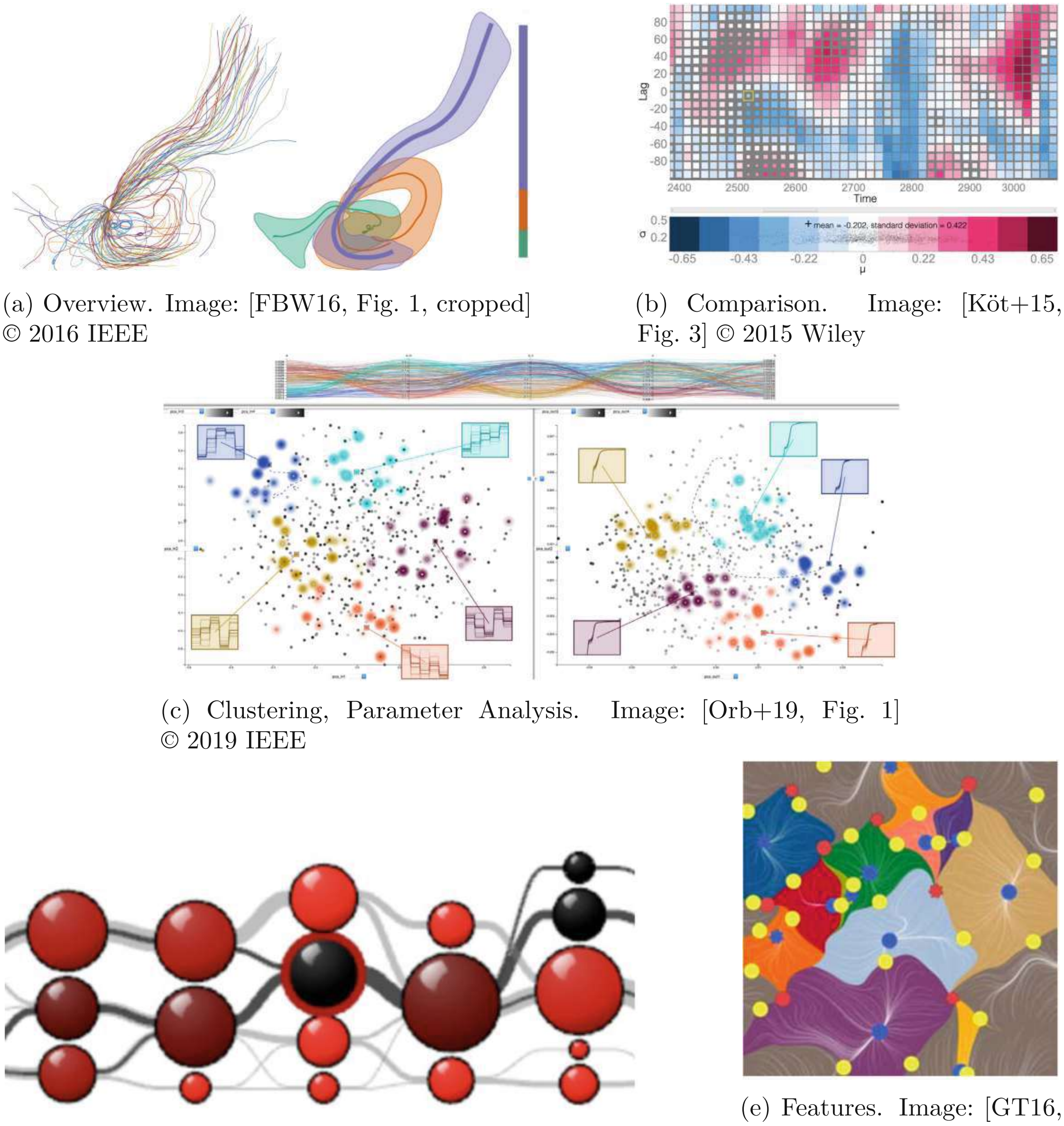
Figure 2.12: Ensemble visualization pipeline after Wang et al. [Wan+19]. Ensemble members are optionally aggregated (e.g., by statistical summaries), then visualized, and finally optionally composited.

Wang et al. [Wan+19] further identified analytic tasks. These include **overview**, where analysts seek a "concise visual summary" that conveys the overall uncertainty. An example would be the streamline variability plots by Ferstl et al. (Figure 2.13a). Through clustering in Principal Compononent space they obtain the major trends in the spaghetti plot, and replace individual lines with contours where width indicates variability. Liu et al. [Liu+17; Liu+19] work on the same data type, but convey the variability through examples by representative sampling (which is akin to a composited hypothetical outcome plot [Kal+19]). Another common analytic task for ensemble data is **comparison**, where analysts seek to compare two or more members or even ensembles. Considerations for visual comparison brought forth by Gleicher et al. [Gle+11; Gle18] apply here. An example for visual comparison of two ensembles of time series was demonstrated by Köthur et al. (Figure 2.13b). To compare individual time series, the authors use windowed cross-correlation (WCC), i.e., time series are first split into equal-length windows, then correlation (e.g., Pearson's) is measured between windows. A lag, i.e., offset, accounts for shifted time series. Thus, the top matrix plots aggregations of WCC in lag vs. time, where a cell's color indicates the median direction and amplitude of correlation, while cell size inversely expresses the spread of WCC values. The bottom strip plots individual window-lag combinations in a scatterplot of mean vs. standard deviation. **Clustering** was identified as another common task. Classifying complex objects is on the one hand useful for overview tasks as well, but also supports reasoning about *why* these groups appear, about the physical processes that generate them. To support clustering, of course actual clustering algorithms, of which there are plenty [XW05; XT15], may be used, although custom distance functions are often required. Another possibility is to use DR embeddings and rely on visual cluster analysis (Figure 2.13c, right plot). Wang et al. also identified **temporal trend analysis** as an important analytic task. Here, the evolution of ensemble members over time is of interest. Common visualization techniques include superpositioned line charts, possibly showing statistical summaries of members (cf. Figure 2.12), small multiples of members at different time points, or specialized visiualizations such as the Trend Graph (Figure 2.13d). Here, each circle represents a cluster of ensemble members and each column a time window. How members move from one state to the next is expressed by lines between circles. Major trends thus appear as thick lines between large circles. The survey authors further include **feature extraction** in their analysis tasks. Analysts want to obtain features from the ensemble set, such as saddles, sources, sinks, eddies or vortexes. After computational methods identified those features, they are usually overlaid over the spatial domain. Figure 2.13e shows critical points in a 2D vector field: Yellow dots are saddles, blue dots are attractors and red dots repellants. Finally, **parameter analysis** (cf. Section 2.1) is relevant to analysts of ensemble data. They want to understand how parameters influence the outcome, both in value and uncertainty, or identify sensitive parameter subspaces. As parameters for simulation models are often multivariate, it is possible to employ visualization techniques for high-dimensional data, such as parallel coordinates plots, scatterplot matrices, or DR embeddings. To build the connection to the simulation output, coordinated multiple views are often used. Consider again Figure 2.13c. The left plot shows a DR view of multivariate parameters, the right plot a DR view of simulation outputs (time series). The top plot shows all simulation outputs. This allows the analyst brushing and linking in the outputs directly (top plot), parameter clusters (left plot), or output clusters (right plot).

Relation to Blind Source Separation. The previously mentioned analytic tasks are relevant for BSS, too, but less so on the *member* dimension. As latent dimensions identified by the same parameter setting are, because of model assumptions, marginally and temporally/spatially uncorrelated, there is, e.g., not much point in *clustering*. *Temporal trends* or *comparison* among latent dimensions are also not relevant within an ensemble. This changes, however, when we consider latent dimensions from *different* BSS methods. No limiting assumptions exist between those and analyzing them can support parameter analysis. Considering the relative importance of the *ensemble* dimension, one has to realize that not many visualization approaches deal with multiple ensembles. Köthur et al. [Köt+15] compare two ensembles of time series, but their approach, as it is so often the case, does not translate to three or more ensembles. Biswas et al. [Bis+17] and Wang et al. [Wan+17] consider three ensembles of precipitation forecasts. Their main goal is to analyze the forecast's *uncertainty* [Sed+14] depending on the simulated grid size. For BSS, the ensemble visualization tasks *overview*, *parameter analysis*, *comparison* and *clustering* have to be supported across many more than three ensembles.

2.5 Dimensionality Reduction

The basic idea in DR is that high-dimensional structures and relationships of data points can be approximated by a low-dimensional subspace. The low-dimensional representation of high-dimensional data is called an *embedding* or *projection*. As a general guideline, high-dimensionally similar data points remain close to each other in the embedding, while



(a) Overview. Image: [FBW16, Fig. 1, cropped] © 2016 IEEE

(b) Comparison. Image: [Köt+15, Fig. 3] © 2015 Wiley

(c) Clustering, Parameter Analysis. Image: [Orb+19, Fig. 1] © 2019 IEEE

(d) Temporal Trends. Image: [OBJ16, Fig. 5] © 2016 IEEE

(e) Features. Image: [GT16, Fig. 5] © 2016 Wiley

Figure 2.13: Examples of visualization techniques to support analytic tasks in ensemble visualization.

high-dimensionally different data points remain far apart. Embeddings are extremely helpful for visualization because position is the most effective visual channel, but we cannot reasonably display more than three dimensions.[2] A lot of algorithms have been

[2] Even three dimensions are generally discouraged for abstract data due to issues like occlusion or perspective distortion.

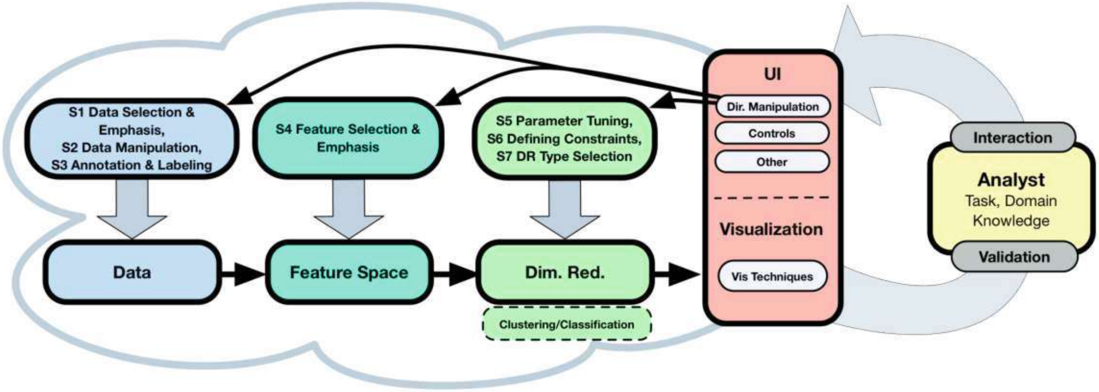

S1 Data Selection & Emphasis, S2 Data Manipulation, S3 Annotation & Labeling
S4 Feature Selection & Emphasis
S5 Parameter Tuning, S6 Defining Constraints, S7 DR Type Selection
UI
Dir. Manipulation
Controls
Other
Visualization
Vis Techniques
Data
Feature Space
Dim. Red.
Clustering/Classification
Interaction
Analyst
Task, Domain Knowledge
Validation

Figure 2.14: Human-in-the-loop process model for interactive DR. Interactions with DR commonly pertain to the data itself (e.g., selecting or weighting data points), the feature space (e.g., selecting or weighting variables), or the DR algorithm itself (e.g., parameter tuning). Image: [Sac+17, Fig. 6] © 2017 IEEE

proposed and used for that purpose over the years [NA19]. A well-known example is, e.g., PCA, which finds orthogonal directions of highest variance. Nonato and Espadoto [NA19] provide a survey of DR techniques in relation to data characteristics (e.g., does it support categorical data?), desired properties (e.g., can it project out-of-sample data points?), and user tasks (e.g., identifying clusters). Regarding the latter, the authors identified four groups: **Generate Map** (produce the embedding itself), **Explore Dimensions** (e.g., mapping synthetic dimensions to original dimensions), **Explore Items in Base Layout** (e.g., identifying clusters in DR scatterplot), and **Explore Items in Enriched Layout** (e.g., identifying class outliers). Sacha et al. [Sac+17] structured visual interaction with DR (Figure 2.14) and found that it can happen in the **data transformations** part of the InfoVis pipeline, i.e., **data selection**, **data manipulation**, **data annotation**, or **feature selection**. Further, **parameter tuning** is relevant for many DR techniques, the **selection** of the technique itself, or any **constraints** that the embedding should adhere to.

However, to human analysts, the DR process is often a “black box” and the inner workings and the resulting embedding are poorly understood. Several questions may arise.

For instance, since the embedding will by definition not perfectly resemble the original high-dimensional space, one could wonder how truthful an embedding actually is. Several quality metrics have been proposed over time (see, e.g., [Aup07; Esp+19; Mor+23] and references therein) that may be used to assess the *projection error*. A main idea here is that the data points’ neighborhoods should be preserved, which is sometimes referred to as *false neighbor* (point is nearby in embedding but should not be) and *missing neighbor* (point is far away in embedding but should not be). Once computed, the amount of error in a points’ neighborhood can be added to the embedding via colored areas [LA11; RKW20; Jeo+22]. In relation to data points, the distances of other points can be corrected [Sta+16]. When the data points belong to multiple categories, it is helpful to know where the category “borders” in the embedding are. Espadoto et al. [Esp+23] propose for that purpose *inverse projections*: Each screen pixel is back-projected into the original data space, where an ensemble of classifiers assigns a class to the high-dimensional point. The screen pixel is then colored based on ensemble agreement. Ma et al. [MM20], on the other hand, developed an interactive VA approach to investigate class separations.

Many state-of-the-art techniques require parameters that have to be carefully set. E.g., UMAP [MHM18], a neighbor-embedding technique based on k-neighborhood graphs, requires (amongst others) a value for k. Setting it too low will lead to a focus on very small neighborhoods and favor local over global structures being visible in the embedding. Setting k too high will have the opposite effect. Another question analysts naturally ask is thus which parameter settings produce a “good” embedding. Espadoto et al. [Esp+19] ran a quantitative survey of DR techniques, in which they calculated several quality metrics for embeddings produced by various techniques on different datasets. While this of course does not take into account the dataset at hand, it can give general guidelines. Etemadpour et al. [Ete+15] conducted controlled experiments to determine the suitability of some DR techniques for visual cluster analysis. Xia et al. [Xia+22] did a similar study and specifically distinguished between DR properties, such as non-linear/linear and local/global. Morariu et al. [Mor+23] investigated how proposed DR quality metrics match with user preference. Their models suggest that, e.g., scagnostics [WAG06], like Sparsity, Skinny and Outlying, correspond well to user preference. Another angle to preference of spatializations was investigated by Wenskovitch et al. [WN20]. They asked participants to place cards, representing high-dimensional data points, in a plane. Participants were only allowed to use *grouping* and *spatialization* operations, i.e., putting two cards in the same group or placing them some distance apart. The results are insofar interesting as original dimensions were both curved and overlapping as well as nested in the produced embedding. Especially the latter is not a common feature of computational DR techniques.

In that study, participants had to think hard about how to organize the data points in order to come up with synthetic dimensions. The layout produced by one participant might not be intuitive for the next. We encounter the same situation when the embedding is computer-generated. Analysts often wonder, e.g., how the space is organized and what the synthetic axes mean. Several visual and interactive aids were proposed to that end. Gleicher et al. [Gle13] reject the premise and suggest a system where synthetic dimensions must adhere to user-defined spatialization constraints. E.g., ordering cities by “more like New York” along the X axis and “more like Los Angeles” on the Y axis, naturally forces the X and and Y axis to correspond to these made-up dimensions. Endert et al. [End+11] called this approach *observation-level interaction* and several variations were proposed over the years [Kim+16; WN17; Sel+18; Dow+19]. Another option is to compute the axis lines and add them to the embedding [FGS19]. Axis lines, being either normal or parallel to each other in a cartesian grid, are curved for non-linear synthetic dimensions and indicate their direction. Plotting the distribution of individual original dimensions in the embedding is a simple way to hint at how synthetic dimensions combine the originals [Sil+15; Sta+16]. This approach also highlights (single-dimension) clusters. Other approaches to do so are, e.g., area overlays for group labels [Soh+22] or adjusting the shape of the data points' visual marks to indicate local cluster structures [Bia+20].

Finally, VA approaches for DR techniques have been proposed, such as iPCA for PCA [Jeo+09] and t-viSNE for t-SNE [CMK20], that combine several of the previously described techniques.

Relation to Blind Source Separation. DR shares several high-level tasks with BSS. Parametrizing the DR algorithm is a challenge here just as much as in BSS, albeit with somewhat simpler parameters and a study suggests that defaults of popular techniques can work well across datasets [Esp+19]. Structures in the embedding have to be found and explained, similar to BSS latent dimensions. If synthetic dimensions are removed, the remaining dimensions do not describe the original data perfectly anymore, and analysts are interested what kind of errors are located where. The same is true in BSS. The applicability of proposed solutions in DR to BSS is influenced by the fact that in BSS, the order of points is fixed in time and space, so the errors that can occur are of a different kind. Regarding parameter settings we refer the reader to Section 2.1, where those challenges are discussed in detail. Finally, to support explaining latent dimensions and patterns therein, certain suggested visualizations could be adapted. E.g., showing the prevalence of original dimensions [Sil+15; Sta+16] could be helpful in BSS too.

2.6 Guidance

An issue of VA is that not only are the data under scrutiny complex, the visual interfaces and automatic analysis methods may be, too. Users can get stuck during analysis and need help. *Guidance* tackles this challenge. It was defined recently by Ceneda et al. [Cen+17, p. 112] as a "computer-assisted process that aims to actively resolve a knowledge gap encountered by users during an interactive visual analytics session." The main aspects of guidance (Figure 2.15), according to the authors, are the type of *knowledge gap*, what it pertains to (*domain*), the guidance *input* and *output*, as well as how much guidance is given to the user (*degree*). The concept of guidance focuses on the direction from the computer to the human, but the implementation of a guidance process may leverage the user's interaction history to decide when and how to provide help. As such, guidance is a *mixed-initiative process*. Ceneda et al. [CGM19] provided a survey about such guidance approaches.

Collins et al. [Col+18] discuss guidance in more practical terms and considerations. They suggest possible goals of guidance: To inform, to mitigate bias, to reduce cognitive load, for training, for engagement, or to verify conclusions. In addition, the authors provide possible types of knowledge (session-specific, prior, situation) that a guidance process may take advantage of. Also to make guidance design more practical, Ceneda et al. [Cen+20] described a framework for designers. It aims for effective guidance (available, trustworthy, adaptive, controllable, non-disruptive) and proposes four steps

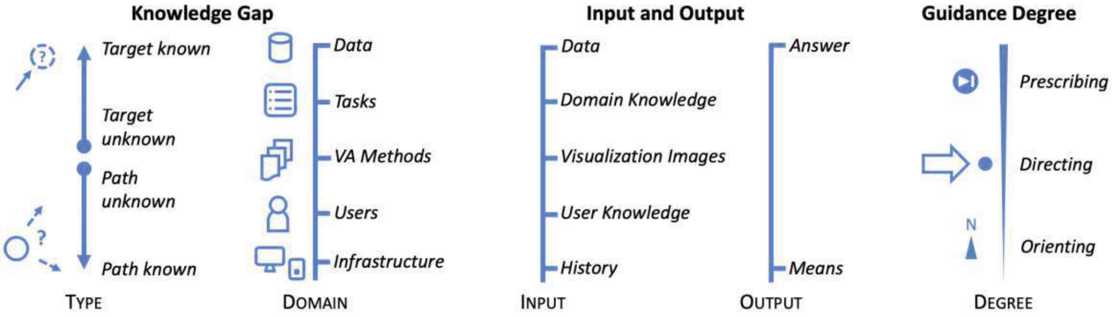


Figure 2.15: Guidance characterization by Ceneda et al. Image: [Cen+17, Fig. 1] © 2017 IEEE

that the designer needs to take. First, they have to understand the analyst's goals and analysis phases. Second, the designer needs to look for and verify possible knowledge gaps the analyst may encounter. Third, guidance is designed for these gaps by looking for possible inputs, outputs, and degrees. Finally, the guidance designer considers how user feedback may be leveraged to adapt guidance. A crucial aspect of guidance is that it is given in a timely manner, i.e., when the user needs it and not before or after. Ceneda et al. [Cen+21] trained a neural network to detect when guidance is needed based on the user's facial expression. To evaluate a designed guidance process, Ceneda et al. [Cen+24] proposed a dual heuristic approach involving experts and end-users. Researchers also tried to include the computer's perspective into guidance models. Sperrle et al. [Spe+21] introduced co-adaptive guidance, arguing that both user and computer adapt over time to the behavior of the other. The guidance design space is supplemented by a learning-teaching axis. Pérez-Messina et al. [Pér+22] introduced a typology of guidance tasks in mixed-initiative VA environments. Based on the user's search task [BM13] and guidance degree, they introduce several guidance tasks. E.g., if target and path are known (*lookup*) and *orienting* guidance is used, then the system can *pinpoint* the sought data. One of their main observations is that if search and guidance tasks are mismatched, the analysis may be disrupted when the search task is forcibly changed. This is the case, e.g., when combining an *explore* task with *prescribing* guidance, which makes it a *lookup*.

Relation to Blind Source Separation. The goals in BSS are relatively clear. Analysts seek suitable tuning parameter settings and common/uncommon latent components with/without relevant features. However, achieving these goals can be daunting, as, e.g., the tuning parameter space is huge (compare RQ1 in Section 1.2). As such, guidance is very likely needed. The analyst's *knowledge gap* pertains to the data (parameter settings, components, features therein) and we expect that the guidance *input* is also mainly the data (input variables) and user/domain knowledge. To complete all steps in the guidance design framework [Cen+20], we have to investigate in the remainder of the thesis possible guidance degrees, outputs, means, and feedback mechanisms.

2.7 Bibliography

[AE20] A. M. Abuzuraiq and H. Erhan. “The Many Faces of Similarity”. In: *Design in the Age of Humans - Proceedings of the 25th CAADRIA Conference*. Vol. 1. Hong Kong: Assiciation for Computer-Aided Architectural Design Research in Asia, 2020, pp. 485–494. ISBN: 978-988-789-174-1. DOI: 10.52842/conf.caadria.2020.1.485 (cit. on pp. 46, 50, 51, 54).

[Aga23] S. Agarwal. “Visualizing Element Interactions in Dynamic Overlapping Sets”. In: *EuroVis 2023 - Short Papers*. The Eurographics Association, 2023. DOI: 10.2312/evs.20231050 (cit. on p. 69).

[AB20] S. Agarwal and F. Beck. “Set Streams: Visual Exploration of Dynamic Overlapping Sets”. In: *Computer Graphics Forum* 39.3 (2020), pp. 383–391. DOI: 10.1111/cgf.13988 (cit. on p. 69).

[Aig+11] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. 1st edition. Berlin, Heidelberg: Springer, 2011. 286 pp. ISBN: 978-0-85729-078-6 (cit. on pp. 61, 62).

[ARB07] D. G. Aliaga, P. A. Rosen, and D. R. Bekins. “Style Grammars for Interactive Visualization of Architecture”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.4 (2007), pp. 786–797. DOI: 10.1109/tvcg.2007.1024 (cit. on p. 59).

[Alp+11] B. Alper, N. Riche, G. Ramos, and M. Czerwinski. “Design Study of LineSets, a Novel Set Visualization Technique”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2259–2267. DOI: 10.1109/tvcg.2011.186 (cit. on p. 69).

[AR17] B. Alsallakh and L. Ren. “PowerSet: A Comprehensive Visualization of Set Intersections”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 361–370. DOI: 10.1109/tvcg.2016.2598496 (cit. on p. 69).

[Als+16] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers. “The State-of-the-Art of Set Visualization”. In: *Computer Graphics Forum* 35.1 (2016), pp. 234–260. DOI: 10.1111/cgf.12722 (cit. on pp. 68, 70).

[Ami+10] A. Amirkhanov, C. Heinzl, M. Reiter, and E. Gröller. “Visual Optimality and Stability Analysis of 3DCT Scan Positions”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 1477–1486. DOI: 10.1109/tvcg.2010.214 (cit. on pp. 47, 51).

[And+07] G. Andrienko, N. Andrienko, P. Jankowski, D. Keim, M.-J. Kraak, A. MacEachren, and S. Wrobel. “Geovisual Analytics for Spatial Decision Support: Setting the Research Agenda”. In: *International Journal of Geographical Information Science* 21.8 (2007), pp. 839–857. DOI: 10.1080/13658810701349011 (cit. on p. 65).

[And+20] N. Andrienko, G. Andrienko, G. Fuchs, A. Slingsby, C. Turkay, and S. Wrobel. *Visual Analytics for Data Scientists.* Cham, Switzerland: Springer, 2020. 440 pp. ISBN: 978-3-030-56145-1 (cit. on p. 65).

[Ang+18] M. Angelini, G. Santucci, H. Schumann, and H.-J. Schulz. "A Review and Characterization of Progressive Visual Analytics". In: *Informatics* 5.3 (3 2018), p. 31. DOI: `10.3390/informatics5030031` (cit. on p. 57).

[Aup07] M. Aupetit. "Visualizing Distortions and Recovering Topology in Continuous Projection Techniques". In: *Neurocomputing* 70.7 (2007), pp. 1304–1330. DOI: `10.1016/j.neucom.2006.11.018` (cit. on p. 74).

[Ayd+21] O. Aydin, Mark. V. Janikas, R. M. Assunção, and T.-H. Lee. "A Quantitative Comparison of Regionalization Methods". In: *International Journal of Geographical Information Science* 35.11 (2021), pp. 2287–2315. DOI: `10.1080/13658816.2021.1905819` (cit. on p. 67).

[Bac+17] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, and S. Carpendale. "A Descriptive Framework for Temporal Data Visualizations Based on Generalized Space-Time Cubes". In: *Computer Graphics Forum* 36.6 (2017), pp. 36–61. DOI: `10.1111/cgf.12804` (cit. on p. 61).

[Bal99] R. Balling. "Design by shopping: A new paradigm". In: *Proceedings of the Third World Congress of structural and multidisciplinary optimization (WCSMO-3).* Vol. 1. University at Buffalo, Buffalo, NY. 1999, pp. 295–297 (cit. on p. 45).

[Bao+13] F. Bao, D.-M. Yan, N. J. Mitra, and P. Wonka. "Generating and Exploring Good Building Layouts". In: *ACM Transactions on Graphics* 32.4 (2013), pp. 1–10. DOI: `10.1145/2461912.2461977` (cit. on p. 54).

[Beh+14] M. Beham, W. Herzner, M. E. Gröller, and J. Kehrer. "Cupid: Cluster-Based Exploration of Geometry Generators with Parallel Coordinates and Radial Trees". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1693–1702. DOI: `10.1109/tvcg.2014.2346626` (cit. on pp. 50, 51, 53, 54).

[Bek+22] M. Bekos, D. Dekker, F. Frank, W. Meulemans, P. Rodgers, A. Schulz, and S. Wessel. "Computing Schematic Layouts for Spatial Hypergraphs on Concentric Circles and Grids". In: *Computer Graphics Forum* 41.6 (2022), pp. 316–335. DOI: `10.1111/cgf.14497` (cit. on p. 69).

[Ber+12] J. Bernard, T. Ruppert, O. Goroll, T. May, and J. Kohlhammer. "Visual-Interactive Preprocessing of Time Series Data". In: *SIGRAD 2012, Interactive Visual Analysis of Data. Proceedings.* Linköping University Electronic Press, 2012, pp. 39–48. ISBN: 978-91-7519-723-4 (cit. on pp. 55, 56).

[Ber+18] J. Bernard, C. Bors, M. Bögl, C. Eichner, T. Gschwandtner, S. Miksch, H. Schumann, and J. Kohlhammer. "Combining the Automated Segmentation and Visual Analysis of Multivariate Time Series". In: *EuroVis Workshop on Visual Analytics (EuroVA)*. Ed. by C. Tominski and T. von Landesberger. The Eurographics Association, 2018. ISBN: 978-3-03868-064-2. DOI: 10.2312/eurova.20181112 (cit. on pp. 42, 54).

[Ber+16] J. Bernard, E. Dobermann, M. Bögl, M. Röhlig, A. Vögele, and J. Kohlhammer. "Visual-Interactive Segmentation of Multivariate Time Series". In: *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, 2016, p. 5. ISBN: 978-3-03868-016-1. DOI: 10.2312/eurova.20161121 (cit. on p. 53).

[Ber+19] J. Bernard, M. Hutter, H. Reinemuth, H. Pfeifer, C. Bors, and J. Kohlhammer. "Visual-Interactive Preprocessing of Multivariate Time Series Data". In: *Computer Graphics Forum* 38.3 (2019), pp. 401–412. DOI: 10.1111/cgf.13698 (cit. on pp. 50, 51, 55).

[Bia+20] R. Bian, Y. Xue, L. Zhou, J. Zhang, B. Chen, D. Weiskopf, and Y. Wang. "Implicit Multidimensional Projection of Local Subspaces". In: *IEEE Transactions on Visualization and Computer Graphics* (2020), pp. 1–1. DOI: 10.1109/tvcg.2020.3030368 (cit. on p. 76).

[Bis+17] A. Biswas, G. Lin, X. Liu, and H.-W. Shen. "Visualization of Time-Varying Weather Ensembles across Multiple Resolutions". In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 841–850. DOI: 10.1109/tvcg.2016.2598869 (cit. on pp. 40, 50, 51, 54, 70, 72).

[Boc+15] A. Bock, A. Pembroke, M. L. Mays, L. Rastaetter, T. Ropinski, and A. Ynnerman. "Visual Verification of Space Weather Ensemble Simulations". In: *IEEE Scientific Visualization Conference*. 2015, pp. 17–24. DOI: 10.1109/SciVis.2015.7429487 (cit. on p. 51).

[Bög+13] M. Bögl, W. Aigner, P. Filzmoser, T. Lammarsch, S. Miksch, and A. Rind. "Visual Analytics for Model Selection in Time Series Analysis". In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2237–2246. DOI: 10.1109/tvcg.2013.222 (cit. on p. 56).

[Bög+14] M. Bögl, W. Aigner, P. Filzmoser, T. Gschwandtner, T. Lammarsch, S. Miksch, and A. Rind. "Visual Analytics Methods to Guide Diagnostics for Time Series Model Predictions". In: *Proceedings of the 2014 IEEE VIS Workshop on Visualization for Predictive Analytics*. Paris, France, 2014 (cit. on p. 56).

[Bor+13] R. Borgo, J. Kehrer, D. H. S. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward, and M. Chen. "Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications". In: *Eurographics 2013 - State of the Art Reports*. Ed. by M. Sbert and L. Szirmay-Kalos. The Eurographics

Association, 2013. DOI: 10.2312/conf/EG2013/stars/039-063 (cit. on p. 65).

[Bor07] E. Borgonovo. “A New Uncertainty Importance Measure”. In: *Reliability Engineering & System Safety* 92.6 (2007), pp. 771–784. DOI: 10.1016/j.ress.2006.04.015 (cit. on p. 49).

[BP16] E. Borgonovo and E. Plischke. “Sensitivity Analysis: A Review of Recent Advances”. In: *European Journal of Operational Research* 248.3 (2016), pp. 869–887. DOI: 10.1016/j.ejor.2015.06.032 (cit. on p. 40).

[Bor+17] C. Bors, M. Bögl, T. Gschwandtner, and S. Miksch. “Visual Support for Rastering of Unequally Spaced Time Series”. In: *Proceedings of the 10th International Symposium on Visual Information Communication and Interaction* (Bangkok, Thailand). New York, NY, USA: ACM, 2017, pp. 53–57. ISBN: 978-1-4503-5292-5. DOI: 10.1145/3105971.3105984 (cit. on pp. 45, 56).

[BC06] V. Braun and V. Clarke. “Using Thematic Analysis in Psychology”. In: *Qualitative Research in Psychology* 3.2 (2006), pp. 77–101. DOI: 10.1191/1478088706qp063oa (cit. on p. 43).

[BC19] V. Braun and V. Clarke. “Reflecting on Reflexive Thematic Analysis”. In: *Qualitative Research in Sport, Exercise and Health* 11.4 (2019), pp. 589–597. DOI: 10.1080/2159676x.2019.1628806 (cit. on p. 43).

[BC21a] V. Braun and V. Clarke. “Can I Use TA? Should I Use TA? Should I Not Use TA? Comparing Reflexive Thematic Analysis and Other Pattern-Based Qualitative Analytic Approaches”. In: *Counselling and Psychotherapy Research* 21.1 (2021), pp. 37–47. DOI: 10.1002/capr.12360 (cit. on p. 43).

[BC21b] V. Braun and V. Clarke. “One Size Fits All? What Counts as Quality Practice in (Reflexive) Thematic Analysis?” In: *Qualitative Research in Psychology* 18.3 (2021), pp. 328–352. DOI: 10.1080/14780887.2020.1769238 (cit. on p. 43).

[BM13] M. Brehmer and T. Munzner. “A Multi-Level Typology of Abstract Visualization Tasks”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2376–2385. DOI: 10.1109/tvcg.2013.124 (cit. on p. 77).

[BM10] S. Bruckner and T. Möller. “Result-Driven Exploration of Simulation Parameter Spaces for Visual Effects Design”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 1468–1476. DOI: 10.1109/tvcg.2010.190 (cit. on pp. 50, 54).

[Bru+16] N. Brunhart-Lupo, B. W. Bush, K. Gruchalla, and S. Smith. “Simulation Exploration through Immersive Parallel Planes”. In: *2016 Workshop on Immersive Analytics (IA)*. 2016, pp. 19–24. DOI: 10.1109/immersive.2016.7932377 (cit. on pp. 45, 59).

[Bry+15] C. Bryan, X. Wu, S. Mniszewski, and K.-L. Ma. “Integrating Predictive Analytics into a Spatiotemporal Epidemic Simulation”. In: *IEEE Conference on Visual Analytics Science and Technology*. IEEE, 2015, pp. 17–24. DOI: 10.1109/vast.2015.7347626 (cit. on pp. 50, 55).

[Cao+18] N. Cao, Y.-R. Lin, D. Gotz, and F. Du. “Z-Glyph: Visualizing Outliers in Multivariate Data”. In: *Information Visualization* 17.1 (2018), pp. 22–40. DOI: 10.1177/1473871616686635 (cit. on p. 63).

[CMS99] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. 1st Edition. Morgan Kaufmann, 1999. ISBN: 1-55860-533-9 (cit. on p. 43).

[Cen+17] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H. Schulz, M. Streit, and C. Tominski. “Characterizing Guidance in Visual Analytics”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 111–120. DOI: 10.1109/tvcg.2016.2598468 (cit. on pp. 56, 76, 77).

[Cen+20] D. Ceneda, N. Andrienko, G. Andrienko, T. Gschwandtner, S. Miksch, N. Piccolotto, T. Schreck, M. Streit, J. Suschnigg, and C. Tominski. “Guide Me in Analysis: A Framework for Guidance Designers”. In: *Computer Graphics Forum* 39.6 (2020), pp. 269–288. DOI: 10.1111/cgf.14017 (cit. on pp. 76, 77).

[Cen+21] D. Ceneda, A. Arleo, T. Gschwandtner, and S. Miksch. “Show Me Your Face: Towards an Automated Method to Provide Timely Guidance in Visual Analytics”. In: *IEEE Transactions on Visualization and Computer Graphics* (2021), pp. 1–1. DOI: 10.1109/tvcg.2021.3094870 (cit. on pp. 57, 77).

[Cen+24] D. Ceneda, C. Collins, M. El-Assady, S. Miksch, C. Tominski, and A. Arleo. “A Heuristic Approach for Dual Expert/End-User Evaluation of Guidance in Visual Analytics”. In: *IEEE Transactions on Visualization and Computer Graphics* 30.1 (2024), pp. 997–1007. DOI: 10.1109/TVCG.2023.3327152 (cit. on p. 77).

[Cen+18] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, M. Streit, and C. Tominski. “Guidance or No Guidance? A Decision Tree Can Help”. In: (2018), p. 5 (cit. on p. 56).

[CGM19] D. Ceneda, T. Gschwandtner, and S. Miksch. “A Review of Guidance Approaches in Visual Data Analysis: A Multifocal Perspective”. In: *Computer Graphics Forum* 38.3 (2019), pp. 861–879. DOI: 10.1111/cgf.13730 (cit. on pp. 56, 76).

[Cha+20] A. Chatzimparmpas, R. M. Martins, I. Jusufi, and A. Kerren. “A Survey of Surveys on the Use of Visualization for Interpreting Machine Learning Models”. In: *Information Visualization* 19.3 (2020), pp. 207–233. DOI: 10.1177/1473871620904671 (cit. on p. 59).

[CMK20] A. Chatzimparmpas, R. M. Martins, and A. Kerren. “T-viSNE: Interactive Assessment and Interpretation of t-SNE Projections”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.8 (2020), pp. 2696–2714. DOI: 10.1109/tvcg.2020.2986996 (cit. on p. 76).

[Che+08] J. Chen, R. E. Roth, A. T. Naito, E. J. Lengerich, and A. M. MacEachren. “Geovisual Analytics to Enhance Spatial Scan Statistic Interpretation: An Analysis of U.S. Cervical Cancer Mortality”. In: *International Journal of Health Geographics* 7.1 (2008), p. 57. DOI: 10.1186/1476-072X-7-57 (cit. on p. 65).

[CM17] A. Chua and A. V. Moere. “BinSq: Visualizing Geographic Dot Density Patterns with Gridded Maps”. In: *Cartography and Geographic Information Science* 44.5 (2017), pp. 390–409. DOI: 10.1080/15230406.2016.1174623 (cit. on p. 68).

[Cib+17] L. Cibulski, B. Klarin, M. Sopouch, B. Preim, H. Theisel, and K. Matković. “Super-Ensembler: Interactive Visual Analysis of Data Surface Sets”. In: *Proceedings of the 33rd Spring Conference on Computer Graphics*. New York, NY, USA: ACM, 2017, pp. 1–9. ISBN: 978-1-4503-5107-2. DOI: 10.1145/3154353.3154362 (cit. on pp. 50, 53).

[CV11] J. H. T. Claessen and J. J. Van Wijk. “Flexible Linked Axes for Multivariate Data Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2310–2316. DOI: 10.1109/tvcg.2011.201 (cit. on p. 64).

[CKB09] A. Cockburn, A. Karlson, and B. B. Bederson. “A Review of Overview+Detail, Zooming, and Focus+Context Interfaces”. In: *ACM Comput. Surv.* 41.1 (2009), pp. 1–31. DOI: 10.1145/1456650.1456652 (cit. on pp. 50, 64).

[Cof+13] D. Coffey, C.-L. Lin, A. G. Erdman, and D. F. Keefe. “Design by Dragging: An Interface for Creative Forward and Inverse Design with Simulation Ensembles”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2783–2791. DOI: 10.1109/tvcg.2013.147 (cit. on pp. 41, 45, 50).

[Col+18] C. Collins, N. Andrienko, T. Schreck, J. Yang, J. Choo, U. Engelke, A. Jena, and T. Dwyer. “Guidance in the Human–Machine Analytics Process”. In: *Visual Informatics* 2.3 (2018), pp. 166–180. DOI: 10.1016/j.visinf.2018.09.003 (cit. on pp. 56, 76).

[CPC09] C. Collins, G. Penn, and S. Carpendale. “Bubble Sets: Revealing Set Relations with Isocontours over Existing Visualizations”. In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (2009), pp. 1009–1016. DOI: 10.1109/tvcg.2009.122 (cit. on p. 69).

[Çöl+17] A. Çöltekin, S. Bleisch, G. Andrienko, and J. Dykes. “Persistent Challenges in Geovisualization – a Community Perspective”. In: *International Journal of Cartography* 3 (sup1 2017), pp. 115–139. DOI: 10.1080/23729333.2017.1302910 (cit. on p. 65).

[Cre93] N. A. Cressie. “Statistics for Spatial Data”. In: *Statistics for Spatial Data*. John Wiley & Sons, Ltd, 1993, pp. 1–26. ISBN: 978-1-119-11515-1. DOI: 10.1002/9781119115151.ch1 (cit. on p. 65).

[Cut+21] R. Cutura, C. Morariu, Z. Cheng, Y. Wang, D. Weiskopf, and M. Sedlmair. “Hagrid — Gridify Scatterplots with Hilbert and Gosper Curves”. In: *The 14th International Symposium on Visual Information Communication and Interaction*. Potsdam Germany: ACM, 2021, pp. 1–8. ISBN: 978-1-4503-8647-0. DOI: 10.1145/3481549.3481569 (cit. on p. 67).

[Dan+15] M. Dang, S. Lienhard, D. Ceylan, B. Neubert, P. Wonka, and M. Pauly. “Interactive Design of Probability Density Functions for Shape Grammars”. In: *ACM Transactions on Graphics* 34.6 (2015), pp. 1–13. DOI: 10.1145/2816795.2818069 (cit. on pp. 53, 55).

[Day+20] N. R. Dayama, K. Todi, T. Saarelainen, and A. Oulasvirta. “GRIDS: Interactive Layout Design with Integer Programming”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Honolulu HI USA: ACM, 2020, pp. 1–13. ISBN: 978-1-4503-6708-0. DOI: 10.1145/3313831.3376553 (cit. on pp. 45, 55).

[DFC08] U. Demšar, S. A. Fotheringham, and M. Charlton. “Combining Geovisual Analytics with Spatial Statistics: The Example of Geographically Weighted Regression”. In: *The Cartographic Journal* 45.3 (2008), pp. 182–192. DOI: 10.1179/000870408x311378 (cit. on p. 65).

[DH11] U. Demšar and P. Harris. “Visual Comparison of Moving-Window Kriging Models”. In: *Cartographica: The International Journal for Geographic Information and Geovisualization* 46.4 (2011), pp. 211–226. DOI: 10.3138/carto.46.4.211 (cit. on p. 65).

[Den96] B. D. Dent. *Cartography: Thematic Map Design*. 4th ed. Dubuque, IA: Wm. C. Brown, 1996. 434 pp. ISBN: 978-0-697-22970-0 (cit. on p. 65).

[Des+19] R. Desai, F. Anderson, J. Matejka, S. Coros, J. McCann, G. Fitzmaurice, and T. Grossman. “Geppetto: Enabling Semantic Design of Expressive Robot Behaviors”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Glasgow Scotland Uk: ACM, 2019, pp. 1–14. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300599 (cit. on pp. 45, 46).

[dBar+20] S. di Bartolomeo, A. Pandey, A. Leventidis, D. Saffo, U. H. Syeda, E. Carstensdottir, M. Seif El-Nasr, M. A. Borkin, and C. Dunne. “Evaluating the Effect of Timeline Shape on Visualization Task Performance”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Honolulu HI USA: ACM, 2020, pp. 1–12. ISBN: 978-1-4503-6708-0. DOI: 10.1145/3313831.3376237 (cit. on p. 62).

[Dor+15] H. Doraiswamy, N. Ferreira, M. Lage, H. Vo, L. Wilson, H. Werner, M. Park, and C. Silva. “Topology-Based Catalogue Exploration Framework for Identifying View-Enhanced Tower Designs”. In: *ACM Transactions on Graphics* 34.6 (2015), pp. 1–13. DOI: 10.1145/2816795.2818134 (cit. on pp. 47, 50, 55).

[Dör+12] M. Dörk, N. Henry Riche, G. Ramos, and S. Dumais. “PivotPaths: Strolling through Faceted Information Spaces”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), pp. 2709–2718. DOI: 10.1109/tvcg.2012.252 (cit. on p. 70).

[Dow+19] M. Dowling, J. Wenskovitch, J. Fry, S. Leman, L. House, and C. North. “SIRIUS: Dual, Symmetric, Interactive Dimension Reductions”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 172–182. DOI: 10.1109/tvcg.2018.2865047 (cit. on p. 75).

[DRS07] J. C. Duque, R. Ramos, and J. Suriñach. “Supervised Regionalization Methods: A Survey”. In: *International Regional Science Review* 30.3 (2007), pp. 195–220. DOI: 10.1177/0160017607301605 (cit. on p. 67).

[DB07] J. Dykes and C. Brunsdon. “Geographically Weighted Visualization: Interactive Graphics for Scale-Varying Exploratory Analysis”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1161–1168. DOI: 10.1109/tvcg.2007.70558 (cit. on p. 65).

[Dyr+00] C. Dyreson, W. Evans, H. Lin, and R. Snodgrass. “Efficiently Supporting Temporal Granularities”. In: *IEEE Transactions on Knowledge and Data Engineering* 12.4 (2000), pp. 568–587. DOI: 10.1109/69.868908 (cit. on p. 61).

[EST20] C. Eichner, H. Schumann, and C. Tominski. “Making Parameter Dependencies of Time-Series Segmentation Visually Understandable”. In: *Computer Graphics Forum* 39.1 (2020), pp. 607–622. DOI: 10.1111/cgf.13894 (cit. on pp. 42, 49, 52, 54).

[Elm13] M. E. Elmer. “Symbol Considerations for Bivariate Thematic Maps”. In: *Proceedings of the 26th International Cartographic Conference*. 2013. ISBN: 978-1-907075-06-3 (cit. on p. 65).

[End+11] A. Endert, C. Han, D. Maiti, L. House, S. Leman, and C. North. “Observation-Level Interaction with Statistical Models for Visual Analytics”. In: *IEEE Conference on Visual Analytics Science and Technology*. IEEE, 2011, pp. 121–130. DOI: 10.1109/vast.2011.6102449 (cit. on p. 75).

[Esp+23] M. Espadoto, G. Appleby, A. Suh, D. Cashman, M. Li, C. Scheidegger, E. W. Anderson, R. Chang, and A. C. Telea. “UnProjection: Leveraging Inverse-Projections for Visual Analytics of High-Dimensional Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 29.2 (2023), pp. 1559–1572. DOI: 10.1109/TVCG.2021.3125576 (cit. on p. 75).

[Esp+19] M. Espadoto, R. M. Martins, A. Kerren, N. S. T. Hirata, and A. C. Telea. “Towards a Quantitative Survey of Dimension Reduction Techniques”. In: *IEEE Transactions on Visualization and Computer Graphics* 27.3 (2019), pp. 2153–2173. DOI: 10.1109/tvcg.2019.2944182 (cit. on pp. 74–76).

[Ete+15] R. Etemadpour, R. Motta, J. G. d. S. Paiva, R. Minghim, M. C. F. de Oliveira, and L. Linsen. “Perception-Based Evaluation of Projection Methods for Multidimensional Data Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 21.1 (2015), pp. 81–94. DOI: 10.1109/tvcg.2014.2330617 (cit. on p. 75).

[FGS19] R. Faust, D. Glickenstein, and C. Scheidegger. “DimReader: Axis Lines That Explain Non-Linear Projections”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 481–490. DOI: 10.1109/tvcg.2018.2865194 (cit. on p. 75).

[FBW16] F. Ferstl, K. Bürger, and R. Westermann. “Streamline Variability Plots for Characterizing the Uncertainty in Vector Field Ensembles”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 767–776. DOI: 10.1109/tvcg.2015.2467204 (cit. on p. 73).

[FD13] P. Foley and U. Demšar. “Using Geovisual Analytics to Compare the Performance of Geographically Weighted Discriminant Analysis versus Its Global Counterpart, Linear Discriminant Analysis”. In: *International Journal of Geographical Information Science* 27.4 (2013), pp. 633–661. DOI: 10.1080/13658816.2012.722638 (cit. on p. 65).

[FMH16] B. Fröhler, T. Möller, and C. Heinzl. “GEMSe: Visualization-Guided Exploration of Multi-channel Segmentation Algorithms”. In: *Computer Graphics Forum* 35.3 (2016), pp. 191–200. DOI: 10.1111/cgf.12895 (cit. on pp. 47, 50, 51, 53–55).

[Fuc+13] J. Fuchs, F. Fischer, F. Mansmann, E. Bertini, and P. Isenberg. “Evaluation of Alternative Glyph Designs for Time Series Data in a Small Multiple Setting”. In: *Proceedings of the 2013 CHI Conference on Human Factors in Computing Systems*. Paris, France: ACM Press, 2013, p. 3237. ISBN: 978-1-4503-1899-0. DOI: 10.1145/2470654.2466443 (cit. on p. 62).

[Gah+02] M. Gahegan, M. Takatsuka, M. Wheeler, and F. Hardisty. “Introducing GeoVISTA Studio: An Integrated Suite of Visualization and Computational Methods for Exploration and Knowledge Construction in Geography”. In: *Computers, Environment and Urban Systems* 26.4 (2002), pp. 267–292. DOI: 10.1016/s0198-9715(01)00046-1 (cit. on p. 65).

[Gle13] M. Gleicher. “Explainers: Expert Explorations with Crafted Projections”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2042–2051. DOI: `10.1109/tvcg.2013.157` (cit. on p. 75).

[Gle18] M. Gleicher. “Considerations for Visualizing Comparison”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2018), pp. 413–423. DOI: `10.1109/tvcg.2017.2744199` (cit. on p. 71).

[Gle+11] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. “Visual Comparison for Information Visualization”. In: *Information Visualization* 10.4 (2011), pp. 289–309. DOI: `10.1177/1473871611416549` (cit. on pp. 48, 62, 71).

[GI12] G. Glen and K. Isaacs. “Estimating Sobol Sensitivity Indices Using Correlations”. In: *Environmental Modelling & Software* 37 (2012), pp. 157–166. DOI: `10.1016/j.envsoft.2012.03.014` (cit. on p. 49).

[Gog+19] A. Gogolou, T. Tsandilas, T. Palpanas, and A. Bezerianos. “Comparing Similarity Perception in Time Series Visualizations”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 523–533. DOI: `10.1109/tvcg.2018.2865077` (cit. on p. 62).

[Goo+16] S. Goodwin, J. Dykes, A. Slingsby, and C. Turkay. “Visualizing Multiple Variables Across Scale and Geography”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 599–608. DOI: `10.1109/tvcg.2015.2467199` (cit. on p. 65).

[GB20] M. Gröbe and D. Burghardt. “Micro Diagrams: Visualization of Categorical Point Data from Location-Based Social Media”. In: *Cartography and Geographic Information Science* 47.4 (2020), pp. 305–320. DOI: `10.1080/15230406.2020.1733438` (cit. on p. 68).

[GW11] G. Grolemund and H. Wickham. “Dates and Times Made Easy with lubridate”. In: *Journal of Statistical Software* 40.3 (2011), pp. 1–25 (cit. on p. 61).

[Gru+16] H. Gruendl, P. Riehmann, Y. Pausch, and B. Froehlich. “Time-Series Plots Integrated in Parallel-Coordinates Displays”. In: *Computer Graphics Forum* 35.3 (2016), pp. 321–330. DOI: `10.1111/cgf.12908` (cit. on p. 64).

[Gsc+16] T. Gschwandtner, M. Bögl, P. Federico, and S. Miksch. “Visual Encodings of Temporal Uncertainty: A Comparative User Study”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 539–548. DOI: `10.1109/tvcg.2015.2467752` (cit. on p. 61).

[GT16] T. Günther and H. Theisel. “Inertial Steady 2D Vector Field Topology”. In: *Computer Graphics Forum* 35.2 (2016), pp. 455–466. DOI: `10.1111/cgf.12846` (cit. on pp. 50, 73).

[Guo+14] X. Guo, J. Lin, K. Xu, and X. Jin. “Creature Grammar for Creative Modeling of 3D Monsters”. In: *Graphical Models* 76.5 (2014), pp. 376–389. DOI: 10.1016/j.gmod.2014.03.019 (cit. on p. 59).

[HHB08] M. Hahsler, K. Hornik, and C. Buchta. “Getting Things in Order: An Introduction to the R Package Seriation”. In: *Journal of Statistical Software* 25 (2008), pp. 1–34. DOI: 10.18637/jss.v025.i03 (cit. on p. 54).

[Ham94] D. M. Hamby. “A Review of Techniques for Parameter Sensitivity Analysis of Environmental Models”. In: *Environmental Monitoring and Assessment* 32.2 (1994), pp. 135–154. DOI: 10.1007/bf00547132 (cit. on pp. 40, 49).

[HKF16] S. Haroz, R. Kosara, and S. L. Franconeri. “The Connected Scatterplot for Presenting Paired Time Series”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.9 (2016), pp. 2174–2186. DOI: 10.1109/tvcg.2015.2502587 (cit. on p. 63).

[Has+91] J. Haslett, R. Bradley, P. Craig, and A. Unwin. “Dynamic Graphics for Exploring Spatial Data with Application to Locating Global and Local Anomalies”. In: *The American Statistician* 45.3 (1991) (cit. on p. 65).

[Haz+20] S. Hazarika, H. Li, K.-C. Wang, H.-W. Shen, and C.-S. Chou. “NNVA: Neural Network Assisted Visual Analysis of Yeast Cell Polarization Simulation”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2020), pp. 34–44. DOI: 10.1109/tvcg.2019.2934591 (cit. on pp. 40, 51, 55, 59).

[He+20] W. He, J. Wang, H. Guo, K.-C. Wang, H.-W. Shen, M. Raj, Y. S. G. Nashed, and T. Peterka. “InSituNet: Deep Image Synthesis for Parameter Space Exploration of Ensemble Simulations”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2020), pp. 23–33. DOI: 10.1109/tvcg.2019.2934312 (cit. on pp. 44, 48, 57).

[HG18] J. Holbein and T. Günther. “Parameter Space Comparison of Inertial Particle Models”. In: *Vision, Modeling and Visualization.* Ed. by F. Beck, C. Dachsbacher, and F. Sadlo. The Eurographics Association, 2018. ISBN: 978-3-03868-072-7. DOI: 10.2312/vmv.20181254 (cit. on p. 60).

[IL15] B. Iooss and P. Lemaître. “A Review on Global Sensitivity Analysis Methods”. In: *Uncertainty Management in Simulation-Optimization of Complex Systems: Algorithms and Applications.* Ed. by G. Dellino and C. Meloni. Vol. 59. Boston, MA: Springer US, 2015, pp. 101–122. ISBN: 978-1-4899-7547-8. DOI: 10.1007/978-1-4899-7547-8_5 (cit. on p. 40).

[JH20] J. Jankowai and I. Hotz. “Feature Level-Sets: Generalizing Iso-Surfaces to Multi-Variate Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.2 (2020), pp. 1308–1319. DOI: 10.1109/tvcg.2018.2867488 (cit. on p. 67).

[JM00] T. Jankun-Kelly and K.-L. Ma. “A Spreadsheet Interface for Visualization Exploration”. In: *Proceedings Visualization 2000. VIS 2000 (Cat. No.00CH37145)*. 2000, pp. 69–76. DOI: 10.1109/visual.2000.885678 (cit. on p. 56).

[JE12] W. Javed and N. Elmqvist. “Exploring the Design Space of Composite Visualization”. In: *2012 IEEE Pacific Visualization Symposium*. Songdo, Korea (South): IEEE, 2012, pp. 1–8. ISBN: 978-1-4673-0866-3 978-1-4673-0863-2 978-1-4673-0865-6. DOI: 10.1109/PacificVis.2012.6183556 (cit. on pp. 46, 48).

[JME10] W. Javed, B. McDonnel, and N. Elmqvist. “Graphical Perception of Multiple Time Series”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 927–934. DOI: 10.1109/tvcg.2010.162 (cit. on p. 63).

[Jeo+22] H. Jeon, H.-K. Ko, J. Jo, Y. Kim, and J. Seo. “Measuring and Explaining the Inter-Cluster Reliability of Multidimensional Projections”. In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2022), pp. 551–561. DOI: 10.1109/tvcg.2021.3114833 (cit. on p. 74).

[Jeo+09] D. H. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang. “iPCA: An Interactive System for PCA-based Visual Analytics”. In: *Computer Graphics Forum* 28.3 (2009), pp. 767–774. DOI: 10.1111/j.1467-8659.2009.01475.x (cit. on p. 76).

[Jo+19] J. Jo, F. Vernier, P. Dragicevic, and J.-D. Fekete. “A Declarative Rendering Model for Multiclass Density Maps”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 470–480. DOI: 10.1109/tvcg.2018.2865141 (cit. on p. 68).

[Kal+19] A. Kale, F. Nguyen, M. Kay, and J. Hullman. “Hypothetical Outcome Plots Help Untrained Observers Judge Trends in Ambiguous Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 892–902. DOI: 10.1109/tvcg.2018.2864909 (cit. on p. 71).

[Kaz+17] R. H. Kazi, T. Grossman, H. Cheong, A. Hashemi, and G. Fitzmaurice. “DreamSketch: Early Stage 3D Design Explorations with Sketching and Generative Design”. In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2017, pp. 401–414. ISBN: 978-1-4503-4981-9. DOI: 10.1145/3126594.3126662 (cit. on pp. 45, 59).

[Keh+22] R. Kehlbeck, J. Görtler, Y. Wang, and O. Deussen. “SPEULER: Semantics-preserving Euler Diagrams”. In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2022), pp. 433–442. DOI: 10.1109/tvcg.2021.3114834 (cit. on p. 69).

[KH13] J. Kehrer and H. Hauser. “Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.3 (2013), pp. 495–513. DOI: 10.1109/tvcg.2012.110 (cit. on p. 70).

[Kei00] D. Keim. “Designing Pixel-Oriented Visualization Techniques: Theory and Applications”. In: *IEEE Transactions on Visualization and Computer Graphics* 6.1 (2000), pp. 59–78. DOI: 10.1109/2945.841121 (cit. on pp. 47, 67).

[Kei+06] D. A. Keim, T. Nietzschmann, N. Schelwies, J. Schneidewind, T. Schreck, and H. Ziegler. “A Spectral Visualization System for Analyzing Financial Time Series Data”. In: *Proceedings of the Eighth Joint Eurographics / IEEE VGTC Conference on Visualization*. 2006, pp. 195–202. DOI: 10.5555/2384796.2384826 (cit. on p. 62).

[KP10] W. B. Kerr and F. Pellacini. “Toward Evaluating Material Design Interface Paradigms for Novice Users”. In: *ACM Transactions on Graphics* 29.4 (2010), pp. 1–10. DOI: 10.1145/1778765.1778772 (cit. on p. 44).

[KGS19] S. Khan, E. Gunpinar, and B. Sener. “GenYacht: An Interactive Generative Design System for Computer-Aided Yacht Hull Design”. In: *Ocean Engineering* 191 (2019), p. 106462. DOI: 10.1016/j.oceaneng.2019.106462 (cit. on pp. 56, 58).

[Kim+16] H. Kim, J. Choo, H. Park, and A. Endert. “InterAxis: Steering Scatterplot Axes via Observation-Level Interaction”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 131–140. DOI: 10.1109/tvcg.2015.2467615 (cit. on p. 75).

[Kon+14] A. Konev, J. Waser, B. Sadransky, D. Cornel, R. A. P. Perdigão, Z. Horváth, and M. E. Gröller. “Run Watchers: Automatic Simulation-Based Decision Support in Flood Management”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1873–1882. DOI: 10.1109/tvcg.2014.2346930 (cit. on p. 59).

[Köt+15] P. Köthur, C. Witt, M. Sips, N. Marwan, S. Schinkel, and D. Dransch. “Visual Analytics for Correlation-Based Comparison of Time Series Ensembles”. In: *Computer Graphics Forum* 34.3 (2015), pp. 411–420. DOI: 10.1111/cgf.12653 (cit. on pp. 72, 73).

[KSI14] Y. Koyama, D. Sakamoto, and T. Igarashi. “Crowd-Powered Parameter Analysis for Visual Design Exploration”. In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. Honolulu Hawaii USA: ACM, 2014, pp. 65–74. ISBN: 978-1-4503-3069-5. DOI: 10.1145/2642918.2647386 (cit. on pp. 45, 46).

[KSG20] Y. Koyama, I. Sato, and M. Goto. “Sequential Gallery for Interactive Visual Design Optimization”. In: *ACM Transactions on Graphics* 39.4 (2020). DOI: 10.1145/3386569.3392444 (cit. on pp. 45, 56, 58).

[Koy+17] Y. Koyama, I. Sato, D. Sakamoto, and T. Igarashi. “Sequential Line Search for Efficient Visual Design Optimization by Crowds”. In: *ACM Transactions on Graphics* 36.4 (2017), pp. 1–11. DOI: `10.1145/3072959.3073598` (cit. on pp. 45, 56, 58).

[KO21] M.-J. Kraak and F. Ormeling. *Cartography: Visualization of Geospatial Data*. Fourth Edition. CRC Press, 2021. ISBN: 978-0-367-50774-9 (cit. on p. 65).

[Lam+12] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. “Empirical Studies in Information Visualization: Seven Scenarios”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.9 (2012), pp. 1520–1536. DOI: `10.1109/tvcg.2011.279` (cit. on p. 60).

[LT20] J.-B. Lamy and R. Tsopra. “RainBio: Proportional Visualization of Large Sets in Biology”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.11 (2020), pp. 3285–3298. DOI: `10.1109/tvcg.2019.2921544` (cit. on p. 69).

[LSN04] J. S. Lehman, T. J. Santner, and W. I. Notz. “Designing Computer Experiments to Determine Robust Control Variables”. In: *Statistica Sinica* 14.2 (2004), pp. 571–590 (cit. on p. 39).

[LA11] S. Lespinats and M. Aupetit. “CheckViz: Sanity Check and Topological Clues for Linear and Non-Linear Mappings”. In: *Computer Graphics Forum* 30.1 (2011), pp. 113–125. DOI: `10.1111/j.1467-8659.2010.01835.x` (cit. on p. 74).

[Lex+14] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, and H. Pfister. “UpSet: Visualization of Intersecting Sets”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1983–1992. DOI: `j` (cit. on p. 70).

[Lii10] I. Liiv. “Seriation and Matrix Reordering Methods: An Historical Overview”. In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 3.2 (2010), pp. 70–91. DOI: `10.1002/sam.10071` (cit. on p. 53).

[Lin+13] W.-C. Lin, T.-S. Huang, T.-C. Ho, Y.-T. Chen, and J.-H. Chuang. “Interactive Lighting Design with Hierarchical Light Representation”. In: *Computer Graphics Forum* 32.4 (2013), pp. 133–142. DOI: `10.1111/cgf.12159` (cit. on p. 45).

[Liu+18] J. Liu, T. Dwyer, K. Marriott, J. Millar, and A. Haworth. “Understanding the Relationship Between Interactive Optimisation and Visual Analytics in the Context of Prostate Brachytherapy”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2018), pp. 319–329. DOI: `10.1109/tvcg.2017.2744418` (cit. on p. 40).

[Liu+21] J. Liu, T. Dwyer, G. Tack, S. Gratzl, and K. Marriott. “Supporting the Problem-Solving Loop: Designing Highly Interactive Optimisation Systems”. In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (2021), pp. 1764–1774. DOI: 10.1109/tvcg.2020.3030364 (cit. on pp. 40, 46, 53, 56).

[Liu+17] L. Liu, A. P. Boone, I. T. Ruginski, L. Padilla, M. Hegarty, S. H. Creem-Regehr, W. B. Thompson, C. Yuksel, and D. H. House. “Uncertainty Visualization by Representative Sampling from Prediction Ensembles”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.9 (2017), pp. 2165–2178. DOI: 10.1109/tvcg.2016.2607204 (cit. on pp. 70, 71).

[Liu+19] L. Liu, L. Padilla, S. H. Creem-Regehr, and D. H. House. “Visualizing Uncertain Tropical Cyclone Predictions Using Representative Samples from Ensembles of Forecast Tracks”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 882–891. DOI: 10.1109/tvcg.2018.2865193 (cit. on p. 71).

[LV22] L. Liu and R. Vuillemot. “GROUPSET: A Set-Based Technique to Explore Time-Varying Data”. In: *EuroVis 2022 - Short Papers*. The Eurographics Association, 2022. DOI: 10.2312/evs.20221096 (cit. on p. 69).

[Lub+15] M. Luboschik, M. Röhlig, A. T. Bittig, N. Andrienko, H. Schumann, and C. Tominski. “Feature-Driven Visual Analytics of Chaotic Parameter-Dependent Movement”. In: *Computer Graphics Forum* 34.3 (2015), pp. 421–430. DOI: 10.1111/cgf.12654 (cit. on pp. 52, 54).

[Lub+12] M. Luboschik, S. Rybacki, R. Ewald, B. Schwarze, H. Schumann, and A. M. Uhrmacher. “Interactive Visual Exploration of Simulator Accuracy: A Case Study for Stochastic Simulation Algorithms”. In: *Proceedings of the 2012 Winter Simulation Conference (WSC)*. 2012, pp. 1–12. DOI: 10.1109/wsc.2012.6465190 (cit. on pp. 52, 54).

[Lub+14] M. Luboschik, S. Rybacki, F. Haack, and H.-J. Schulz. “Supporting the Integrated Visual Analysis of Input Parameters and Simulation Trajectories”. In: *Computers & Graphics* 39 (2014), pp. 37–47. DOI: 10.1016/j.cag.2013.09.004 (cit. on p. 54).

[MM20] Y. Ma and R. Maciejewski. “Visual Analysis of Class Separations with Locally Linear Segments”. In: *IEEE Transactions on Visualization and Computer Graphics* (2020), pp. 1–1. DOI: 10.1109/tvcg.2020.3011155 (cit. on p. 75).

[MHG10] M. M. Malik, C. Heinzl, and M. E. Groeller. “Comparative Visualization for Parameter Studies of Dataset Series”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.5 (2010), pp. 829–840. DOI: 10.1109/tvcg.2010.20 (cit. on pp. 48, 52, 53).

[Mar13] X. Marsault. "A Multiobjective and Interactive Genetic Algorithm to Optimize the Building Form in Early Design Stages". In: *Proceedings of BS2013: 13th Conference of IBPSA*. Ed. by E. Wurtz. Toronto, Canada: International Building Performance Simulation Association, 2013. ISBN: 978-2-7466-6294-0 (cit. on pp. 41, 58).

[Mat+18] J. Matejka, M. Glueck, E. Bradner, A. Hashemi, T. Grossman, and G. Fitzmaurice. "Dream Lens: Exploration and Visualization of Large-Scale Generative Design Datasets". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Montreal QC, Canada: ACM Press, 2018, pp. 1–12. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173943 (cit. on pp. 47, 50, 51, 53).

[Mat+10] K. Matković, D. Gračanin, M. Jelović, A. Ammer, A. Lež, and H. Hauser. "Interactive Visual Analysis of Multiple Simulation Runs Using the Simulation Model View: Understanding and Tuning of an Electronic Unit Injector". In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 1449–1457. DOI: 10.1109/tvcg.2010.171 (cit. on pp. 50–52, 55).

[Mat+14] K. Matković, D. Gračanin, R. Splechtna, M. Jelović, B. Stehno, H. Hauser, and W. Purgathofer. "Visual Analytics for Complex Engineering Systems: Hybrid Visual Steering of Simulation Ensembles". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1803–1812. DOI: 10.1109/tvcg.2014.2346744 (cit. on pp. 40, 50, 51, 55).

[Mat+13] K. Matković, M. Duras, D. Gračanin, R. Splechtna, B. Stehno, and H. Hauser. "Interactive Visual Analysis in the Concept Stage of a Hybrid-Vehicle Design". In: *EuroVis Workshop on Visual Analytics* (2013), 5 pages. DOI: 10.2312/pe.eurovast.eurova13.061-065 (cit. on p. 50).

[Mat+08] K. Matković, W. Freiler, D. Gračanin, and H. Hauser. "ComVis: A Coordinated Multiple Views System for Prototyping New Visualization Technology". In: *2008 12th International Conference Information Visualisation*. 2008, pp. 215–220. DOI: 10.1109/iv.2008.87 (cit. on p. 50).

[Mat+17] K. Matković, H. Abraham, M. Jelović, and H. Hauser. "Quantitative Externalization of Visual Data Analysis Results Using Local Regression Models". In: *1st International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Ed. by A. Holzinger, P. Kieseberg, A. M. Tjoa, and E. Weippl. Vol. 10410. Cham: Springer International Publishing, 2017, pp. 199–218. ISBN: 978-3-319-66807-9 978-3-319-66808-6. DOI: 10.1007/978-3-319-66808-6_14 (cit. on pp. 50, 55).

[MG13] A. Mayorga and M. Gleicher. "Splatterplots: Overcoming Overdraw in Scatter Plots". In: *IEEE Transactions on Visualization and Computer Graphics* 19.9 (2013), pp. 1526–1538. DOI: 10.1109/tvcg.2013.65 (cit. on p. 68).

[MHM18] L. McInnes, J. Healy, and J. Melville. “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction”. In: *arXiv* (2018). DOI: 10.48550/arXiv.1802.0342 (cit. on p. 75).

[McL+08] P. McLachlan, T. Munzner, E. Koutsofios, and S. North. “LiveRAC: Interactive Visual Exploration of System Management Time-series Data”. In: *Proceedings of the 2008 CHI Conference on Human Factors in Computing Systems* (Florence, Italy). New York, NY, USA: ACM, 2008, pp. 1483–1492. ISBN: 978-1-60558-011-1. DOI: 10.1145/1357054.1357286 (cit. on p. 62).

[ML19] L. McNabb and R. S. Laramee. “Multivariate Maps—A Glyph-Placement Algorithm to Support Multivariate Geospatial Visualization”. In: *Information* 10.10 (2019), p. 302. DOI: 10.3390/info10100302 (cit. on p. 67).

[Mer+11] P. Merrell, E. Schkufza, Z. Li, M. Agrawala, and V. Koltun. “Interactive Furniture Layout Using Interior Design Guidelines”. In: *ACM SIGGRAPH Conference*. Vancouver, British Columbia, Canada: ACM Press, 2011, p. 1. ISBN: 978-1-4503-0943-1. DOI: 10.1145/1964921.1964982 (cit. on p. 45).

[Meu19] W. Meulemans. “Efficient Optimal Overlap Removal: Algorithms and Experiments”. In: *Computer Graphics Forum* 38.3 (2019), pp. 713–723. DOI: 10.1111/cgf.13722 (cit. on p. 67).

[MSS21] W. Meulemans, M. Sondag, and B. Speckmann. “A Simple Pipeline for Coherent Grid Maps”. In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (2021), pp. 1236–1246. DOI: 10.1109/tvcg.2020.3028953 (cit. on p. 65).

[Meu+13] W. Meulemans, N. H. Riche, B. Speckmann, B. Alper, and T. Dwyer. “KelpFusion: A Hybrid Set Visualization Technique”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.11 (2013), pp. 1846–1858. DOI: 10.1109/tvcg.2013.76 (cit. on p. 70).

[MW20] A. Mohiuddin and R. Woodbury. “Interactive Parallel Coordinates for Parametric Design Space Exploration”. In: *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. Honolulu HI USA: ACM, 2020, pp. 1–9. ISBN: 978-1-4503-6819-3. DOI: 10.1145/3334480.3383101 (cit. on pp. 44, 45, 59).

[Mor+23] C. Morariu, A. Bibal, R. Cutura, B. Frénay, and M. Sedlmair. “Predicting User Preferences of Dimensionality Reduction Embedding Quality”. In: *IEEE Transactions on Visualization and Computer Graphics* 29.1 (2023), pp. 745–755. DOI: 10.1109/tvcg.2022.3209449 (cit. on pp. 74, 75).

[Mue14] I. Muehlenhaus. *Web Cartography: Map Design for Interactive and Mobile Devices*. Boca Raton, Fla.: CRC Press, 2014. 240 pp. ISBN: 978-1-4398-7622-0 (cit. on p. 65).

[Ngu+16] P. H. Nguyen, K. Xu, R. Walker, and B. W. Wong. “TimeSets: Timeline Visualization with Set Relations”. In: *Information Visualization* 15.3 (2016), pp. 253–269. DOI: 10.1177/1473871615605347 (cit. on p. 69).

[Nic+22] S. Nickel, M. Sondag, W. Meulemans, S. Kobourov, J. Peltonen, and M. Nöllenburg. “Multicriteria Optimization for Dynamic Demers Cartograms”. In: *IEEE Transactions on Visualization and Computer Graphics* 28.6 (2022), pp. 2376–2387. DOI: 10.1109/tvcg.2022.3151227 (cit. on p. 65).

[NA19] L. G. Nonato and M. Aupetit. “Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Enrichment”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.8 (2019), pp. 2650–2673. DOI: 10.1109/tvcg.2018.2846735 (cit. on p. 74).

[Nus+18] S. Nusrat, M. J. Alam, C. Scheidegger, and S. Kobourov. “Cartogram Visualization for Bivariate Geo-Statistical Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.10 (2018), pp. 2675–2688. DOI: 10.1109/tvcg.2017.2765330 (cit. on p. 65).

[NK16] S. Nusrat and S. Kobourov. “The State of the Art in Cartograms”. In: *Computer Graphics Forum* 35.3 (2016), pp. 619–642. DOI: 10.1111/cgf.12932 (cit. on p. 65).

[OAH15] P. O’Donovan, A. Agarwala, and A. Hertzmann. “DesignScape: Design with Interactive Layout Suggestions”. In: *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems*. Seoul Republic of Korea: ACM, 2015, pp. 1221–1224. ISBN: 978-1-4503-3145-6. DOI: 10.1145/2702123.2702149 (cit. on pp. 45, 55).

[OBJ16] H. Obermaier, K. Bensema, and K. I. Joy. “Visual Trends Analysis in Time-Varying Ensembles”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.10 (2016), pp. 2331–2342. DOI: 10.1109/tvcg.2015.2507592 (cit. on pp. 50, 52, 54, 73).

[Opa+19] T. Opach, J. Korycka-Skorupa, I. Karsznia, T. Nowacki, I. Gołębiowska, and J. K. Rød. “Visual Clutter Reduction in Zoomable Proportional Point Symbol Maps”. In: *Cartography and Geographic Information Science* 46.4 (2019), pp. 347–367. DOI: 10.1080/15230406.2018.1490202 (cit. on p. 67).

[Opa+18] T. Opach, S. Popelka, J. Dolezalova, and J. K. Rød. “Star and Polyline Glyphs in a Grid Plot and on a Map Display: Which Perform Better?” In: *Cartography and Geographic Information Science* 45.5 (2018), pp. 400–419. DOI: 10.1080/15230406.2017.1364169 (cit. on pp. 65, 66).

[Orb+19] D. Orban, D. F. Keefe, A. Biswas, J. Ahrens, and D. Rogers. “Drag and Track: A Direct Manipulation Interface for Contextualizing Data Instances within a Continuous Parameter Space”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 256–266. DOI: 10.1109/tvcg.2018.2865051 (cit. on pp. 46, 50, 73).

[Pae+23] P. Paetzold, R. Kehlbeck, H. Strobelt, Y. Xue, S. Storandt, and O. Deussen. “RectEuler: Visualizing Intersecting Sets Using Rectangles”. In: *Computer Graphics Forum* 42.3 (2023). DOI: 10.1111/cgf.14814 (cit. on p. 70).

[PP10a] Y. Park and J. Park. “Disk Diagram: An Interactive Visualization Technique of Fuzzy Set Operations for the Analysis of Fuzzy Data”. In: *Information Visualization* 9.3 (2010), pp. 220–232. DOI: 10.1057/ivs.2010.3 (cit. on p. 70).

[PP10b] Y. Park and J. Park. “Interactive Visualization of Fuzzy Set Operations”. In: *Visualization and Data Analysis*. Ed. by J. Park, M. C. Hao, P. C. Wong, and C. Chen. Vol. 753002. San Jose, California, 2010. DOI: 10.1117/12.840229 (cit. on p. 70).

[Pér+22] I. Pérez-Messina, D. Ceneda, M. El-Assady, S. Miksch, and F. Sperrle. “A Typology of Guidance Tasks in Mixed-Initiative Visual Analytics Environments”. In: *Computer Graphics Forum* 41.3 (2022), pp. 465–476. DOI: 10.1111/cgf.14555 (cit. on pp. 57, 77).

[PBM23a] N. Piccolotto, M. Bögl, and S. Miksch. “Multi-Ensemble Visual Analytics via Fuzzy Sets”. In: *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, 2023. ISBN: 978-3-03868-222-6. DOI: 10.2312/eurova.20231092 (cit. on p. 70).

[PBM23b] N. Piccolotto, M. Bögl, and S. Miksch. “Visual Parameter Space Exploration in Time and Space”. In: *Computer Graphics Forum* 42.6 (2023), e14785. DOI: 10.1111/cgf.14785 (cit. on pp. 42, 57).

[Poc+14] J. Poco, A. Dasgupta, Y. Wei, W. Hargrove, C. R. Schwalm, D. N. Huntzinger, R. Cook, E. Bertini, and C. T. Silva. “Visual Reconciliation of Alternative Similarity Spaces in Climate Modeling”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1923–1932. DOI: 10.1109/tvcg.2014.2346755 (cit. on p. 54).

[Pre+11] A. J. Pretorius, M. Bray, A. E. Carpenter, and R. A. Ruddle. “Visualization of Parameter Space for Image Analysis”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2402–2411. DOI: 10.1109/tvcg.2011.253 (cit. on pp. 54, 56).

[PZR15] A. Pretorius, Y. Zhou, and R. Ruddle. “Visual Parameter Optimisation for Biomedical Image Processing”. In: *BMC Bioinformatics* 16.S11 (2015), S9. DOI: 10.1186/1471-2105-16-s11-s9 (cit. on pp. 51, 54).

[Rai+16] R. G. Raidou, O. Casares-Magaz, L. P. Muren, U. A. van der Heide, J. Rørvik, M. Breeuwer, and A. Vilanova. “Visual Analysis of Tumor Control Models for Prediction of Radiotherapy Response”. In: *Computer Graphics Forum* 35.3 (2016), pp. 231–240. DOI: 10.1111/cgf.12899 (cit. on p. 53).

[Rai+14] R. G. Raidou, M. Breeuwer, A. Vilanova, U. A. V. D. Heide, and P. J. V. Houdt. “The iCoCooN: Integration of Cobweb Charts with Parallel Coordinates for Visual Analysis of DCE-MRI Modeling Variations”. In: *Eurographics Workshop on Visual Computing for Biology and Medicine* (2014), 10 pages. DOI: 10.2312/vcbm.20141178 (cit. on pp. 47, 59).

[RSG21] F. Raith, G. Scheuermann, and C. Gillmann. “Uncertainty-Aware Detection and Visualization of Ocean Eddies in Ensemble Flow Fields - A Case Study of the Red Sea”. In: *Workshop on Visualisation in Environmental Sciences (EnvirVis)*. The Eurographics Association, 2021. ISBN: 978-3-03868-148-9. DOI: 10.2312/envirvis.20211080 (cit. on p. 48).

[RKW20] C. Reinbold, A. Kumpf, and R. Westermann. “Visualizing the Stability of 2D Point Sets from Dimensionality Reduction Techniques”. In: *Computer Graphics Forum* 39.1 (2020), pp. 333–346. DOI: 10.1111/cgf.13806 (cit. on p. 74).

[RPI19] A. Ribés, J. Pouderoux, and B. Iooss. “A Visual Sensitivity Analysis for Parameter-Augmented Ensembles of Curves”. In: *Journal of Verification, Validation and Uncertainty Quantification* 4.4 (2019). DOI: 10.1115/1.4046020 (cit. on pp. 50, 53).

[Rib+13] H. Ribičić, J. Waser, R. Fuchs, G. Blöschl, and E. Gröller. “Visual Analysis and Steering of Flooding Simulations”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.6 (2013), pp. 1062–1075. DOI: 10.1109/tvcg.2012.175 (cit. on pp. 50, 51, 53, 54).

[RSC15] P. Rodgers, G. Stapleton, and P. Chapman. “Visualizing Sets with Linear Diagrams”. In: *ACM Transactions on Computer-Human Interaction* 22.6 (2015), pp. 1–39. DOI: 10.1145/2810012 (cit. on p. 69).

[Röh+15] M. Röhlig, M. Luboschik, F. Krüger, T. Kirste, H. Schumann, M. Bögl, B. Alsallakh, and S. Miksch. “Supporting Activity Recognition by Visual Analytics”. In: *IEEE Conference on Visual Analytics Science and Technology*. IEEE, 2015, pp. 41–48. DOI: 10.1109/vast.2015.7347629 (cit. on pp. 52, 54).

[RGG18] I. B. Rojo, M. Gross, and T. Günther. “Visualizing the Phase Space of Heterogeneous Inertial Particles in 2D Flows”. In: *Computer Graphics Forum* 37.3 (2018), pp. 289–300. DOI: 10.1111/cgf.13420 (cit. on p. 53).

[Rup+14] T. Ruppert, J. Bernard, A. Ulmer, H. Lücke-Tieke, and J. Kohlhammer. "Visual Access to an Agent-Based Simulation Model to Support Political Decision Making". In: *Proceedings of the 14th International Conference on Knowledge Technologies and Data-driven Business - i-KNOW '14*. Graz, Austria: ACM Press, 2014, pp. 1–8. ISBN: 978-1-4503-2769-5. DOI: `10.1145/2637748.2638410` (cit. on p. 44).

[Sac+17] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. "Visual Interaction with Dimensionality Reduction: A Structured Literature Analysis". In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 241–250. DOI: `10.1109/tvcg.2016.2598495` (cit. on p. 74).

[Sac+14] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim. "Knowledge Generation Model for Visual Analytics". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1604–1613. DOI: `10.1109/tvcg.2014.2346481` (cit. on pp. 42, 50).

[Sad+14] R. Sadana, T. Major, A. Dove, and J. Stasko. "OnSet: A Visualization Technique for Large-scale Binary Set Data". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1993–2002. DOI: `10.1109/tvcg.2014.2346249` (cit. on p. 70).

[Sag+17] A. Sagristà, S. Jordan, A. Just, F. Dias, L. G. Nonato, and F. Sadlo. "Topological Analysis of Inertial Dynamics". In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 950–959. DOI: `10.1109/tvcg.2016.2599018` (cit. on pp. 50, 53).

[SJS20] A. Sagristà, S. Jordan, and F. Sadlo. "Visual Analysis of the Finite-Time Lyapunov Exponent". In: *Computer Graphics Forum* 39.3 (2020), pp. 331–342. DOI: `10.1111/cgf.13984` (cit. on p. 52).

[Sai+05] T. Saito, H. Miyamura, M. Yamamoto, H. Saito, Y. Hoshiya, and T. Kaseda. "Two-Tone Pseudo Coloring: Compact Visualization for One-Dimensional Data". In: *IEEE Symposium on Information Visualization*. 2005, pp. 173–180. DOI: `10.1109/infvis.2005.1532144` (cit. on p. 62).

[SWN03] T. J. Santner, B. J. Williams, and W. I. Notz. *The Design and Analysis of Computer Experiments*. First edition. New York, NY: Springer, 2003. ISBN: 0-387-95420-1 (cit. on p. 39).

[SK13] T. Schultz and G. L. Kindlmann. "Open-Box Spectral Clustering: Applications to Medical Image Analysis". In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2100–2108. DOI: `10.1109/tvcg.2013.181` (cit. on pp. 51, 56, 58).

[Sch+18] A. Schulz, H. Wang, E. Grinspun, J. Solomon, and W. Matusik. "Interactive Exploration of Design Trade-Offs". In: *ACM Transactions on Graphics* 37.4 (2018), 131:1–131:14. DOI: `10.1145/3197517.3201385` (cit. on pp. 48, 54).

[Sch+17] A. Schulz, J. Xu, B. Zhu, C. Zheng, E. Grinspun, and W. Matusik. "Interactive Design Space Exploration and Optimization for CAD Models". In: *ACM Transactions on Graphics* 36.4 (2017), pp. 1–14. DOI: 10.1145/3072959.3073688 (cit. on p. 44).

[Sch+19] M. Schwarzl, L. Autin, G. Johnson, T. Torsney-Weir, and T. Möller. "Cellpackexplorer: Interactive Model Building for Volumetric Data of Complex Cells". In: *Computers & Graphics: X* 2 (2019), p. 100010. DOI: 10.1016/j.cagx.2019.100010 (cit. on p. 53).

[Sed+14] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. "Visual Parameter Space Analysis: A Conceptual Framework". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2161–2170. DOI: 10.1109/tvcg.2014.2346321 (cit. on pp. 39–43, 56–58, 60, 72).

[Sel+18] J. Z. Self, M. Dowling, J. Wenskovitch, I. Crandell, M. Wang, L. House, S. Leman, and C. North. "Observation-Level and Parametric Interaction for High-Dimensional Data Analysis". In: *ACM Transactions on Interactive Intelligent Systems* 8.2 (2018), pp. 1–36. DOI: 10.1145/3158230 (cit. on p. 75).

[Shn83] B. Shneiderman. "Direct Manipulation: A Step Beyond Programming Languages". In: *Computer* 16.8 (1983), pp. 57–69. DOI: 10.1109/mc.1983.1654471 (cit. on p. 44).

[Sil+15] R. R. O. da Silva, P. E. Rauber, R. M. Martins, R. Minghim, and A. C. Telea. "Attribute-Based Visual Explanation of Multidimensional Projections". In: *EuroVis Workshop on Visual Analytics (EuroVA)* (2015), 5 pages. DOI: 10.2312/eurova.20151100 (cit. on p. 76).

[Soh+22] J.-T. Sohns, M. Schmitt, F. Jirasek, H. Hasse, and H. Leitte. "Attribute-Based Explanation of Non-Linear Embeddings of High-Dimensional Data". In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2022), pp. 540–550. DOI: 10.1109/tvcg.2021.3114870 (cit. on p. 76).

[Sor+16] J. Sorger, T. Ortner, C. Luksch, M. Schwärzler, E. Gröller, and H. Piringer. "LiteVis: Integrated Visualization for Simulation-Based Decision Support in Lighting Design". In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 290–299. DOI: 10.1109/tvcg.2015.2468011 (cit. on pp. 50, 51, 54).

[Spe+21] F. Sperrle, A. Jeitler, J. Bernard, D. Keim, and M. El-Assady. "Co-Adaptive Visual Data Analysis and Guidance Processes". In: *Computers & Graphics* 100 (2021), pp. 93–105. DOI: 10.1016/j.cag.2021.06.016 (cit. on p. 77).

[Spe+20] F. Sperrle, A. Jeitler, J. Bernard, D. A. Keim, and M. El-Assady. "Learning and Teaching in Co-Adaptive Guidance for Mixed-Initiative Visual Analytics". In: *EuroVis Workshop on Visual Analytics (EuroVA)*. Ed. by C. Turkay and K. Vrotsou. The Eurographics Association, 2020. ISBN: 978-3-03868-116-8. DOI: 10.2312/eurova.20201088 (cit. on p. 56).

[Sta+16] J. Stahnke, M. Dörk, B. Müller, and A. Thom. "Probing Projections: Interaction Techniques for Interpreting Arrangements and Errors of Dimensionality Reductions". In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 629–638. DOI: 10.1109/tvcg.2015.2467717 (cit. on pp. 74, 76).

[Ste+13] C. A. Steed, D. M. Ricciuto, G. Shipman, B. Smith, P. E. Thornton, D. Wang, X. Shi, and D. N. Williams. "Big Data Visual Analytics for Exploratory Earth System Simulation Analysis". In: *Computers & Geosciences* 61 (2013), pp. 71–82. DOI: 10.1016/j.cageo.2013.07.025 (cit. on pp. 47, 50).

[Ste+17] B. Steiner, E. Mousavian, F. M. Saradj, M. Wimmer, and P. Musialski. "Integrated Structural–Architectural Design for Interactive Planning". In: *Computer Graphics Forum* 36.8 (2017), pp. 80–94. DOI: 10.1111/cgf.12996 (cit. on p. 45).

[Sti+16] H. Stitz, S. Gratzl, W. Aigner, and M. Streit. "ThermalPlot: Visualizing Multi-Attribute Time-Series Data Using a Thermal Metaphor". In: *IEEE Transactions on Visualization and Computer Graphics* 22.12 (2016), pp. 2594–2607. DOI: 10.1109/tvcg.2015.2513389 (cit. on p. 64).

[Sus+21] J. Suschnigg, B. Mutlu, G. Koutroulis, V. Sabol, S. Thalmann, and T. Schreck. "Visual Exploration of Anomalies in Cyclic Time Series Data with Matrix and Glyph Representations". In: *Big Data Research* 26 (2021), p. 100251. DOI: 10.1016/j.bdr.2021.100251 (cit. on p. 63).

[Swe+20] A. Swearngin, C. Wang, A. Oleson, J. Fogarty, and A. J. Ko. "Scout: Rapid Exploration of Interface Layout Alternatives through High-Level Design Constraints". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Honolulu HI USA: ACM, 2020, pp. 1–13. ISBN: 978-1-4503-6708-0. DOI: 10.1145/3313831.3376593 (cit. on pp. 50, 55).

[Tob70] W. R. Tobler. "A Computer Movie Simulating Urban Growth in the Detroit Region". In: *Economic Geography* 46 (1970), p. 234. DOI: 10.2307/143141 (cit. on p. 41).

[TAS04] C. Tominski, J. Abello, and H. Schumann. "Axes-Based Visualizations with Radial Layouts". In: *Proceedings of the 2004 ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2004, pp. 1242–1247. ISBN: 978-1-58113-812-2. DOI: 10.1145/967900.968153 (cit. on p. 64).

[Tom+23] C. Tominski, M. Behrisch, S. Bleisch, S. I. Fabrikant, E. Mayr, S. Miksch, and H. Purchase. “Visualizing Uncertainty in Sets”. In: *IEEE Computer Graphics and Applications* 43.5 (2023). DOI: 10.1109/mcg.2023.3300441 (cit. on p. 70).

[TML18] C. Tong, L. McNabb, and R. S. Laramee. “Cartograms with Topological Features”. In: *Computer Graphics and Visual Computing (CGVC)* (2018), 8 pages. DOI: 10.2312/cgvc.20181217 (cit. on p. 65).

[Tor+11] T. Torsney-Weir, A. Saad, T. Möller, H.-C. Hege, B. Weber, J.-M. Verbavatz, and S. Bergner. “Tuner: Principled Parameter Finding for Image Segmentation Algorithms Using Visual Response Surface Exploration”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 1892–1901. DOI: 10.1109/tvcg.2011.248 (cit. on pp. 43, 51, 55, 56).

[Tur+14] C. Turkay, A. Slingsby, H. Hauser, J. Wood, and J. Dykes. “Attribute Signatures: Dynamic Visual Summaries for Analyzing Multivariate Geographical Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2033–2042. DOI: 10.1109/tvcg.2014.2346265 (cit. on p. 66).

[UIM12] N. Umetani, T. Igarashi, and N. J. Mitra. “Guided Exploration of Physically Valid Shapes for Furniture Design”. In: *ACM Transactions on Graphics* 31.4 (2012), pp. 1–11. DOI: 10.1145/2185520.2185582 (cit. on p. 45).

[Ume+14] N. Umetani, Y. Koyama, R. Schmidt, and T. Igarashi. “Pteromys: Interactive Design and Optimization of Free-Formed Free-Flight Model Airplanes”. In: *ACM Transactions on Graphics* 33.4 (2014), pp. 1–10. DOI: 10.1145/2601097.2601129 (cit. on pp. 45, 48).

[Ung+12] A. Unger, S. Schulte, V. Klemann, and D. Dransch. “A Visual Analysis Concept for the Validation of Geoscientific Simulation Models”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), pp. 2216–2225. DOI: 10.1109/tvcg.2012.190 (cit. on p. 52).

[vLFR17] T. von Landesberger, D. W. Fellner, and R. A. Ruddle. “Visualization System Requirements for Data Processing Pipeline Design and Optimization”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.8 (2017), pp. 2028–2041. DOI: 10.1109/tvcg.2016.2603178 (cit. on p. 43).

[vLan+13] T. von Landesberger, S. Bremm, M. Kirschner, S. Wesarg, and A. Kuijper. “Visual Analytics for Model-Based Medical Image Segmentation: Opportunities and Challenges”. In: *Expert Systems with Applications* 40.12 (2013), pp. 4934–4943. DOI: 10.1016/j.eswa.2013.03.006 (cit. on pp. 53, 55).

[Wac03] H. Wackernagel. *Multivariate geostatistics: an introduction with applications.* Berlin, Heidelberg: Springer Science & Business Media, 2003. DOI: 10.1007/978-3-662-05294-5 (cit. on p. 65).

[Wal+20] A. Walch, M. Schwärzler, C. Luksch, E. Eisemann, and T. Gschwandtner. "LightGuider: Guiding Interactive Lighting Design Using Suggestions, Provenance, and Quality Visualization". In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2020), pp. 569–578. DOI: 10.1109/tvcg.2019.2934658 (cit. on pp. 45, 51, 56).

[WDN23] M. Wallinger, A. Dobler, and M. Nollenburg. "LinSets.Zip: Compressing Linear Set Diagrams". In: *IEEE Transactions on Visualization and Computer Graphics* (2023), pp. 1–12. DOI: 10.1109/tvcg.2023.3261934 (cit. on p. 69).

[Wan+19] J. Wang, S. Hazarika, C. Li, and H.-W. Shen. "Visualization and Visual Analysis of Ensemble Data: A Survey". In: *IEEE Transactions on Visualization and Computer Graphics* 25.9 (2019), pp. 2853–2872. DOI: 10.1109/tvcg.2018.2853721 (cit. on pp. 52, 70, 71).

[Wan+17] J. Wang, X. Liu, H.-W. Shen, and G. Lin. "Multi-Resolution Climate Ensemble Parameter Analysis with Nested Parallel Coordinates Plots". In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 81–90. DOI: 10.1109/tvcg.2016.2598830 (cit. on pp. 40, 72).

[War02] M. O. Ward. "A Taxonomy of Glyph Placement Strategies for Multidimensional Data Visualization". In: *Information Visualization* 1.3-4 (2002), pp. 194–210. DOI: 10.1057/palgrave.ivs.9500025 (cit. on p. 65).

[Was+14] J. Waser, A. Konev, B. Sadransky, Z. Horváth, H. Ribičić, R. Carnecky, P. Kluding, and B. Schindler. "Many Plans: Multidimensional Ensembles for Visual Decision Support in Flood Management". In: *Computer Graphics Forum* 33.3 (2014), pp. 281–290. DOI: 10.1111/cgf.12384 (cit. on pp. 50, 54, 56).

[Was+10] J. Waser, R. Fuchs, H. Ribičič, B. Schindler, G. Blöschl, and E. Gröller. "World Lines". In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 1458–1467. DOI: 10.1109/tvcg.2010.223 (cit. on pp. 41, 50, 53).

[WRK21] R. Wei, S. Rey, and E. Knaap. "Efficient Regionalization for Spatially Explicit Neighborhood Delineation". In: *International Journal of Geographical Information Science* 35.1 (2021), pp. 135–151. DOI: 10.1080/13658816.2020.1759806 (cit. on p. 67).

[Wei+16] J. Weissenböck, A. Amirkhanov, E. Gröller, J. Kastner, and C. Heinzl. "PorosityAnalyzer: Visual Analysis and Evaluation of Segmentation Pipelines to Determine the Porosity in Fiber-Reinforced Polymers". In: *IEEE Conference on Visual Analytics Science and Technology.* New York: IEEE, 2016, pp. 101–110. DOI: 10.1109/vast.2016.7883516 (cit. on pp. 48, 51, 55, 59).

[WN20] J. Wenskovitch and C. North. "An Examination of Grouping and Spatial Organization Tasks for High-Dimensional Data Exploration". In: *IEEE Transactions on Visualization and Computer Graphics* (2020), pp. 1–1. DOI: 10.1109/tvcg.2020.3028890 (cit. on p. 75).

[WN17] J. Wenskovitch and C. North. "Observation-Level Interaction with Clustering and Dimension Reduction Algorithms". In: *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*. Chicago, IL, USA: ACM Press, 2017, pp. 1–6. ISBN: 978-1-4503-5029-7. DOI: 10.1145/3077257.3077259 (cit. on p. 75).

[WAG06] L. Wilkinson, A. Anand, and R. 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. 1363–1372. DOI: 10.1109/tvcg.2006.94 (cit. on p. 75).

[WD08] J. Wood and J. Dykes. "Spatially Ordered Treemaps". In: *IEEE Transactions on Visualization and Computer Graphics* 14.6 (2008), pp. 1348–1355. DOI: 10.1109/tvcg.2008.165 (cit. on p. 65).

[Woo+17] R. Woodbury, A. Mohiuddin, M. Cichy, and V. Mueller. "Interactive Design Galleries: A General Approach to Interacting with Design Alternatives". In: *Design Studies* 52 (2017), pp. 40–72. DOI: 10.1016/j.destud.2017.05.001 (cit. on pp. 54, 59).

[WDR11] H. Wu, J. Dorsey, and H. Rushmeier. "Physically-Based Interactive Bi-Scale Material Design". In: *Proceedings of the 2011 SIGGRAPH Asia Conference*. Hong Kong, China: ACM Press, 2011, p. 1. ISBN: 978-1-4503-0807-6. DOI: 10.1145/2024156.2024179 (cit. on p. 56).

[Xia+22] J. Xia, Y. Zhang, J. Song, Y. Chen, Y. Wang, and S. Liu. "Revisiting Dimensionality Reduction Techniques for Visual Cluster Analysis: An Empirical Study". In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2022), pp. 529–539. DOI: 10.1109/tvcg.2021.3114694 (cit. on p. 75).

[XT15] D. Xu and Y. Tian. "A Comprehensive Survey of Clustering Algorithms". In: *Annals of Data Science* 2.2 (2015), pp. 165–193. DOI: 10.1007/s40745-015-0040-1 (cit. on p. 71).

[Xu+20] K. Xu, A. Ottley, C. Walchshofer, M. Streit, R. Chang, and J. Wenskovitch. "Survey on the Analysis of User Interactions and Visualization Provenance". In: *Computer Graphics Forum* 39.3 (2020), pp. 757–783. DOI: 10.1111/cgf.14035 (cit. on pp. 55, 58).

[Xu+19] K. Xu, M. Xia, X. Mu, Y. Wang, and N. Cao. “EnsembleLens: Ensemble-based Visual Exploration of Anomaly Detection Algorithms with Multidimensional Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 109–119. DOI: 10.1109/tvcg.2018.2864825 (cit. on p. 70).

[XW05] R. Xu and D. WunschII. “Survey of Clustering Algorithms”. In: *IEEE Transactions on Neural Networks* 16.3 (2005), pp. 645–678. DOI: 10.1109/tnn.2005.845141 (cit. on p. 71).

[Yañ+17] A. Yañez, J. Duggan, C. Hayes, M. Jilani, and M. Connolly. “PandemCap: Decision Support Tool for Epidemic Management”. In: *2017 IEEE Workshop on Visual Analytics in Healthcare (VAHC)*. 2017, pp. 24–30. DOI: 10.1109/vahc.2017.8387497 (cit. on pp. 50, 56).

[Yum+15] M. E. Yumer, S. Chaudhuri, J. K. Hodgins, and L. B. Kara. “Semantic Shape Editing Using Deformation Handles”. In: *ACM Transactions on Graphics* 34.4 (2015), 86:1–86:12. DOI: 10.1145/2766908 (cit. on pp. 46, 48).

[Zam+15] L. Zaman, W. Stuerzlinger, C. Neugebauer, R. Woodbury, M. Elkhaldi, N. Shireen, and M. Terry. “GEM-NI: A System for Creating and Managing Alternatives In Generative Design”. In: *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems*. Seoul Republic of Korea: ACM, 2015, pp. 1201–1210. ISBN: 978-1-4503-3145-6. DOI: 10.1145/2702123.2702398 (cit. on pp. 46, 56, 59).

[ZP04] X. Zhang and M. Pazner. “The Icon Imagemap Technique for Multivariate Geospatial Data Visualization: Approach and Software System”. In: *Cartography and Geographic Information Science* 31.1 (2004), pp. 29–41. DOI: 10.1559/152304004773112758 (cit. on p. 67).

[Zha+20] A. Zhao, J. Xu, M. Konaković-Luković, J. Hughes, A. Spielberg, D. Rus, and W. Matusik. “RoboGrammar: Graph Grammar for Terrain-Optimized Robot Design”. In: *ACM Transactions on Graphics* 39.6 (2020), 188:1–188:16. DOI: 10.1145/3414685.3417831 (cit. on p. 59).

[Zha+21] J. Zhao, X. Liu, C. Guo, C. Qian, and Y. V. Chen. “Phoenixmap: An Abstract Approach to Visualize 2D Spatial Distributions”. In: *IEEE Transactions on Visualization and Computer Graphics* 27.3 (2021), pp. 2000–2014. DOI: 10.1109/tvcg.2019.2945960 (cit. on pp. 66, 68, 69).

[Zho+17] M. Zhou, J. Tian, F. Xiong, and R. Wang. “Point Grid Map: A New Type of Thematic Map for Statistical Data Associated with Geographic Points”. In: *Cartography and Geographic Information Science* 44.5 (2017), pp. 374–389. DOI: 10.1080/15230406.2016.1160797 (cit. on p. 67).

[Zhu+18] L. Zhu, W. Xia, J. Liu, and A. Song. “Visualizing Fuzzy Sets Using Opacity-Varying Freeform Diagrams”. In: *Information Visualization* 17.2 (2018), pp. 146–160. DOI: 10.1177/1473871617698517 (cit. on p. 70).

CHAPTER 3

Visual Parameter Analysis for Temporal Blind Source Separation

The content of this chapter is published in:

Nikolaus Piccolotto, Markus Bögl, Theresia Gschwandtner, Christoph Muehlmann, Klaus Nordhausen, Peter Filzmoser, Silvia Miksch. *TBSSvis: Visual Analytics for Temporal Blind Source Separation*. Visual Informatics, vol 6, no. 4, 2022. DOI: 10.1016/j.visinf.2022.10.002.

Context. The following publication summarizes our first experiences with BSS and presents the initial results of our collaboration with statistics/BSS experts (co-authors Christoph Muehlmann, Klaus Nordhausen, and Peter Filzmoser). The particular BSS method under investigation is gSOBI (Section 1.1.3). We propose a task abstraction following the multi-level typology by Brehmer and Munzner [BM13] that lists both the data interesting to analysts and the necessary tasks (e.g., *compare parameter settings*). The VA prototype guides analysts in selecting the lag set parameter. It supports the PSA tasks *optimization* and *sensitivity*. As part of this design study, we suggest novel data mining and visualization techniques that support the particular data structures found in TBSS, e.g., a set-aware k-medoids clustering and interweaved histograms. We evaluated the VA prototype in expert interviews with five BSS experts.

RQ's Concerned: RQ1, RQ2, RQ3, RQ4.

3.1 Abstract

Temporal Blind Source Separation (TBSS) is used to obtain the true underlying processes from noisy temporal multivariate data, such as electrocardiograms. TBSS has similarities to Principal Component Analysis (PCA) as it separates the input data into univariate components and is applicable to suitable datasets from various domains, such as medicine, finance, or civil engineering. Despite TBSS's broad applicability, the involved tasks are not well supported in current tools, which offer only text-based interactions and single static images. Analysts are limited in analyzing and comparing obtained results, which consist of diverse data such as matrices and sets of time series. Additionally, parameter settings have a big impact on separation performance, but as a consequence of improper tooling, analysts currently do not consider the whole parameter space. We propose to solve these problems by applying visual analytics (VA) principles. Our primary contribution is a design study for TBSS, which so far has not been explored by the visualization community. We developed a task abstraction and visualization design in a user-centered design process. Task-specific assembling of well-established visualization techniques and algorithms to gain insights in the TBSS processes is our secondary contribution. We present TBSSvis, an interactive web-based VA prototype, which we evaluated extensively in two interviews with five TBSS experts. Feedback and observations from these interviews show that TBSSvis supports the actual workflow and combination of interactive visualizations that facilitate the tasks involved in analyzing TBSS results.

3.2 Introduction

Multivariate measurements of a phenomenon are common in many domains. Medical doctors place electrodes on a patient's body to analyze processes such as brain activity, eye movements, or heart rhythm. Civil engineers measure vibrations on different parts of a structure, such as a bridge, to detect possible faults. Financial managers invest money in stocks, which are in a way sensors of economic processes, to gain wealth. Common to all these examples is the time-oriented data and the assumption that data from different sensors are in some way correlated and/or influenced by noise. However, analysts are usually only interested in the "true" underlying processes.

To obtain these processes, analysts turn to Blind Source Separation (BSS). BSS comprises established methods for signal separation that were applied, among others, in the mentioned domains of medicine [CJ10; dLdMV00; Van+17], civil engineering [AA16] and finance [OKM00]. Temporal Blind Source Separation (TBSS) refers to a subset of BSS methods that specifically account for temporal correlation. TBSS is similar to Principal Component Analysis (PCA) in the sense that i) TBSS methods work on any multivariate dataset with quantitative variables, ii) they work on measured data only (hence "blind") and iii) separate it into a linear combination of uncorrelated components, like PCA. Unlike PCA, TBSS accounts for temporal correlation and often requires complex tuning parameters. As both TBSS and PCA can be considered forms of dimensionality reduction, analysts use TBSS and PCA for similar reasons, like data analysis or modeling/prediction.

During these activities, it is at some point necessary to inspect components visually. Like with PCA, components are hidden until the separation algorithm is executed, but TBSS's complex parameter space severely complicates the issue: It is known that parameter settings greatly influence the result, but not in which way a change in parameters translates to change in components. Experts regard automated analysis by extensive sampling [Sed+14] not a feasible option and there is little guidance from the literature, which parameters to pick. Because a ground truth is rarely available, TBSS analysis is inherently open-ended and exploratory as there are no known insights to confirm. The workflow of TBSS analysts can broadly be described as i) pick a parameter setting, ii) see if obtained components are useful or interesting and if not, go to i).

Some challenges make TBSS difficult to use in practice. Despite the important role of visualization in their workflow, the current tool used by the analysts does not support them well in this regard. Analysts need to manually program static visualizations, which requires time they could otherwise spend on data analysis. Another challenge is the amount of components. Each parametrization on a p-variate dataset yields a set of p components that need inspection and comparison to previous sets. Analysts are, for example, interested in commonly found components, but very quickly confronted with hundreds of components to consider. This is a common task in ensemble visualization [Wan+19], but made more difficult by components appearing in sets instead of one by one. Also, when comparing multiple results, analysts will eventually find competing options for their final choice. As there is usually no ground truth available to compare the result to, analysts need detailed ways to compare individual results to make an informed decision.

Visual analytics (VA [TC05]) as defined by Keim et al. [Kei+08] "combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data sets." Considering the strong focus of BSS analysis on visual inspection on multiple levels of detail, in combination with mentioned challenges, we propose applying VA principles to overcome these. We designed TBSSvis according to Munzner's Nested Model [Mun09] for the TBSS method "generalized Second Order Blind Identification" (gSOBI) [Mie+20]. We chose gSOBI because it is recent and well suited to real-world datasets due to its flexibility (see Section 3.4). The source code of TBSSvis is available at `https://github.com/npiccolotto/tbss-vis`.

Our primary research contribution is a design study [Sed16] for TBSS, which improves the visualization community's knowledge about an area that it did not explore so far. Specifically we provide:

- A task abstraction for TBSS which we obtained through a user-centered visualization design process with TBSS experts (Section 3.6).

- A VA design for gSOBI, a TBSS method, that supports the abstracted tasks by combining visualizations, interactions, and guidance methods (Section 3.7).
- Confirmation of the effectiveness of our design in two interviews with five TBSS experts (Section 3.9).

As part of this design study we put well-established visualization techniques together to support the identified tasks. They include a multivariate autocorrelation function plot and the application of a slope graph to sets of time series. These, together with a set-aware clustering scheme (Section 3.7.3), are our secondary contribution.

3.3 Related Work

In the following, we elaborate on different approaches to visualize and compare time series, ensembles, and models.

3.3.1 Time Series Visualization

Temporal data is ubiquitous in many domains such as finance, health, or biology, and has been visualized for centuries since the first line graph was introduced by Playfair [Tuf01]. Various other visual encodings have been proposed afterwards, such as tile maps, sparklines, or horizon graphs [Aig+11]. They use different visual variables [Mac86] such as position, color, or slope, and therefore exhibit different perceptual properties, which makes them suitable for different analysis tasks. E.g., Gogolou et al. [Gog+19] investigated the relation between different time series visualization idioms and perceived similarity. They recommend to use horizon graphs when local variations in temporal position or speed is important, while others (line graph, color band) are better suited for notions of similarity where amplitude is less important. As this is the case with TBSS, where analysts look for patterns independent of amplitude, we show time series as the familiar line graph.

When multiple time series are at hand, their respective visualizations need to be composed. Two popular approaches to do so are superposition and juxtaposition. Superimposed encodings trade decreased usage of display space for legibility, as they do not scale well after a couple of variables due to occlusion. An example besides the well known superimposed line graph is the braided graph [JME10], which superimposes multiple area-based marks. Because of the varying data dimensionality in TBSS, superimposition is generally not a promising strategy. Multiple time series can also be composed with juxtaposition, as is the case in LiveRAC [McL+08]. Various system measures (columns) are displayed per machine (rows) in a space-filling table design, using semantic zooming to change the level of detail between color bars, sparklines, and labeled line graphs. When not using all available space, one could use small multiples [Tuf01] in different arrangements. For instance, Stitz et al. [Sti+16] arrange small multiples of stocks by price and price change in a user-selected time frame. Liu et al. [Liu+18], on the other hand, lay them out with a modified Multidimensional Scaling (MDS) algorithm such that similar items are near each other. These approaches proved to be very useful for individual time series, but cannot be applied as such to TBSS, where sets of time series are involved. In the experience of our collaborators only some time series in TBSS will carry a signal and be interesting for closer inspection. Our approach employs various strategies to account for both facts, e.g., grouping time series by similarity and sorting representatives by a user-selected degree-of-interest function, or juxtaposing time series sets in a table-like design.

To keep features of long time series visible, designers often turn to focus-and-context techniques, such as lenses [Tom+17]. In the simplest case, a lens mainly enlarges an area of interest, such as in SignalLens [Kin10]. But more complex interactions are possible, such as in ChronoLenses [Zha+11], where users can combine and stack multiple lenses. As we designed TBSSvis for analysts who are accustomed to text-based interfaces, we took care to avoid complex interactions. Time series may be enlarged up to a certain level of detail in discrete steps and filtered to a contiguous subset of the currently visible time interval with simple direct manipulation interactions. We describe them in Section 3.7.1. We did not employ data reduction methods, neither in a data-driven [Shu+18] nor visualization-driven way, e.g., by line simplification [Ros+20], as one risks that important features are removed.

3.3.2 Ensemble Visualization

The goal in ensemble visualization is to make sense of a set of similar complex data items, such as trajectories, often produced by a simulation with perturbed parameter settings. Component sets obtained from different TBSS parametrizations constitute such an ensemble, where each ensemble member is a set of time series. Ensemble visualization has its origin in meteorology [Pot+09], but since expanded to more domains [Wan+19]. Analytic tasks for ensemble data [Wan+19] indicate popular strategies, such as comparing members or grouping them by similarity, to support the stated goal. Existing works [HHB16; Fer+17] often use popular clustering techniques (with domain-specific distance functions) to support the latter task. This is not straightforward in TBSS as one has to take care to not mix members of different sets into the same cluster. We discuss our approach, a custom clustering algorithm that respects this constraint, in Section 3.7.3.

Time is a common part of ensemble data, but not a requirement [Mat+09; Pir+12; MGH18; Xu+19]. One possible case is when ensemble members are univariate time series, such as for Köthur et al. [Köt+15], who encoded the correlation between members in a heatmap to support comparison of two ensembles. More commonly, other data types have an associated time dimension such as multivariate data [OBJ16], particle data [HHB16], network security data [HHH15], or spatial data [Buc+19]. However, ensembles of sets of time series, as in our case, are not thoroughly explored so far and our paper presents a first step in that direction.

3.3.3 VA for Model Construction

VA supported the construction and validation of various kinds of models, such as linear regression [MP13; Zha+14], logistic regression [Din+19], dimensionality reduction [AWD12], classification [Cho+10], or artificial neural networks [Zha+19; Wex+20]. Most works in the literature focus on non-temporal data, Bögl et al. [Bög+13] (univariate time series modeling) and Sun et al. [Sun+20] (univariate time series forecasting) provided two exceptions. TBSSvis, supporting construction and comparison of TBSS model alternatives, extends the state of the art as TBSS works on multivariate time series. During the construction step, questions of analysts pertain to which variables should be included, how many parameters should the model have, and which subgroups should be modeled. The latter question is closely related to model validation, where analysts, e.g., verify that a model works for diverse data cases, or how multiple models agree/differ on outputs, such as predicted class labels. Established diagnostic plots or data exist for several of these procedures, e.g., residual plots in time series modeling [Bög+13], or confusion matrices in classification [Wex+20]. In contrast, the quality of a TBSS model is solely defined by the presence of domain-specific interesting features in the output, thus TBSSvis focuses on comparing multiple alternatives in terms of similarity of their output and parameter settings. A complicating factor that we tackle is that TBSS outputs are *sets* of time series.

3.4 Temporal Blind Source Separation

The statistical analysis of multiple measurements taken at different times is a challenging task. Often, such multivariate time series are analyzed by transforming the data in certain simple ways to uncover latent processes which generated the data. Probably the most used method for such a task is the classical PCA, which uses linear transformations of the data that result in components which have highest variance and are uncorrelated. Uncorrelated components imply that the covariance between the found linear combinations is zero. The linear combinations are given by diagonalizing the covariance matrix. Furthermore, as the nature of the transformation is linear, interpretations of the results can be carried out by the simple and well studied loading-scores scheme. However, PCA might not be the best choice when the data at hand shows dependencies in time, as the main source of information is in that case not covariance, but rather serial dependence. Serial dependence is characterized by autocovariance, i.e., covariance between measurements separated in time by a given lag. In analogy to PCA it would be desirable to find linear combinations of the multivariate time series data which are not only uncorrelated marginally (zero covariance between variables at each time step), but also uncorrelated in time (zero autocovariance between variables for any lag). TBSS is a field of multivariate statistics that studies methods delivering the former desired properties. Generally, BSS is a well established model-based framework. It assumes that the observed data are a linear mixture of latent components, which are considered usually easier to model and/or more meaningful for interpretation than multivariate models. The goal of BSS is to recover these components based on the observed data alone. BSS is formulated and used for many types of data, as outlined in recent reviews [CJ10; NO18; Pan+; NR22]. In the following, we outline the concept of TBSS.

The model of TBSS considered here is $\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{c}_t$, where $\boldsymbol{x}_t$ denotes the observed p-variate time series, $\boldsymbol{A}$ is the full-rank $p \times p$ mixing matrix and $\boldsymbol{c}_t = (c_{1,t}, \dots, c_{p,t})^\top$ is the set of p latent components, which should be estimated. Thus the goal is to find a $p \times p$ unmixing matrix $\boldsymbol{W} = (\boldsymbol{w}_1, \dots, \boldsymbol{w}_p)^\top$, such that $\boldsymbol{c}_t = \boldsymbol{W}\boldsymbol{x}_t$ up to sign and order of the components in $\boldsymbol{c}_t$. To facilitate the recovery, the assumption is made that the components in $\boldsymbol{c}_t$ have $\mathrm{Cov}(\boldsymbol{c}_t) = \boldsymbol{I}_p$ and are uncorrelated (or independent) with mutually distinct serial dependence. This means, for example, that all cross-moment matrices, such as autocovariance matrices, of $\boldsymbol{c}_t$ are diagonal matrices.

A very first approach for TBSS is the Second-Order Blind Identification (SOBI) algorithm [Bel+97; Mie+14; Mie+16]. It finds the linear combinations of the data which make autocovariance matrices for several lags as diagonal as possible. Hence, found components are uncorrelated marginally and uncorrelated in time. It is well known in the statistical analysis of time series data, that time series emerging from different scientific fields have different key characteristics. For example, financial time series are not well characterized by autocovariance matrices, but instead higher-order moments carry the most information. This is denoted as stochastic volatility and in the TBSS literature it is shown that SOBI fails for such time series [Mat+17]. Higher-order moments relate often to skewness and kurtosis and, for example in our context, to the covariance of the squared data and are meant to detect more unusual observations (heavy tails). An intuitive notion of higher-order moments is that they translate to quickly changing effects, such as stock prices that increase/decrease by large margins within short time frames (high volatility). In such cases, higher-order moments describe volatility better than second-order moments. To overcome this issue, a new TBSS method, denoted as a variant of SOBI (vSOBI) [Mat+17], was introduced. Similar to SOBI, vSOBI finds the latent time series by diagonalizing matrices of lagged fourth moments. Uncovered latent components are uncorrelated marginally and additionally have zero fourth-order dependence.

Generally, time series might carry information both in the autocovariance and in the higher-order time dependence, thus a combination of SOBI and vSOBI might deliver the best results. Indeed, Miettinen et al. [Mie+20] proposed such a method, referred to as generalized SOBI (gSOBI), which we focus on in this manuscript. It diagonalizes several autocovariance matrices (SOBI part) and several matrices of lagged fourth moments (vSOBI part). This method has three rather involved tuning parameters. The first one $b \in [0, 1]$ weighs SOBI versus vSOBI, where SOBI ($b = 1$) and vSOBI ($b = 0$) are the extreme cases. The second ($\boldsymbol{k}_1$) and third ($\boldsymbol{k}_2$) tuning parameters provide the sets of lags used for the SOBI and vSOBI part, respectively. A lag is a time interval given by a number of time steps, the size of which is determined by the resolution of the underlying time series. For instance, a lag of 6 in an hourly observed thermometer refers to an interval of 6 hours. Common default values for gSOBI are $b = 0.9$, $\boldsymbol{k}_1 = \{1, \dots, 12\}$ and $\boldsymbol{k}_2 = \{1, 2, 3\}$ [Mie+20], but Miettinen et al. also show that the selection of lag sets and weight has a huge impact on the performance [Mie+20]. Vague guidelines for these tuning parameters exist in the community, such as lag sets should not be too small or too large, and the lags should be chosen so that the corresponding (cross-)moment matrices for the *latent* components have diagonal values far apart. Thus, parameter selection in the context of SOBI is a highly complex problem with no practical solution yet [TLS05; TMN16]. First steps for an informed trial & error routine can be determined from those guidelines and by looking at the data. As an example, if the time series at hand show substantial volatility, then the b parameter (weight of second- vs. fourth-order moments) would initially be chosen closer to 0. Otherwise, volatility observed in the dataset might not be visible in latent components. Lag sets generally can be chosen by observing how long any visible patterns, like volatility, last. If they are rather short, then short lags are more suitable than longer lags, and vice versa. The interdependence between parameter choice and the variation in the output depends a lot on the dataset at hand, so much so that general statements about it would be misleading. Instead we propose an advanced VA approach, which allows defining alternative parameter choices and comparing the respective outputs effectively, to discover such relations.

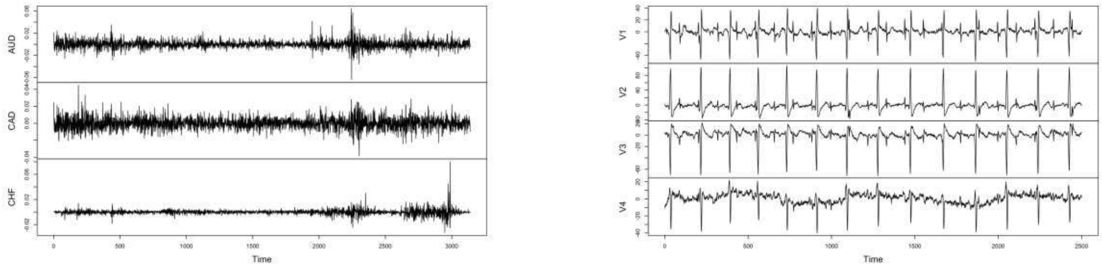
The R implementation of gSOBI used in the following is available in the package tsBSS [Nor+21]. We call one execution of gSOBI a *run*. As outlined before it yields a set of p univariate time series, which we call *components*. The outcomes of multiple runs with varying parameter settings form an ensemble, where each member corresponds to a single run. A member has the used parameters $\boldsymbol{k}_1$, $\boldsymbol{k}_2$ and b associated, as well as the output of gSOBI. The latter is either the component set $\boldsymbol{c}_t$ and the estimated unmixing matrix $\hat{\boldsymbol{W}}$, or nothing, in case the (cross-)moment matrices could not be diagonalized in a predefined number of iterations. We call a run *succeeding* or *failing*, depending on the outcome.

3.5 Datasets

In this section we introduce two datasets, one from the financial domain (Figure 3.1a) and one from the medical domain (Figure 3.1b), along with reasons why TBSS analysis of them can be desired. Analysis of both datasets shares similar tasks. For instance, analysts are interested in relevant parameter subspaces, common components and alternatives to them, as well as the stability of obtained results. We formalize typical tasks and questions involved in TBSS analysis in Section 3.6.

3.5.1 Financial data

Goods, currencies, and company stocks are traded every day at high frequencies. In simple terms, investors make money by buying something at a price X and selling it later at a price Y larger than X. To maximise $Y - X$ in a short time frame the idea here is to find a volatile collection of currencies or stocks (a portfolio), i.e., one that is subject to sudden and extreme changes in value. To do so, we look at the daily exchange rate of 23 currencies to Euro between the years 2000–2012 (23 variables, 3 139 time steps). We preprocess the data to get logarithmic returns, a common measure in quantitative



(a) First three currencies of the *exrates* daily currency exchange rate dataset.

(b) First four variables in the ECG dataset.

Figure 3.1: Datasets in this paper.

finance when the temporal behavior of return is of interest. The first three variables are shown in Figure 3.1a.

3.5.2 Medical data

An electrocardiogram (ECG) is a recording of the heart's electrical activity. To obtain it, electrodes are placed on the patient's skin. These electrodes detect small electrical changes which occur due to muscle de- and repolarization. ECGs are important for medical analysis as many cardiac abnormalities show deviations to the normal ECG pattern. Analysis of fetal ECGs may detect problems during fetal development, such as fetal distress. While invasive methods exist to measure the fetal ECG directly, a non-invasive method is often preferred as it does not harm neither mother nor fetus. The fetal ECG is visible in the mother's ECG, but it is weak and mixed with, e.g., respiratory noise or frequency interference (compare first three rows in Figure 3.1b). Using TBSS on ten seconds of the ECG of a pregnant woman (8 dimensions, 2 500 time steps), we try to extract the fetal ECG following previous work [dLdMV00].

3.6 Task Abstraction

In this section we present a task abstraction for TBSS. We structure it according to the data-users-tasks triangle by Miksch and Aigner [MA14] and use the terminology by Brehmer and Munzner [BM13] for tasks. We developed the abstraction together with the visualizations in an iterative design process following Munzner's Nested Model [Mun09] with three collaborators, who are co-authors of this paper and experts in BSS. In this user-centered design process model, we first conducted unstructured interviews in order to understand their problems and made ourselves familiar with literature they provided. After that, we discussed our assumptions and ideas regularly with them over a course of nine months. We discussed iteratively developed prototypes ranging from hand-drawn sketches, to static digital images, to an interactive application which is described in Section 3.7. During these sessions, we also questioned our current understanding of their tasks either implicitly through visualization designs or explicitly through discussions. In the end, we interviewed five TBSS experts, who did not collaborate with us on the design, to further validate our abstracted tasks (Section 3.9). The presented task abstraction is a reflection on this process.

We touched upon the involved data with TBSS in Section 3.4 and Section 3.5 already. These are a multivariate time series (input data), one real and two sets of integers (TBSS parameters b, $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$) and a set of univariate time series (latent components). The temporal dimension is discrete and linear.

3.6.1 Users

Our users are data analysts or data scientists with formal education in statistics/math and basic knowledge of BSS. They may also be experts in a specific application domain, like medicine or finance. They work mostly with R [R C23], a language and environment for statistical computation in which most BSS researchers publish their implementations. The preferred work environment is RStudio, a popular text-based development environment for R. Currently, they use built-in plotting functionality, and sometimes they use, for example, ggplot2 to build customized visualizations. The output of either option is a static visualization, of which RStudio by default displays only one at a time. Because of this, our users are accustomed to well known static statistics visualizations such as histograms, line graphs, box plots, etc.

3.6.2 Tasks

During this user-centered design process we identified the following tasks, which we describe using the abstraction terminology by Brehmer and Munzner [BM13].

The high-level workflow can be separated into three phases, which are depicted in Figure 3.2: Analysts first inspect the raw input data, continue to find parameter settings, and then analyze obtained components. Given the exploratory nature of their analysis process, analysts switch between the latter two phases until they feel they exhausted the parameter space or obtained a useful result.

Generally, analysts want to *discover* observations or *derive* a modified dataset with reduced dimensionality. There are two main *targets* of analysis. Components, which are mostly analyzed as sets, are one target. Still, analysts want to *discover* and *explore* interesting components, whatever interesting means in the data domain. Parameters are the other analysis target. Analysts look for a “stable” result, i.e., one that can be obtained with rather diverse parameter settings. The assumption is that its components are then more likely to represent real processes. To this end, they need to *compare* components and parameters of different runs. As an additional obstacle, when lacking intuition and/or domain knowledge, analysts struggle to select (*lookup*, *locate*) parameters and need guidance support [Cen+17] so they can *browse* and *explore* parameter settings in an informed way. All these observations lead us to some *low-level query actions*:

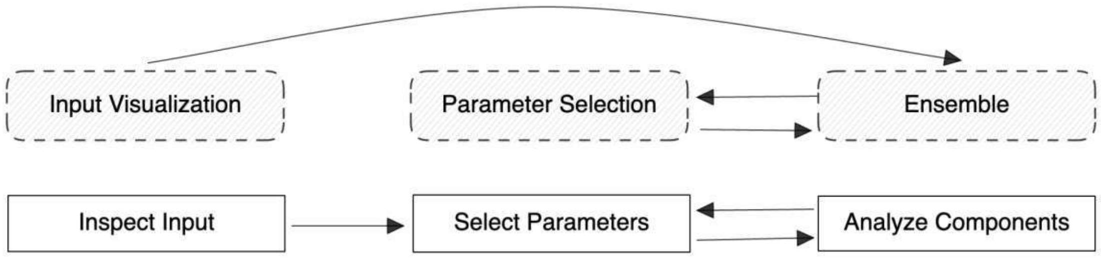


Figure 3.2: The existing analysis workflow (bottom) and the corresponding screens in TBSSvis (top). The new workflow automatically obtains initial results and analysts can start exploring immediately.

I1: Identify used parameters. Analysts want to see values of existing parametrizations. In case of lag sets they inspect the distribution of chosen lags and if one lag set contains more lags than the other.

I2: Identify unmixing matrix. Analysts turn to the unmixing matrix to interpret components and to understand how they were formed. They look for large absolute values per component.

I3: Identify cross-moment diagonality. Analysts want to inspect runs on a technical level that is currently inconvenient to obtain and difficult to quantify. If the TBSS model holds, then all cross-moment matrices are exactly diagonalized by the unmixing matrix estimate. For real data, this is, however, rarely the case and thus analysts are interested in the impact of the parameters on the diagonality of the different cross-moment matrices.

I4: Identify components. In a single component, analysts look for interesting features like outliers or uncommon changes in shape. They thereby also check the absence of features (noisiness). Analysts are further interested in the stability of a component, i.e., in how many ensemble members the component is present.

C1: Compare success. First of all, prior to any comparisons, analysts must know what can be compared. If a run did not succeed, only parameters could be compared as no components were found.

C2: Compare parameters. To carry out sensitivity analysis, analysts need to compare parameters between runs. They mainly look at differences in lag distribution and amount.

C3: Compare unmixing matrices. Before inspecting factors of individual components, analysts investigate the similarity of unmixing matrixes by means of a custom metric (MD-Index [Ilm+10]).

C4: Compare component sets. This task mostly relates to membership, which, however, is difficult to assert with complex objects, such as time series, where one usually speaks of similarity instead of equality. Analysts compare components only between sets and want to know which component exists in multiple sets, and if so at which ranks, and if not which is the most similar component, plus in which time frames components disagree.

C5: Compare possible parameters. When choosing a new parametrization, analysts need guidance through the parameter space and the ability to compare possible parameters in some meaningful way to find promising settings.

3.7 Visualization Design & Justification

In this section, we present the visualization design we obtained based on the task abstraction (Section 3.6) and implemented in a web-based prototype for gSOBI. A design goal was to make TBSSvis generic enough to allow its use in many application domains, because, like PCA, TBSS is a domain-independent method. After a cursory literature search of TBSS applications and considering the available amount of pixels on common screen sizes, we designed TBSSvis for inputs with the length of up to 5 000 time steps, and up to 50 dimensions in mind. These limits do not accommodate extreme cases we found, like EEG data (128 dimensions, 1.2 million time steps [TLS05]), but are suitable to financial (40 dimensions, 140 time steps [OKM00]) or civil engineering (3 dimensions, 9 000 time steps [Liu+19]) use cases we found. Our design guidelines denote soft limits of an interactive TBSS application: The quality of visualizations will gradually decline with data of higher complexity and the execution time of the used TBSS algorithm will increase.

While we implemented visualizations for all abstracted tasks, for brevity we will focus on an illustrative subset of those. Specifically, we will discuss visualizations for tasks that pertain to

- identifying and comparing components (or sets thereof),
- identifying and comparing used parameters, and
- comparing possible parameter settings.

TBSSvis consists of three screens, which are depicted with their connection to analysis phases in Figure 3.2. The *Input Visualization* screen shows the raw input data, a feature requested by our collaborators. The *Ensemble* screen allows exploration of parameter settings and components. Finally, the *Parameter Selection* screen is used to select new parameter settings. We will focus on the latter two. How presented visualizations work together is illustrated in the usage scenarios (Section 3.8).

3.7.1 Time Series Visualization and Interactions

Time series are plotted vertically aligned to facilitate comparison and ordered by variable name (for input variables) or by an interestingness function (for latent components, see below). The display of and interaction with all time series in TBSSvis is handled by the same logic as shown in Figure 3.3. Due to the length and amount of time series, we employ semantic zooming and at first save display space by drastically shrinking their Y axis and omitting any labels by default. This can be changed with interaction: On hover, we display axis labels for the hovered time series. The Y axis of an individual time series can be enlarged in discrete steps by another interaction (pressing hotkey on keyboard and clicking). If an analyst is interested in a contiguous subset of the time series, it is possible to zoom in with brushing, which will affect all time series in the application. Both the semantic and temporal zoom can be reset with interactions recommended by Schwab et al. [Sch+19].

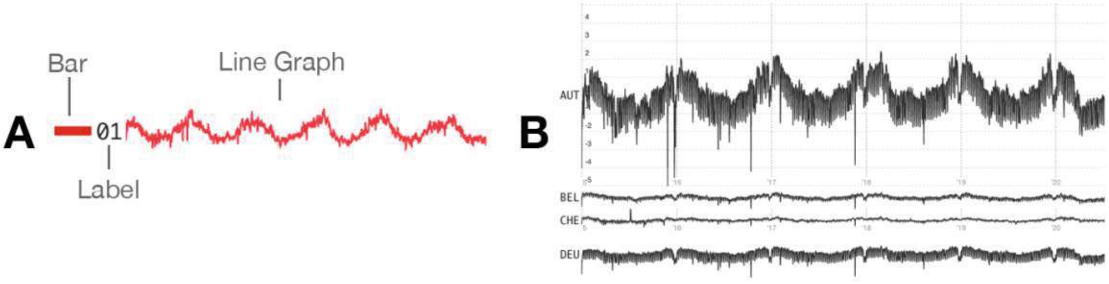


Figure 3.3: Display of and interaction with time series. (A) A time series is displayed with a line graph, an optional label, and an optional bar to its left. The bar encodes a DOI function of a time series. (B) Typical vertical arrangement of multiple time series in TBSSvis. The user enlarged the first and last time series to different sizes and hovered over the third, thus its X axis labels are shown.

As described in Section 3.4, the order of components is not defined. In practice, this means that analysts use measures which are sign-independent to compare components, such as absolute Pearson correlation, and impose an order by sorting components according to a function. We will call this a *degree-of-interestingness function* (DOI), and require it to be any function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ that maps a time series of length n to a scalar. Because TBSS is a domain-independent method, many DOI functions could be useful [Fu11] depending on what the domain's interesting features are. E.g., for detailed cardiac analysis, different widths and types of ECG wave patterns could be mined. Based on discussions with our collaborators we use the absolute third (skewness) and fourth moment (kurtosis) in TBSSvis. These are useful to find the most skewed components and those with the most outlying values, as the first two moments (mean and variance) of all components are identical. We added a measure for periodicity [VYC05] after our user studies. The DOI function can be changed in the toolbar, and all views that show component-related data will update as the set members are sorted in descending order based on the new DOI function values.

3.7.2 Color

According to Mackinlay [Mac86], color is the most effective visual variable for nominal data after position, and, therefore, often used to encode different data classes. In multiple views, the same classes should be encoded with the same palette [QH18]. Because humans can only reasonably distinguish a few different colors, we cannot statically assign colors to all ensemble members. We, therefore, use a user-controlled dynamic assignment of colors of a qualitative palette to encode data related to user-selected members. The available colors are displayed in the toolbar and can be reordered with drag & drop. When hovering over an unselected member, the next free color (left to right) is a) used to highlight its related data in all views and b) associated to the member when selected. As one color is always needed for highlighting, the last free color cannot be used for selection. The color order determines the plotting order in all comparison views.

3.7.3 Tasks I4/C4: Identify/Compare Components (Sets)

We precompute initial parameter settings automatically, to allow immediate exploration of the output space: Variations of gSOBI's R package's defaults (3 settings), a recommendation from the literature [TLS05] (1 setting), and an additional user-defined number of random settings. This overcomes initial hesitation towards parameter setting choice and may give an estimate of what the relevant parameter subspace is.

Each successful run (Section 3.4) produces a set of time series. Already at the start of the analysis after precomputation, the amount of components to consider might be in the hundreds. Clustering is an established approach to counteract this, where data cases are grouped by similarity. This allows an analyst to focus on representative elements of the clusters. Many clustering techniques exist [XW05; XT15], but using them with all components from all sets has a major drawback: The clustering scheme will put components from the same set into the same group, which our collaborators found undesirable. The grouping should respect the set structure in the data and group components only between sets, not within them. Additional requirements we gathered for the clustering scheme are that it should not depend on a distance metric (unlike, e.g., k-means) and produce an existing data case as cluster representative (again unlike, e.g., k-means). The former is related to the similarity measure for components suggested by our collaborators, the difference in absolute Pearson correlation $dist_{cor} = 1 - |cor(c_i, c_j)|$. Since we do not know if it supports the triangle inequality, we should not rely on it. The latter requirement stems from the design principle to show actual data over visual abstractions.

Clustering Algorithm

We developed a custom clustering scheme to achieve our requirements. Starting from the realization that we basically want k-medoids, as it does not need a distance and produces existing representatives (medoids), we looked for a way to constrain the clustering process to obey the set structure. Constrained versions exist for k-means [Wag+01], but we did not find one for k-medoids. However, it was possible to adapt it using a k-means-like formulation of k-medoids [PJ09]. Constraints in our case are of the type *cannot-link*, i.e., they express which data cases must not be grouped into the same cluster. We add one cannot-link constraint per pair of elements that belong to the same set. For m sets, each containing p data cases, this amounts to $mp(p-1)/2$ constraints in total.

Algorithm 3.1 shows pseudocode of our custom clustering scheme. The initial medoids are obtained by an unconstrained k-medoids algorithm [SR19] on line 2. Following [PJ09] we use the sum of distances from all data cases to their medoids as cost function (line 3) and compute it for the initial clustering. Then, while constraints are violated, i.e., there is a cannot-link constraint for any two data cases assigned to the same medoid, we update the clustering (line 5). If there are no violated constraints, there is nothing to do as the initial clustering is a valid solution. Otherwise, we first select the most central of data cases assigned to same medoid (i.e., with smallest sum of distance to other data cases in that cluster) as a new medoid (line 6). Data cases are then reassigned to the nearest medoid that does not already contain another data case for which there exists a cannot-link constraint (line 7). These are steps 2 and 3 in the k-means-like formulation for k-medoids [PJ09]. We update the cost for the current clustering (lines 8–9) and repeat this loop (lines 5–12) until no constraints are violated anymore. Small necessary checks, e.g., whether or not there are still violations after the loop, were left out for brevity.

Algorithm 3.1: Pseudocode of constrained k-medoids, with which we obtain a clustering on sets of time series (Section 3.7.3).

```
   Data: Dissimilarity Matrix $D$, constraints $C$, medoids $M$, assignments of data
         cases to medoids $A$, number of partitions $k$
 1 Function $constrainedPAM(D, C, k)$ is
 2     $M, A \leftarrow$ FastPAM($D$, $k$)
 3     $cost \leftarrow$ getCost($A$, $D$)
 4     $cost' \leftarrow cost$
 5     while violatesConstraint($A$, $C$) $or\ cost - cost' > \epsilon$ do
 6         $M' \leftarrow$ findNewMedoids($A$, $D$, $k$)
 7         $A' \leftarrow$ assignToNearestPossibleMedoid($M$’, $C$, $D$)
 8         $cost \leftarrow cost'$
 9         $cost' \leftarrow$ getCost($A$’, $D$)
10         $A \leftarrow A'$
11         $M \leftarrow M'$
12     end
13     return $(M, A)$
14 end
```

Clustering Quality and Number of Partitions

The constrained k-medoids clustering takes one user-provided parameter, which is the desired number of clusters. We use a scented widget [WHA07] to allow setting this parameter in an informed way (Figure 3.4-A). The bar chart in the widget shows the average cluster separation as a clustering quality measure for a given number of clusters. Therefore, values with high bars suggest the number of meaningfully different components in all currently available sets.

Component Overview

The cluster medoids are shown underneath the Clustering Quality visualization, vertically aligned in a list, sorted by the DOI rank of the medoid (Figure 3.4-B). To further support Task C4, we show a histogram to the left of the medoid. The histogram shows the rank distribution of the contained components in their respective sets. Additionally, we encode $dist_{cor}$ to the cluster medoid with opacity. This way, stable (stacked bars with high opacity) and unstable (scattered bars with low opacity) components have distinct histogram shapes.

Analysts can inspect components in a cluster by clicking the “eye” icon, after which the list item expands and lists contained components in the same fashion as cluster medoids. Clicking a bar in the histogram or a time series label selects the associated ensemble member.

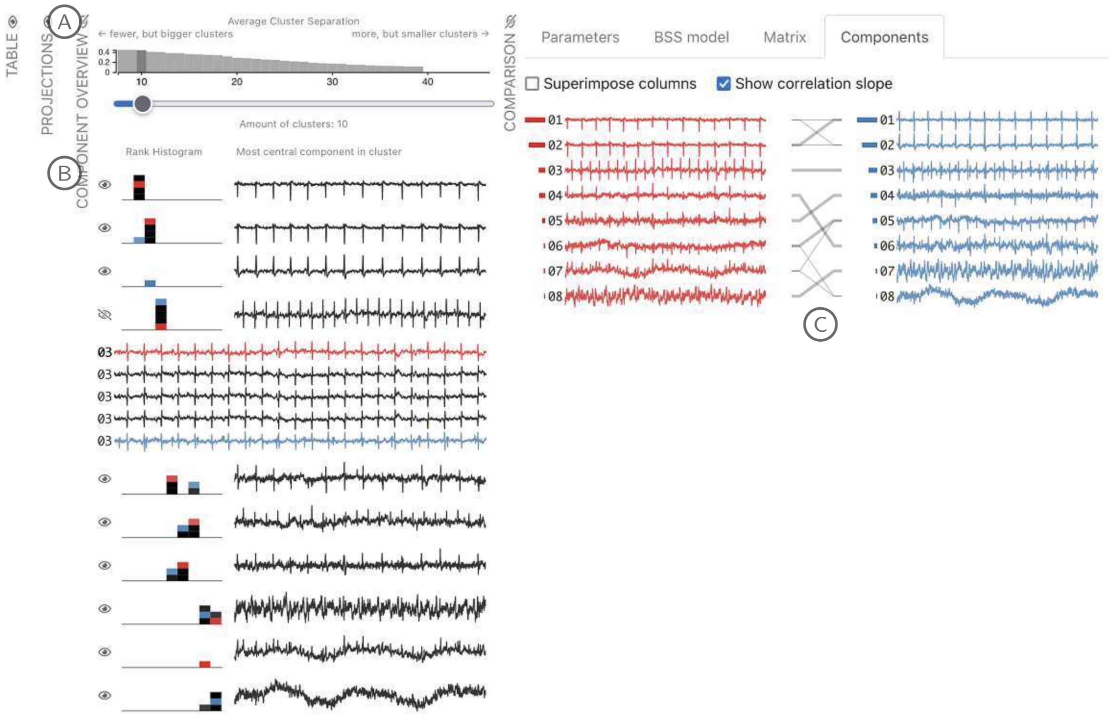


Figure 3.4: *Ensemble* screen of TBSSvis (medical data) configured to facilitate comparison and inspection of components and sets thereof. Left part shows the Clustering Quality (A), which suggests an optimal clustering with 8–10 partitions. Medoids of the 10 clusters are listed underneath (B). The fourth list item readily shows the fetal heart signal and was expanded to show cluster members. Two component sets (red and blue) were selected for detailed comparison (right half). The Slope Graph (C) highlights similar components, their rank changes and set similarity overall.

Slope Graph

Components of selected sets are visible in a separate view, again vertically aligned and sorted by DOI (Figure 3.4-C). Each selected set has a unique assigned color and all associated data is shown in this color. Multiple selections are juxtaposed horizontally in columns, which can be rearranged by the analyst. Analysts can inspect components visually as they are, or they can also display a slope graph between columns. Lines of the slope graph connect similar components, and thickness encodes similarity from high correlation (thick) to low. This way, it is easy to see stable (thick, single, mostly straight lines) and unstable components (no or thin, multiple, tilted lines), their rank changes and set similarity at a glance.

3.7.4 Tasks I1/C2: Identify/Compare Used Parameters

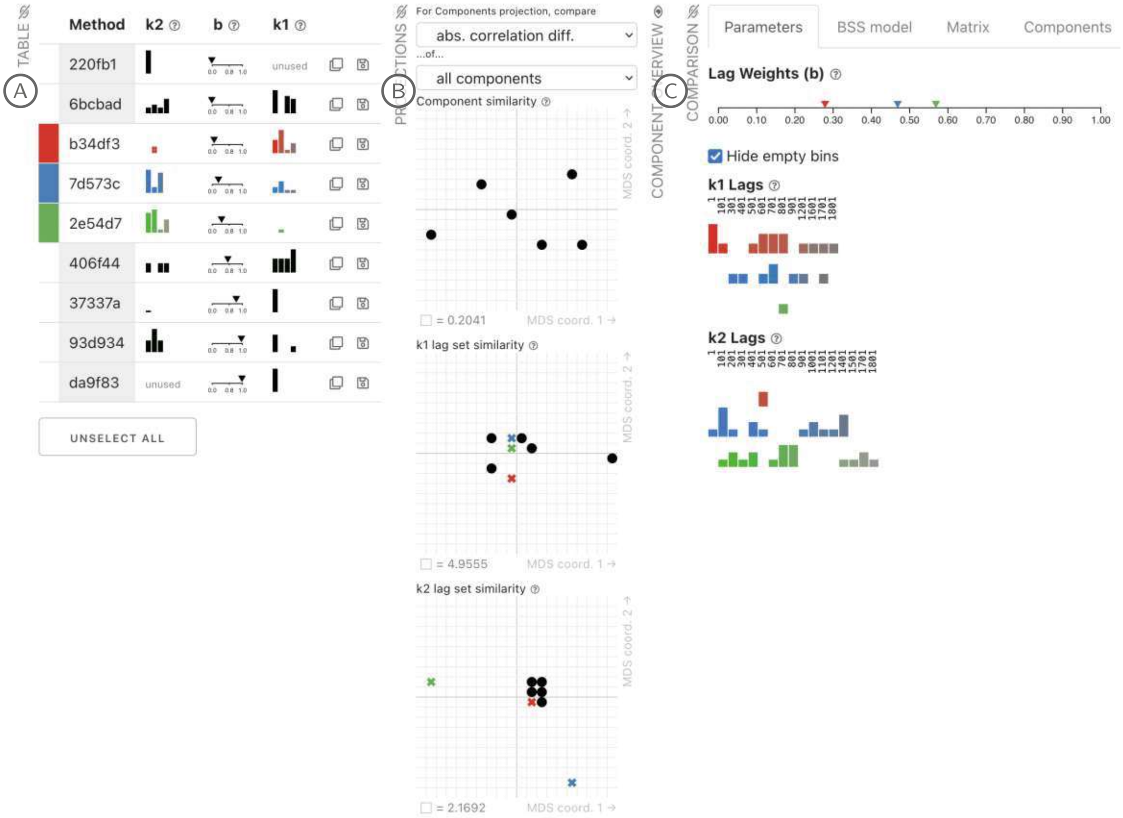
Parameter space analysis [Sed+14] is another important task for BSS experts, where they are mainly interested in *sensitivity analysis* and *partitioning*. We facilitate these tasks with tailored visualizations (Figure 3.5).

Similarity Views

Similarity of so far obtained component sets, as well as selected parameters, are shown in three separate dimensionally-reduced views. Marks that are close to each other suggest similar components and $\boldsymbol{k}_1/\boldsymbol{k}_2$ parameters. Multidimensional Scaling (MDS) is an appropriate dimensionality reduction technique for global cluster analysis according to recent publications [NA19; Xia+22]. We use non-metric MDS [VR10] as we do not always have a distance metric. As MDS will project elements with same values in high-dimensional space to the same low-dimensional points, we would soon run into an occlusion problem—consider an analyst who keeps lag sets the same, but varies only the weight. There are a couple of ways to deal with occlusion, most notably lenses [Tom+17]. However, our users are not used to complex interactions, so we changed the tradeoff between position accuracy and occlusion. As an implementation of CorrelatedMultiples [Liu+18] was not available, we only rasterize the MDS plot and move overlapping points to the next free cell. When hovering over a point, the other points will change their size proportionally to the original dissimilarity, thereby allowing analysts to investigate projection errors.

Parameter Comparison

To compare weights of different parametrizations, we encode triangle marks on a shared axis. Triangles are stacked if they would otherwise completely occlude each other. To compare lag sets, we use interweaved histograms where the color saturation of a bar encodes the lag size to give an additional visual hint of the lag distribution, and to be consistent with the encoding in the lag selection (Section 3.7.5). Figure 3.6 shows how they are generated. First, individual bars are positioned in a grid such that bars of the same lag set are in the same row, and bars representing the same bin are in the



	Method	k2	b	k1
	220fb1			unused
	6bcbad			
🟥	b34df3			
🟦	7d573c			
🟩	2e54d7			
	406f44			
	37337a			
	93d934			
	da9f83	unused		

UNSELECT ALL

For Components projection, compare
abs. correlation diff.
...of...
all components

Component similarity
□ = 0.2041

k1 lag set similarity
□ = 4.9555

k2 lag set similarity
□ = 2.1692

Parameters | BSS model | Matrix | Components

Lag Weights (b)

☑ Hide empty bins

k1 Lags

k2 Lags

Figure 3.5: *Ensemble* screen of TBSSvis (medical data) configured to facilitate comparison and inspection of parameters. Three failing runs were selected. Left column shows a tabular overview (A). Middle column shows DR projections of component and parameter similarities (B). Right column shows detailed comparison views to facilitate parameter comparison (C). It is apparent that failing runs had a weight parameter b of 0.25–0.6 and $\boldsymbol{k}_1/\boldsymbol{k}_2$ lag sets that span the whole range, which suggests that this parameter subspace should be avoided.

same column (base view). To save display space, empty columns are hidden by default (condensed view), but can be shown after user interaction. Increasing the bin size leads to familiar histogram shapes (aggregated view). Interweaved histograms show distinct images for same (bars align vertically and have similar height) and different lag sets (bars appear interweaved).

3.7.5 Task C5: Compare Possible Parameters

To obtain a new result, analysts need to select parameters. They consist in the case of gSOBI of two lag sets and one weight (Section 3.4).

To facilitate this selection process, we used the guidance design framework [Cen+20] to

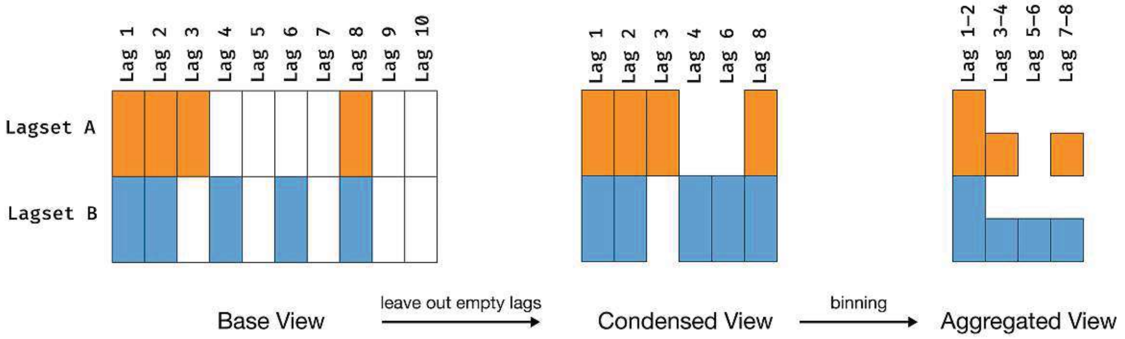


Figure 3.6: Interweaved histograms encoding two lag sets A and B facilitate comparison. The base version encodes the presence of a lag by filling the corresponding rectangle with color hue. The condensed version, which is the default, omits lags that are in no lag set. Finally, when increasing the bin size (pictured: to 2), the analyst sees the aggregated view, where rectangles are transformed in small bar charts.

design appropriate guidance [Cen+17]. Analysts do not know which lags to select and are generally aware of this *knowledge gap*. As discussed in Section 3.6.2, the *analysis goal* is to obtain a new/interesting result. Issues occur in the phase of lag selection, because the space of possible lag sets is huge. Analysts currently do not use additional information about lags, mostly due to time constraints. The *knowledge gap* lies in the execution and relates to the input data. We opt for *orienting* guidance, because analysts select lags also based on past experience and domain knowledge, so stronger guidance could be detrimental, and because our guidance *input* is not (cannot be) the "true" data: We compute it from the input data, which are per BSS model a linear combination of the components we are interested in. Based on the input data, we calculate guidance *output* per lag that help relate them to each other:

Guidance Output (GO) 1: Calendar relation. We compute which lag fits best to intervals in bigger calendar granules. The benefit of this is two-fold. First, lags are abstract and do not consider the calendar used in the data, so thinking in terms of days, weeks, etc., is a more intuitive alternative for someone familiar with the data. Second, it allows us to organize lags by filtering to those which correspond to a difference in a given calendar granule, thereby reducing the amount of lags to reason about.

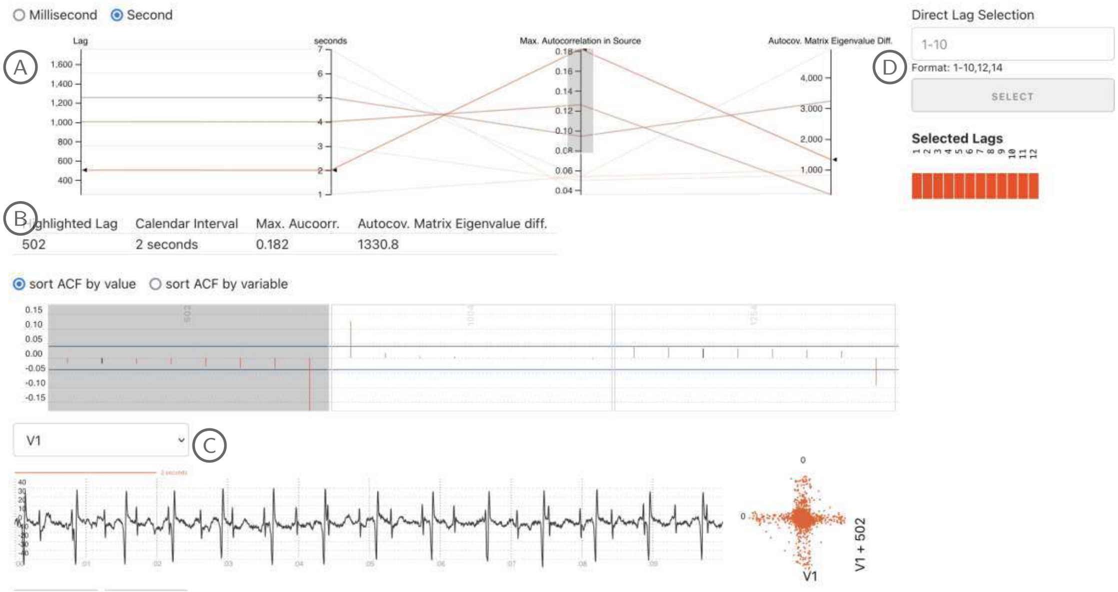
GO2: Largest autocorrelation in input time series. White noise is a serially uncorrelated process, i.e., does not exhibit autocorrelation, so this measure indicates a latent component might not be white noise.

GO3: Eigenvalue difference in autocovariance matrices. The analysts use it to learn more about the input data and it can inform the parameter selection as lags should be chosen such that this eigenvalue difference is big (see Section 3.4).

GO4: Cross-moment matrix diagonality. This can only be computed when a parametrization of a successful run is refined, i.e., an unmixing matrix estimate exists. It shows the analyst which selected lags had an impact on the diagonality of autocovariance and fourth cross-cumulant matrices. It can be understood as feedback into the guidance system.

Lag Selection

We support selection of a single lag set with multiple coordinated views (see Figure 3.7). The lag size is encoded with color saturation, to make long, medium, and short lags distinguishable in all views, which is roughly how analysts reason about lag sets.



○ Millisecond ● Second

Direct Lag Selection

1-10

Format: 1-10,12,14

SELECT

Selected Lags

Highlighted Lag	Calendar Interval	Max. Aucoorr.	Autocov. Matrix Eigenvalue diff.
502	2 seconds	0.182	1330.8

● sort ACF by value ○ sort ACF by variable

V1

Figure 3.7: The Lag Selection view (ECG data): Lag size is encoded with color saturation. Lags are filtered to those corresponding to a temporal difference of multiple seconds in the underlying calendar. The PCP (A) further narrows them down to those with high autocorrelation. A MACF plot (B) shows the autocorrelation of input data at brushed lags. Lags can be selected by clicking and highlighted by hovering in the MACF plot. A user-selected input time series is shown underneath (C) next to a scatterplot of the datapoints of the series at the currently highlighted lag. The right-most column (D) allows analysts to skip the interaction, and shows the current selection.

A parallel coordinates plot (PCP) displays all lags corresponding to a selected calendar granule (Figure 3.7-A), which can be configured by the user. Its dimensions are GO1–4, and values of selected lags are displayed as triangle marks next to the axes. The PCP supports common interactions such as inverting dimensions, reordering dimensions, and brushing. It is used to reduce the parameter space to a manageable subset. This subset is then visualized in a multivariate autocorrelation function plot (MACF), so analysts can view the temporal structure of all variables (Figure 3.7-B). The MACF shows all univariate autocorrelation function plots, composited through nesting: One box contains the autocorrelations of all variables at a given lag. Autocorrelations are encoded as bars, as in the univariate version, and can be sorted by variable name or by value. The latter is the default because it shows the distribution of autocorrelation values. Hovering over a box highlights the lag, which affects the next view below it. Clicking a box adds or removes the lag from/to the selection, which is shown in the right column in the same fashion as interweaved histograms (Figure 3.6).

Underneath the MACF we display a user-selected input time series as line graph, and a scatterplot of the time series' values vs. the values lagged by the currently highlighted lag (Figure 3.7-C). This allows the analyst to find correlation patterns which are not surfaced by the MACF or the time series itself. A line on top of the time series shows the extent of the currently highlighted lag in context.

These views allow the analyst to interactively explore possible lags. Should they exactly know what they want to select, or rather not use an interactive system because they are used to static tools, they can enter the desired lags in the input box in the right column (Figure 3.7-D) in a format similar to R's `seq` shorthand syntax and proceed.

3.7.6 Task C3/I2: Compare Unmixing Matrices

We support this task (I2) by showing the factors as a heatmap where a univariate color scale encodes the absolute value in a row with white (low value) to black. When analysts see interesting patterns, they can select cells, and the respective input data and components will be shown underneath the matrices (Figure 3.8-H). This allows to investigate the relationship between inputs and components. Task C3 is also supported, for which we encode a BSS-specific similarity measure [Ilm+10] in a heatmap with a univariate color scale.

3.8 Usage Scenarios

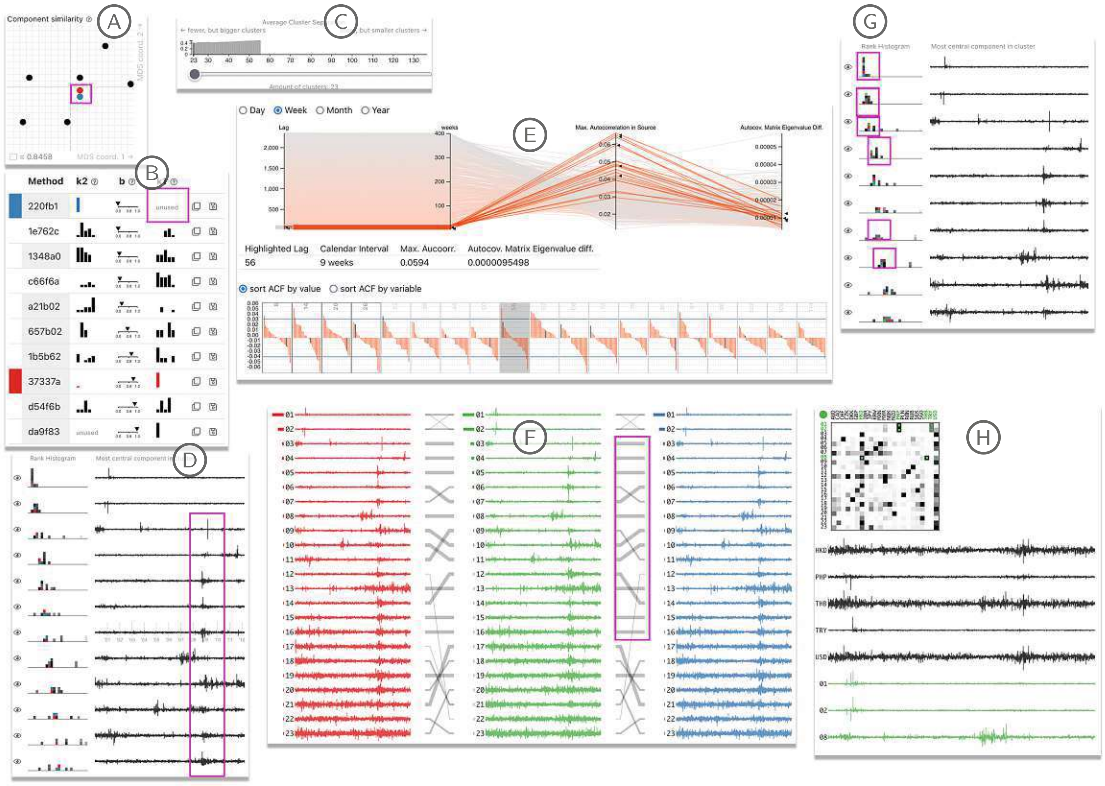
In this section we describe how the designed visualizations (Section 3.7) provide insights into the presented datasets (Section 3.5). The financial dataset was used in our user studies (Section 3.9), while we added the medical dataset ourselves to provide broader context to the reader. The usage scenarios we describe are based on what we learned during aforementioned user studies and also during discussions with our collaborators.

3.8.1 Financial data

We load the financial dataset (Section 3.5.1) of 23 currency exchange rates to Euro into TBSSvis and start with 10 parameter settings. From the *Component Similarity View* (Figure 3.8-A, Section 3.7.4) we can immediately see that two component sets are very similar as they are very close to each other (purple highlight) and hovering does not change their relative sizes. Selecting them reveals that one of them did not use the $\boldsymbol{k}_1$ lag set at all (Figure 3.8-B), suggesting that this parameter's influence is small and we should focus on $\boldsymbol{k}_2$ when selecting parameters. This is in line with our expectations of financial data (Section 3.4). Looking at the clustering visualizations, no clear picture emerges. The *Clustering Quality* (Line 14) increases slowly with the number of clusters, but there is no distinctive peak (Figure 3.8-C). Thus, we expect that all components are somewhat similar to each other. Inspection of the *Component Overview* (Line 14) confirms that, as many components share a similar pattern: They are very noisy with more extreme values during the years 2008–2009 (Figure 3.8-D, purple highlight marks years 2008–2009). This was the time of the global financial crisis. We try to obtain an alternative result and go to the *Parameter Selection*. We set the weight b to zero and do not use SOBI part ($\boldsymbol{k}_1$ parameter) at all, following our initial hypothesis. In the *Lag Selection* (Figure 3.8-E, Section 3.7.5) for $\boldsymbol{k}_2$ we quickly select lags that correspond to 1–3 days, 1–4 weeks, 1–3 months and 1 year intervals in the underlying calendar. We do it this way because available guidance outputs do not seem informative due to the amount of noise in the dataset. E.g., the autocorrelation (GO2) of weekly lags is very low (at most 0.06). The newly computed result is colored green in TBSSvis and automatically selected. We look at its components and compare it to the two identical results. The *Slope Graph* (Figure 3.8-F, Line 14) shows many thick lines that connect identical components (purple highlight). As we want to find currencies to invest in, we turn to the *Component Overview* again. The histograms show stacked and saturated bars, thus suggesting that the first couple of components are stable and common in all results (Figure 3.8-G). We, therefore, pick three that have volatile segments outside of 2008–2009 to rule out a global financial crisis as the cause for volatility. The *Unmixing Matrix* visualization (Section 3.7.6) shows which currencies are associated with these components (Figure 3.8-H, black time series). We will ask our financial advisor about investing in Thai bhat, US dollars, Turkish lira, or Philippine pesos.

3.8.2 Medical data

We load the ECG dataset (Section 3.5.2) from a pregnant woman into TBSSvis. Looking at the raw inputs in the *Input Visualization* we can confirm that the fetal heart signal is visible in the mother's ECG (Figure 3.9-A, purple highlight). We start with 10 precomputed parameter settings, 7 of which succeed. The *Clustering Quality* (Figure 3.9-B, Line 14) suggests that 8–11 meaningfully different components were obtained, as the height of bars steadily declines afterwards. We set the clustering to 10 partitions. A healthy fetus has a heart rate of 110–160 beats/minute on average, which is higher than that of an adult (60–100). A candidate component for the fetal heart signal, which shows peaks of increased frequency, is readily visible as 4th (sorted by kurtosis) in the *Component Overview* (Figure 3.9-C, Line 14). The rank histogram next to the cluster medoid shows that components in the cluster are very similar, as all boxes are quite black, and it can be confirmed by looking at components directly (Figure 3.9-C). We select a couple of results containing this component to compare their parameters. We see that the parameters vary wildly (Figure 3.9-D), and the fetal heart signal was found using long and short lags for either lag set with different weights. This, along with the absence of other candidate components, suggests that we found the correct signal. A medical doctor would be able to inspect the obtained fetal ECG wave patterns in detail


A
Component similarity
MDS coord. 1 →
MDS coord. 2 →
≤ 0.8458
B
C
Average Cluster Separation
← fewer, but bigger clusters
more, but smaller clusters →
Amount of clusters: 23
D
Rank Histogram
Most central component in cluster
E
○ Day ● Week ○ Month ○ Year
Lag
weeks
Max. Autocorrelation in Source
Autocov. Matrix Eigenvalue Diff.
● sort ACF by value ○ sort ACF by variable
F
G
H
HKD
PHP
THB
TRY
USD

Method	k2	b	k3
220fb1			unused
1e762c			
1348a0			
c66f6a			
a21b02			
657b02			
1b5b62			
37337a			
d54f6b			
da9f83	unused		

Highlighted Lag	Calendar Interval	Max. Aucoorr.	Autocov. Matrix Eigenvalue diff.
56	9 weeks	0.0594	0.0000095498

Figure 3.8: Usage scenario on financial dataset, see Section 3.8.1 for details.

and determine whether or not it is healthy.

Looking at the values of the three parameter settings that did not produce results, we can also form an initial hypothesis about the useful parameter subspace (Figure 3.9-E). The weight *b* alone did not seem to play too much of a role as values span a wide range (0.11–0.94) and we do have several successful results within that range. But an apparent difference to those parameter settings is that both lag sets in failing results had lags that were distributed over the whole range instead of sticking to either the short or long end. This can be seen from Figure 3.9-D and E, as lags of the former appear blocked, whereas they are more interweaved in the latter. Thus, when trying to find new parameters for this dataset, we would steer clear of such lag sets.

3.9 Evaluation

To assess the usefulness of our visualization design, we conducted two interviews with five TBSS experts external to the project. Our research questions were:

RQ1 What are advantages and disadvantages of TBSSvis in comparison to their current tools?

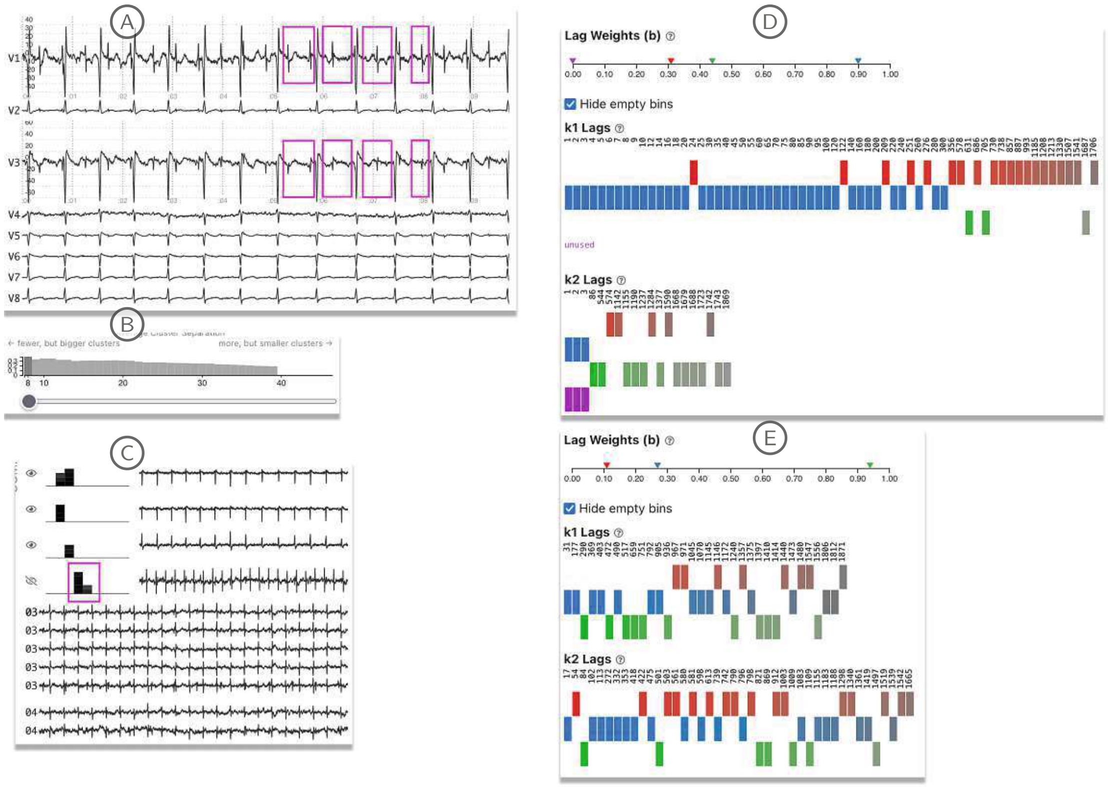


Figure 3.9: Usage scenario on medical dataset, see Section 3.8.2 for details. Note that (D) and (E) show different parameter settings.

RQ2 Does TBSSvis in fact support the analysis tasks?

RQ3 What are possible improvements to TBSSvis?

We decided for an *Expert Review* [EY15] using interviews, as no comparable tool for a quantitative evaluation exists and qualitative data allows much deeper insights. Two interview cycles were conducted: In the first we gathered initial external feedback and supporting evidence for our task abstraction, and in the second we verified that this feedback was integrated accordingly. They lasted 2.5 hours and 1 hour, respectively.

3.9.1 Participants

Participants were the same for both interviews and previous collaborators of our co-authors. They participated voluntarily without promised benefits, financial or other. All are adults and not dependent on any author, be it financially, professionally or personally.

Our five experts (E1–E5) all hold a Ph.D. degree in mathematics or statistics. Four obtained their Ph.D. with research in BSS somewhat recently, while the other researches

BSS already for 10 years. Therefore, they more than fit the *basic knowledge of BSS* and *formal education in math/statistics* requirements from our user characterization (Section 3.6.1). Since participants applied (T)BSS on diverse datasets and collaborated with various domain experts both in the context of (T)BSS (e.g., genome biology, cancer research) as well as outside of it (e.g., ecology, neurology), we think they are very well suited to answer our research questions. Although they cannot provide us with deep data-related insights as they are not application domain experts, they are our primary intended users and bring sufficient experience and a broad perspective to our research questions around TBSS analysis and involved tasks. This helps us to keep TBSSvis generic, yet effective, as was our design goal (Section 3.7).

No participant uses visual-interactive tools regularly. Their self-assessed experience in visualization is “basic”, as E4 put it: “I only use what R has to offer, like ggplot and the base graphics (...) scatterplots, time series plots, (...) box plots. I tend to stick with these basic kinds of plots (...)”.

3.9.2 Methodology

The interviews were conducted and recorded via Zoom with explicit consent by participants.[1] Two researchers were involved in each interview, one tasked with moderation and one took notes. Participants used TBSSvis on their own machines and shared their screen during usage. We used Zoom annotations to point out relevant parts of TBSSvis when necessary.

Both sessions were structured the same. We compiled a text explanation with images of TBSSvis, so that participants can familiarize themselves with it beforehand. The tutorial document was sent to participants together with the consent form ahead of the interview. Steps during the interview were as follows:

1. (Only in first session.) We conducted a structured interview about their background and experiences with (T)BSS (15–30 minutes).

2. We gave participants a structured introduction to interactions and visualizations in TBSSvis (up to 1–hour). The dataset used was synthetic and unfamiliar to them. We asked participants to solve small tasks to practice what we explained. We skipped these tasks when we either saw that they understood it, or when we were short on time.

3. (Optional.) Participants were allowed to further use TBSSvis for some minutes on their own.

[1] As of manuscript submission, the TU Wien has a Pilot Research Ethics Committee. Approaching it for peer review of research with human participants is not required by the TU Wien, and its response is non-binding. Therefore we do not provide an official ethics approval. Nonetheless, we believe we conducted our research adhering to sufficient ethical standards.

4. We asked participants to conduct an open analysis on the dataset used in Section 3.5.1, which most have worked on in the past, and articulate their thoughts and plans ("think aloud"). We pointed out parts of TBSSvis they did not use or consider so far. This step took around 30–minutes.

5. We discussed tasks, visualizations, interactions and possible further improvements in an unstructured fashion (15–30 minutes). Before we finished the session, we encouraged participants to use TBSSvis more without our supervision.

To answer RQ2 we found it sufficient to check whether or not participants can interpret our visualizations, and if visualizations show the necessary data in the right moment to support their tasks. To do so, we analyzed the recorded video and notes after each session. We looked for articulated suggestions, discussions, and situations where users interacted with visualizations. These instances were transcribed and grouped by tasks (Section 3.6.2). Feedback and possible issues of participants were noted, deduplicated, and presented to our collaborators. Subsequent discussions then informed changes to the first design, which we confirmed in the second interview.

The interview guide, tutorial documents, datasets, and our transcripts of the interviews are available as supplementary material.

3.9.3 Expert Feedback

We describe evidence for our research questions in this section.

RQ1: Advantages and Disadvantages

Our participants agreed that TBSSvis has clear advantages compared to current tools used and greatly improves the analysis process. E5 even said that TBSSvis is "an absolute time saver" and "very useful for applied work." The majority of them mentioned that it is easier than in RStudio to compare components, matrices, and parameters. The same outcomes can be achieved faster in TBSSvis and it provides useful new visualizations they could not have in RStudio, such as the component overview/comparison. All participants mentioned to enjoy playing with TBSSvis. Even in our limited time we saw indications how TBSSvis can change the way they work. We observed E4 in the open session to pursue an analysis process resembling binary search, toggling individual lags on and off. Asked about it later, E4 mentioned to be "not sure if I'd have thought about [this approach] just with RStudio." E5 was very eager to get hold of TBSSvis, as the intention was to recommend it to their students. E1 stated that better supported comparison tasks give more structure to the analysis process, so all this suggests that TBSSvis allows new or more streamlined analyses.

As for disadvantages, there is one very basic: RStudio allows more flexible and specialized computations than TBSSvis. However, this was not explicitly mentioned by participants. Some said it took time to put everything together, but all our participants managed to do so quickly. A few plots were difficult to understand at first, but after explanations it was relatively easy to use for all participants. In addition, we observed some participants having trouble with idioms that are common and popular in the visualization community, such as PCPs and multiple linked views, which could be overcome by visual literacy efforts.

RQ2: Supported Tasks

In this section, we discuss how TBSSvis supports analysis tasks (Section 3.6). We provide quotes from participants to let them speak for themselves, but their sentiment is shared by the majority and not an isolated opinion.

Identify used parameters (I1): The tabular overview (Figure 3.5-A) was considered "really useful" (E2) and participants thought it "makes a lot of sense" (E4).

Identify unmixing matrix (I2): Participants could easily identify similar matrices and dominant factors of components. Viewing involved time series (Figure 3.8-H) was considered useful.

Identify cross-moment diagonality (I3): It is "something I don't usually have the time and energy to compute" (E5) and "very interesting" (E1), but also something they do not regularly use for their analysis today.

Identify components (I4): Our participants found the added interactivity compared to RStudio very useful.

Compare success (C1): They had no trouble with visual encodings, but participants sometimes forgot that failure is an option.

Compare parameters (C2): While the interweaved lag histograms were easy to interpret, it took some time for participants to realize that it is a regular histogram with hidden bins (Figure 3.6). Similarity projections of parameters (Figure 3.5-B) were rarely used by our participants. A possible explanation is because histograms show more data and participants worked with only 5–7 parametrizations, they could use their working memory. We believe their benefits would have become apparent with more parametrizations.

Compare unmixing matrices (C3): Some (E3–E5) mentioned that interpreting the MD-Index for other than extreme values is not easy as it depends on the data dimensionality. While both visualizations were used by all, some participants seemed to prefer the MD-Index (E3, E4) over the factors (E2) to compare matrices.

Compare component sets (C4): Participants understood the slope graph (Figure 3.4) easily and immediately saw its benefits. E3 mentioned that using it is "easier than looking at a correlation matrix." The projection view was in fact used to see how similar ensemble members are. For this purpose participants also appreciated the component overview ("you can very fast get an idea of how similar different methods are", E3), although most did not change the initial clustering parameter.

Compare possible parameters (C5): After we introduced participants to individual views and interactions, they learned quickly how to use it and found it useful and convenient. They understood how and why to filter visualized lags, but were not sure about the data-driven calendar-based approach, presumably because they currently analyze data detached from any calendar. Participants appreciated the PCP with its dimensions, even though they sometimes did not know right away how to interpret all of them: For example, E2 asked what the eigenvalue metric means, what the optimal choice is, and if lower or higher is better. Participants were also sometimes irritated by the number of dimensions, as they depend on the outcome of the refined run.

RQ3: Possible Improvements

When asked about improvements to TBSSvis, we got responses mainly pertaining to the parameter selection. E4 would prefer if the syntax to directly select lags matched commands available in R. E2–E4 often ended up with an empty selection in the PCP because they expected brushes to be combined with union instead of intersection. They also want to select all filtered lags and remove all selected lags at once. Aside from the lag selection improvements, more DOI functions would be appreciated. We added one measure for periodicity [VYC05] following the suggestion of one participant. E5 suggested to support loading precomputed results, possibly from other TBSS methods. E2 asked for more legends, explanations, and a stronger guidance degree in general. E1 suggested the ability to freely reorder components everywhere, and providing alternative color palettes. With E1 we also discussed the option of showing correlations between input data in the *Input Visualization* screen as another sanity check.

3.10 Reflection and Discussion

Reflecting on our findings and lessons learned during our design study with experts in BSS, we claim that TBSSvis supports tasks involved with TBSS analysis (Section 3.6) and encourages usage of TBSS in various application domains. Despite differences in what an application domain considers interesting in latent dimensions (e.g., doctors might search for specific wave patterns, while investors look for sudden and extreme changes), many tasks are the same. We showed this transferability to financial and medical datasets in Section 3.8. We developed and evaluated TBSSvis with TBSS experts, who are our primary intended users. They worked with many domain experts in the past to apply TBSS in their respective fields. Their practical experience with different use cases for TBSS informed our visualization design (Section 3.7). Therefore, based on the mostly positive feedback by our interview participants, we expect that TBSSvis can be useful in many application domains.

In line with the design study methodology [SMM12], we used well known visualization idioms and data mining algorithms, applied them in a new context and extended them as necessary. As a consequence, individual parts of TBSSvis will be useful to other visualization researchers and designers. For instance, a slope graph usually shows categorical data cases and their change of rank by line slope. We adapted it to time series by encoding similarity in line thickness. In our user studies it was considered an easy-to-understand visualization to visually compare sets of time series. The clustering scheme (Section 3.7.3) is useful whenever members of sets should be clustered and set membership must be taken into account. It works with any dissimilarity measure because it is based on k-medoids. Set-typed data is prevalent [Als+16], so we expect this to be useful to others.

3.10.1 Design Process

Following the recommendations of the data-users-tasks design triangle [MA14] our proposed visualizations are close to what TBSS experts are used to and therefore quite simple. We also did not include more advanced interactions than highlighting, filtering, hovering, or brushing because TBSS experts come from a text-based software where even these do not exist. Looking back, we think this was a good decision, as in our interviews some participants had initially trouble using, e.g., the PCP.

What was difficult for us visualization researchers during the design is the domain-independence of TBSS. Our goal, therefore, was to make TBSSvis applicable in a wide range of domain-specific contexts, e.g., in medicine or finance. But both size and complexity of the data vary considerably among the domains, as do the definitions of "interesting" features and the location and role of TBSS in the data processing pipeline [vLFR17]. Therefore, we opted in the end for simple interactions and generic/extendable approaches, such as the use of DOI functions, to avoid a "lock-in" to any specific application domain.

3.10.2 Limitations and Future Work

We discuss some limitations in our paper. Most study participants used the financial dataset (Section 3.5.1) at some point in the past to test varying TBSS methods. Although participants fit well to our user description (Section 3.6.1), they were not as intimately familiar with the dataset as it is often the case in visualization-related evaluations. Had this been the case, we may have found additional analysis goals and insights. Nevertheless, we maintain that our study methodology and participant selection was sufficient and appropriate to investigate how TBSSvis impacts involved tasks (Section 3.6.2). Participants used TBSSvis in both interviews for in total around 45 minutes on their own terms. More time using it may have surfaced additional necessary analysis tasks or improvement suggestions.

As part of our future work, we would like to integrate the suggested improvements by our experts, support larger datasets and allow provision of custom DOI functions.

3.11 Summary and Conclusion

We presented TBSSvis, a VA solution for TBSS. TBSS is in a way similar to PCA, in that it can be used to analyze suitable datasets from any application domain, such as biomedical analysis, finance, or civil engineering. Unlike PCA, TBSS properly accounts for temporal correlation and requires complex tuning parameters. Because of these parameter settings, TBSS analysis is inherently open-ended and exploratory as there are no known insights to confirm. TBSSvis is based on a task abstraction and visualization design that we developed together in a user-centered design process with TBSS experts. We evaluated the final interactive prototype with five other TBSS experts, who did not participate in the design process, by conducting two interviews. Feedback from these shows that TBSSvis supports the actual workflow and combination of interactive visualizations that facilitate the tasks involved in analyzing TBSS results—this process was previously a laborious back-and-forth for which analysts had to manually program static visualizations and data mining algorithms. TBSSvis also provides guidance to facilitate the analysis of the data at hand and informed parameter selection, which was previously mostly a guessing game.

Acknowledgements

We would like to thank our experts for spending so much of their valuable time on discussions and evaluations with us. This work was supported by the Austrian Science Fund (FWF) under grant P31881-N32.

3.12 Bibliography

[Aig+11] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. 1st edition. Berlin, Heidelberg: Springer, 2011. 286 pp. ISBN: 978-0-85729-078-6 (cit. on p. 108).

[Als+16] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers. "The State-of-the-Art of Set Visualization". In: *Computer Graphics Forum* 35.1 (2016), pp. 234–260. DOI: 10.1111/cgf.12722 (cit. on p. 133).

[AA16] J. P. Amezquita-Sanchez and H. Adeli. "Signal Processing Techniques for Vibration-Based Health Monitoring of Smart Structures". In: *Archives of Computational Methods in Engineering* 23.1 (2016), pp. 1–15. DOI: 10.1007/s11831-014-9135-7 (cit. on p. 106).

[AWD12] A. Anand, L. Wilkinson, and T. N. Dang. "Visual Pattern Discovery Using Random Projections". In: *IEEE Conference on Visual Analytics Science and Technology*. New York, NY, USA: IEEE, 2012, pp. 43–52. DOI: 10.1109/vast.2012.6400490 (cit. on p. 110).

[Bel+97] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines. “A Blind Source Separation Technique Using Second-Order Statistics”. In: *IEEE Transactions on Signal Processing* 45.2 (1997), pp. 434–444. DOI: 10.1109/78.554307 (cit. on p. 111).

[Bög+13] M. Bögl, W. Aigner, P. Filzmoser, T. Lammarsch, S. Miksch, and A. Rind. “Visual Analytics for Model Selection in Time Series Analysis”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2237–2246. DOI: 10.1109/tvcg.2013.222 (cit. on p. 110).

[BM13] M. Brehmer and T. Munzner. “A Multi-Level Typology of Abstract Visualization Tasks”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2376–2385. DOI: 10.1109/tvcg.2013.124 (cit. on pp. 105, 113, 114).

[Buc+19] J. Buchmüller, D. Jäckle, E. Cakmak, U. Brandes, and D. A. Keim. “MotionRugs: Visualizing Collective Trends in Space and Time”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 76–86. DOI: 10.1109/tvcg.2018.2865049 (cit. on p. 109).

[Cen+17] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H. Schulz, M. Streit, and C. Tominski. “Characterizing Guidance in Visual Analytics”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 111–120. DOI: 10.1109/tvcg.2016.2598468 (cit. on pp. 114, 123).

[Cen+20] D. Ceneda, N. Andrienko, G. Andrienko, T. Gschwandtner, S. Miksch, N. Piccolotto, T. Schreck, M. Streit, J. Suschnigg, and C. Tominski. “Guide Me in Analysis: A Framework for Guidance Designers”. In: *Computer Graphics Forum* 39.6 (2020), pp. 269–288. DOI: 10.1111/cgf.14017 (cit. on p. 122).

[Cho+10] J. Choo, H. Lee, J. Kihm, and H. Park. “iVisClassifier: An Interactive Visual Analytics System for Classification Based on Supervised Dimension Reduction”. In: *IEEE Conference on Visual Analytics Science and Technology*. New York, NY, USA: IEEE, 2010, pp. 27–34. DOI: 10.1109/vast.2010.5652443 (cit. on p. 110).

[CJ10] P. Comon and C. Jutten. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Academic Press, 2010 (cit. on pp. 106, 111).

[dLdMV00] L. de Lathauwer, B. de Moor, and J. Vandewalle. “Fetal Electrocardiogram Extraction by Blind Source Subspace Separation”. In: *IEEE Transactions on Biomedical Engineering* 47.5 (2000), pp. 567–572. DOI: 10.1109/10.841326 (cit. on pp. 106, 113).

[Din+19] D. Dingen, M. van't Veer, P. Houthuizen, E. H. J. Mestrom, E. H. Korsten, A. R. Bouwman, and J. van Wijk. "RegressionExplorer: Interactive Exploration of Logistic Regression Models with Subgroup Analysis". In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 246–255. DOI: 10.1109/tvcg.2018.2865043 (cit. on p. 110).

[EY15] N. Elmqvist and J. S. Yi. "Patterns for visualization evaluation". In: *Information Visualization* 14.3 (2015), pp. 250–269. DOI: https://doi.org/10.1177/1473871613513228 (cit. on p. 128).

[Fer+17] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann. "Time-Hierarchical Clustering and Visualization of Weather Forecast Ensembles". In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 831–840. DOI: 10.1109/tvcg.2016.2598868 (cit. on p. 109).

[Fu11] T.-c. Fu. "A Review on Time Series Data Mining". In: *Engineering Applications of Artificial Intelligence* 24.1 (2011), pp. 164–181. DOI: 10.1016/j.engappai.2010.09.007 (cit. on p. 117).

[Gog+19] A. Gogolou, T. Tsandilas, T. Palpanas, and A. Bezerianos. "Comparing Similarity Perception in Time Series Visualizations". In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 523–533. DOI: 10.1109/tvcg.2018.2865077 (cit. on p. 108).

[HHB16] L. Hao, C. G. Healey, and S. A. Bass. "Effective Visualization of Temporal Ensembles". In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 787–796. DOI: 10.1109/tvcg.2015.2468093 (cit. on p. 109).

[HHH15] L. Hao, C. G. Healey, and S. E. Hutchinson. "Ensemble Visualization for Cyber Situation Awareness of Network Security Data". In: *IEEE Symposium on Visualization for Cyber Security (VizSec)*. New York, NY, USA: IEEE, 2015, pp. 1–8. ISBN: 978-1-4673-7599-3. DOI: 10.1109/vizsec.2015.7312766 (cit. on p. 109).

[Ilm+10] P. Ilmonen, K. Nordhausen, H. Oja, and E. Ollila. "A New Performance Index for ICA: Properties, Computation and Asymptotic Analysis". In: *Latent Variable Analysis and Signal Separation*. Ed. by V. Vigneron, V. Zarzoso, E. Moreau, R. Gribonval, and E. Vincent. Berlin, Heidelberg: Springer, 2010, pp. 229–236. ISBN: 978-3-642-15995-4. DOI: 10.1007/978-3-642-15995-4_29 (cit. on pp. 115, 125).

[JME10] W. Javed, B. McDonnel, and N. Elmqvist. "Graphical Perception of Multiple Time Series". In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 927–934. DOI: 10.1109/tvcg.2010.162 (cit. on p. 108).

[Kei+08] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. “Visual Analytics: Definition, Process, and Challenges”. In: *Information Visualization*. Ed. by A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North. 4950. Berlin, Heidelberg: Springer, 2008, pp. 154–175. ISBN: 978-3-540-70955-8 978-3-540-70956-5. DOI: 10.1007/978-3-540-70956-5_7 (cit. on p. 107).

[Kin10] R. Kincaid. “SignalLens: Focus+Context Applied to Electronic Time Series”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 900–907. DOI: 10.1109/tvcg.2010.193 (cit. on p. 109).

[Köt+15] P. Köthur, C. Witt, M. Sips, N. Marwan, S. Schinkel, and D. Dransch. “Visual Analytics for Correlation-Based Comparison of Time Series Ensembles”. In: *Computer Graphics Forum* 34.3 (2015), pp. 411–420. DOI: 10.1111/cgf.12653 (cit. on p. 109).

[Liu+19] X. Liu, H. Wang, M. Huang, and W. Yang. “An Improved Second-Order Blind Identification (SOBI) Signal De-Noising Method for Dynamic Deflection Measurements of Bridges Using Ground-Based Synthetic Aperture Radar (GBSAR)”. In: *Applied Sciences* 9.17 (2019), p. 3561. DOI: 10.3390/app9173561 (cit. on p. 116).

[Liu+18] X. Liu, Y. Hu, S. North, and H.-W. Shen. “CorrelatedMultiples: Spatially Coherent Small Multiples With Constrained Multi-Dimensional Scaling”. In: *Computer Graphics Forum* 37.1 (2018), pp. 7–18. DOI: 10.1111/cgf.12526 (cit. on pp. 108, 121).

[Mac86] J. Mackinlay. “Automating the Design of Graphical Presentations of Relational Information”. In: *ACM Transactions on Graphics* 5.2 (1986), pp. 110–141. DOI: 10.1145/22949.22950 (cit. on pp. 108, 117).

[Mat+17] M. Matilainen, J. Miettinen, K. Nordhausen, H. Oja, and S. Taskinen. “On Independent Component Analysis with Stochastic Volatility Models”. In: *Austrian Journal of Statistics* 46.3-4 (2017), pp. 57–66. DOI: 10.17713/ajs.v46i3-4.671 (cit. on p. 111).

[Mat+09] K. Matkovic, D. Gracanin, B. Klarin, and H. Hauser. “Interactive Visual Analysis of Complex Scientific Data as Families of Data Surfaces”. In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (2009), pp. 1351–1358. DOI: 10.1109/tvcg.2009.155 (cit. on p. 109).

[MGH18] K. Matković, D. Gračanin, and H. Hauser. “Visual Analytics for Simulation Ensembles”. In: *Proceedings of the Winter Simulation Conference*. New York, NY, USA: IEEE, 2018, pp. 321–335. DOI: 10.1109/wsc.2018.8632312 (cit. on p. 109).

[McL+08] P. McLachlan, T. Munzner, E. Koutsofios, and S. North. “LiveRAC: Interactive Visual Exploration of System Management Time-series Data”. In: *Proceedings of the 2008 CHI Conference on Human Factors in Computing Systems* (Florence, Italy). New York, NY, USA: ACM, 2008, pp. 1483–1492. ISBN: 978-1-60558-011-1. DOI: `10.1145/1357054.1357286` (cit. on p. 108).

[Mie+16] J. Miettinen, K. Illner, K. Nordhausen, H. Oja, S. Taskinen, and F. Theis. “Separation of Uncorrelated Stationary Time Series Using Autocovariance Matrices”. In: *Journal of Time Series Analysis* 37.3 (2016), pp. 337–354 (cit. on p. 111).

[Mie+20] J. Miettinen, M. Matilainen, K. Nordhausen, and S. Taskinen. “Extracting Conditionally Heteroskedastic Components Using Independent Component Analysis”. In: *Journal of Time Series Analysis* 41.2 (2020), pp. 293–311. DOI: `10.1111/jtsa.12505` (cit. on pp. 107, 111, 112).

[Mie+14] J. Miettinen, K. Nordhausen, H. Oja, and S. Taskinen. “Deflation-based separation of uncorrelated stationary time series”. In: *Journal of Multivariate Analysis* 123 (2014), pp. 214–227. DOI: `https://doi.org/10.1016/j.jmva.2013.09.009` (cit. on p. 111).

[MA14] S. Miksch and W. Aigner. “A Matter of Time: Applying a Data-users-tasks Design Triangle to Visual Analytics of Time-oriented Data”. In: *Computers & Graphics* 38 (2014), pp. 286–290. DOI: `10.1016/j.cag.2013.11.002` (cit. on pp. 113, 133).

[MP13] T. Mühlbacher and H. Piringer. “A Partition-Based Framework for Building and Validating Regression Models”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 1962–1971. DOI: `10.1109/tvcg.2013.125` (cit. on p. 110).

[Mun09] T. Munzner. “A Nested Model for Visualization Design and Validation”. In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (2009), pp. 921–928. DOI: `10.1109/tvcg.2009.111` (cit. on pp. 107, 113).

[NA19] L. G. Nonato and M. Aupetit. “Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Enrichment”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.8 (2019), pp. 2650–2673. DOI: `10.1109/tvcg.2018.2846735` (cit. on p. 121).

[Nor+21] K. Nordhausen, M. Matilainen, J. Miettinen, J. Virta, and S. Taskinen. “Dimension Reduction for Time Series in a Blind Source Separation Context Using R”. In: *Journal of Statistical Software, Articles* 98.15 (2021), pp. 1–30. DOI: `https://doi.org/10.18637/jss.v098.i15` (cit. on p. 112).

[NO18] K. Nordhausen and H. Oja. “Independent component analysis: A statistical perspective”. In: *WIREs Computational Statistics* 10.5 (2018), e1440. DOI: `https://doi.org/10.1002/wics.1440` (cit. on p. 111).

[NR22] K. Nordhausen and A. Ruiz-Gazen. “On the usage of joint diagonalization in multivariate statistics”. In: *Journal of Multivariate Analysis* 188 (2022), p. 104844. DOI: `https://doi.org/10.1016/j.jmva.2021.104844` (cit. on p. 111).

[OBJ16] H. Obermaier, K. Bensema, and K. I. Joy. “Visual Trends Analysis in Time-Varying Ensembles”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.10 (2016), pp. 2331–2342. DOI: `10.1109/tvcg.2015.2507592` (cit. on p. 109).

[OKM00] E. Oja, K. Kiviluoto, and S. Malaroiu. “Independent Component Analysis for Financial Time Series”. In: *Proceedings of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*. New York, NY, USA: IEEE, 2000, pp. 111–116. DOI: `10.1109/asspcc.2000.882456` (cit. on pp. 106, 116).

[Pan+] Y. Pan, M. Matilainen, S. Taskinen, and K. Nordhausen. “A review of second-order blind identification methods”. In: *WIREs Computational Statistics* (). First published Feb. 2021, to appear. DOI: `https://doi.org/10.1002/wics.1550` (cit. on p. 111).

[PJ09] H.-S. Park and C.-H. Jun. “A Simple and Fast Algorithm for K-medoids Clustering”. In: *Expert Systems with Applications* 36.2 (2009), pp. 3336–3341. DOI: `10.1016/j.eswa.2008.01.039` (cit. on pp. 118, 119).

[Pir+12] H. Piringer, S. Pajer, W. Berger, and H. Teichmann. “Comparative Visual Analysis of 2D Function Ensembles”. In: *Computer Graphics Forum* 31.3pt3 (2012), pp. 1195–1204. DOI: `10.1111/j.1467-8659.2012.03112.x` (cit. on p. 109).

[Pot+09] K. Potter, A. Wilson, P.-T. Bremer, D. Williams, C. Doutriaux, V. Pas, and C. R. Johnson. “Ensemble-Vis: A Framework for the Statistical Visualization of Ensemble Data”. In: *Proceedings ICDM Workshops*. New York, NY, USA: IEEE, 2009, pp. 233–240. ISBN: 978-1-4244-5384-9. DOI: `10.1109/icdmw.2009.55` (cit. on p. 109).

[QH18] Z. Qu and J. Hullman. “Keeping Multiple Views Consistent: Constraints, Validations, and Exceptions in Visualization Authoring”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2018), pp. 468–477. DOI: `10.1109/tvcg.2017.2744198` (cit. on p. 117).

[R C23] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2023 (cit. on p. 114).

[Ros+20] P. Rosen, A. Suh, C. Salgado, and M. Hajij. “TopoLines: Topological Smoothing for Line Charts”. In: *EuroVis 2020 - Short Papers*. Geneve, Switzerland: Eurographics Association, 2020, pp. 85–89. DOI: `10.2312/evs.20201053` (cit. on p. 109).

[SR19] E. Schubert and P. J. Rousseeuw. "Faster K-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms". In: *Proceedings of the International Conference on Similarity Search and Applications*. Vol. 11807. Berlin, Heidelberg: Springer, 2019, pp. 171–187. ISBN: 978-3-030-32047-8. DOI: 10.1007/978-3-030-32047-8_16 (cit. on p. 119).

[Sch+19] M. Schwab, S. Hao, O. Vitek, J. Tompkin, J. Huang, and M. A. Borkin. "Evaluating Pan and Zoom Timelines and Sliders". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2019, pp. 1–12. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300786 (cit. on p. 117).

[SMM12] M. Sedlmair, M. Meyer, and T. Munzner. "Design Study Methodology: Reflections from the Trenches and the Stacks". In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), pp. 2431–2440. DOI: 10.1109/tvcg.2012.213 (cit. on p. 132).

[Sed16] M. Sedlmair. "Design Study Contributions Come in Different Guises: Seven Guiding Scenarios". In: *Proceedings of the Sixth Workshop on Beyond Time and Errors on Novel Evaluation Methods for Visualization*. New York, NY, USA: ACM, 2016, pp. 152–161. ISBN: 978-1-4503-4818-8. DOI: 10.1145/2993901.2993913 (cit. on p. 107).

[Sed+14] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. "Visual Parameter Space Analysis: A Conceptual Framework". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2161–2170. DOI: 10.1109/tvcg.2014.2346321 (cit. on pp. 107, 121).

[Shu+18] G. Shurkhovetskyy, N. Andrienko, G. Andrienko, and G. Fuchs. "Data Abstraction for Visualizing Large Time Series". In: *Computer Graphics Forum* 37.1 (2018), pp. 125–144. DOI: 10.1111/cgf.13237 (cit. on p. 109).

[Sti+16] H. Stitz, S. Gratzl, W. Aigner, and M. Streit. "ThermalPlot: Visualizing Multi-Attribute Time-Series Data Using a Thermal Metaphor". In: *IEEE Transactions on Visualization and Computer Graphics* 22.12 (2016), pp. 2594–2607. DOI: 10.1109/tvcg.2015.2513389 (cit. on p. 108).

[Sun+20] D. Sun, Z. Feng, Y. Chen, Y. Wang, J. Zeng, M. Yuan, T.-C. Pong, and H. Qu. "DFSeer: A Visual Analytics Approach to Facilitate Model Selection for Demand Forecasting". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2020, pp. 1–13. ISBN: 978-1-4503-6708-0. DOI: 10.1145/3313831.3376866 (cit. on p. 110).

[TLS05] A. C. Tang, J.-Y. Liu, and M. T. Sutherland. "Recovery of Correlated Neuronal Sources from EEG: The Good and Bad Ways of Using SOBI". In: *NeuroImage* 28.2 (2005), pp. 507–519. DOI: 10.1016/j.neuroimage.2005.06.062 (cit. on pp. 112, 116, 118).

[TMN16] S. Taskinen, J. Miettinen, and K. Nordhausen. “A more efficient second order blind identification method for separation of uncorrelated stationary time series”. In: *Statistics & Probability Letters* 116 (2016), pp. 21–26. DOI: `https://doi.org/10.1016/j.spl.2016.04.007` (cit. on p. 112).

[TC05] J. J. Thomas and K. A. Cook. *Illuminating the Path: An R&D Agenda for Visual Analytics*. National Visualization and Analytics Center, Department of Homeland Security, IEEE Computer Society, 2005. ISBN: 0-7695-2323-4 (cit. on p. 107).

[Tom+17] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann. “Interactive Lenses for Visualization: An Extended Survey”. In: *Computer Graphics Forum* 36.6 (2017), pp. 173–200. DOI: `10.1111/cgf.12871` (cit. on pp. 109, 121).

[Tuf01] E. R. Tufte. *The Visual Display of Quantitative Information*. Second Edition. Graphics Press, 2001. ISBN: 978-1-930824-13-3 (cit. on p. 108).

[Van+17] P. Van Thanh, D. Nhu Dinh, N. Duc Anh, N. Tien Anh, C. Duc Hoang, and T. Duc-Tan. “Automatic Removal of EOG Artifacts Using SOBI Algorithm Combined with Intelligent Source Identification Technique”. In: *Proceedings of the IEEE International Conference on Advanced Technologies for Communications*. New York, NY, USA: IEEE, 2017, pp. 260–264. DOI: `10.1109/atc.2017.8167629` (cit. on p. 106).

[VR10] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. 4. ed. New York: Springer, 2010. ISBN: 978-0-387-21706-2 (cit. on p. 121).

[VYC05] M. Vlachos, P. Yu, and V. Castelli. “On Periodicity Detection and Structural Periodic Similarity”. In: *Proceedings of the 2005 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2005, pp. 449–460. ISBN: 978-0-89871-593-4 978-1-61197-275-7. DOI: `10.1137/1.9781611972757.40` (cit. on pp. 117, 132).

[vLFR17] T. von Landesberger, D. W. Fellner, and R. A. Ruddle. “Visualization System Requirements for Data Processing Pipeline Design and Optimization”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.8 (2017), pp. 2028–2041. DOI: `10.1109/tvcg.2016.2603178` (cit. on p. 133).

[Wag+01] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. “Constrained K-Means Clustering with Background Knowledge”. In: *Proceedings of the International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann, 2001, pp. 577–584. ISBN: 978-1-55860-778-1. DOI: `10.5555/645530.655669` (cit. on p. 118).

[Wan+19] J. Wang, S. Hazarika, C. Li, and H.-W. Shen. “Visualization and Visual Analysis of Ensemble Data: A Survey”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.9 (2019), pp. 2853–2872. DOI: `10.1109/tvcg.2018.2853721` (cit. on pp. 107, 109).

[Wex+20] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viegas, and J. Wilson. “The What-If Tool: Interactive Probing of Machine Learning Models”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2020), pp. 56–65. DOI: 10.1109/tvcg.2019.2934619 (cit. on p. 110).

[WHA07] W. Willett, J. Heer, and M. Agrawala. “Scented Widgets: Improving Navigation Cues with Embedded Visualizations”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1129–1136. DOI: 10.1109/tvcg.2007.70589 (cit. on p. 119).

[Xia+22] J. Xia, Y. Zhang, J. Song, Y. Chen, Y. Wang, and S. Liu. “Revisiting Dimensionality Reduction Techniques for Visual Cluster Analysis: An Empirical Study”. In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2022), pp. 529–539. DOI: 10.1109/tvcg.2021.3114694 (cit. on p. 121).

[XT15] D. Xu and Y. Tian. “A Comprehensive Survey of Clustering Algorithms”. In: *Annals of Data Science* 2.2 (2015), pp. 165–193. DOI: 10.1007/s40745-015-0040-1 (cit. on p. 118).

[Xu+19] K. Xu, M. Xia, X. Mu, Y. Wang, and N. Cao. “EnsembleLens: Ensemble-based Visual Exploration of Anomaly Detection Algorithms with Multidimensional Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 109–119. DOI: 10.1109/tvcg.2018.2864825 (cit. on p. 109).

[XW05] R. Xu and D. WunschII. “Survey of Clustering Algorithms”. In: *IEEE Transactions on Neural Networks* 16.3 (2005), pp. 645–678. DOI: 10.1109/tnn.2005.845141 (cit. on p. 118).

[Zha+19] J. Zhang, Y. Wang, P. Molino, L. Li, and D. S. Ebert. “Manifold: A Model-Agnostic Framework for Interpretation and Diagnosis of Machine Learning Models”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 364–373. DOI: 10.1109/tvcg.2018.2864499 (cit. on p. 110).

[Zha+11] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan. “Exploratory Analysis of Time-Series with ChronoLenses”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2422–2431. DOI: 10.1109/tvcg.2011.195 (cit. on p. 109).

[Zha+14] K. Zhao, M. O. Ward, E. A. Rundensteiner, and H. N. Higgins. “LoVis: Local Pattern Visualization for Model Refinement”. In: *Computer Graphics Forum* 33.3 (2014), pp. 331–340. DOI: 10.1111/cgf.12389 (cit. on p. 110).

CHAPTER 4

Visual Parameter Optimization for Spatial Blind Source Separation

> The content of this chapter is published in:
>
> Nikolaus Piccolotto, Markus Bögl, Christoph Muehlmann, Klaus Nordhausen, Peter Filzmoser, Silvia Miksch. *Visual Parameter Selection for Spatial Blind Source Separation*. Computer Graphics Forum, vol. 41, no. 3, 2022. DOI: 10.1111/cgf14530.

Context. In the following publication, we consider SBSS, particularly a non-stationary model (Section 1.1.3), in the context of geochemical surveys and focus only on the parameter selection. Overall, the BSS task abstraction from Chapter 3 applies here too, and thus, we propose a characterization just for the parameter selection. Together with our collaborators (Christoph Muehlmann, Klaus Nordhausen, and Peter Filzmoser) and a geochemist (Clemens Reimann), we designed a VA prototype to support the identified tasks. We evaluated the prototype heuristically with visualization experts and in interviews with two SBSS experts. The best-received functionality was the automatic regionalization of multivariate spatial data, which showed accurate borders between soil types. Latent spatial dimensions identified with a parameter setting obtained by the authors showed known and surprising patterns, particularly because our dataset did not include the relevant chemical elements.

RQ's Concerned: RQ1, RQ3.

4.1 Abstract

Analysis of spatial multivariate data, i.e., measurements at irregularly-spaced locations, is a challenging topic in visualization and statistics alike. Such data are integral to many domains, e.g., indicators of valuable minerals are measured for mine prospecting. Popular analysis methods, like PCA, often by design do not account for the spatial nature of the data. Thus they, together with their spatial variants, must be employed very carefully. Clearly, it is preferable to use methods that were specifically designed for such data, like spatial blind source separation (SBSS). However, SBSS requires two tuning parameters, which are themselves complex spatial objects. Setting these parameters involves navigating two large and interdependent parameter spaces, while also taking into account prior knowledge of the physical reality represented by the data. To support analysts in this process, we developed a visual analytics prototype. We evaluated it with experts in visualization, SBSS, and geochemistry. Our evaluations show that our interactive prototype allows to define complex and realistic parameter settings efficiently, which was so far impractical. Settings identified by a non-expert led to remarkable and surprising insights for a domain expert. Therefore, this paper presents important first steps to enable the use of a promising analysis method for spatial multivariate data.

4.2 Introduction

Many domains work with multivariate quantitative measurements at different locations, i.e., multivariate spatial data. Such data can stem from, e.g., geochemical analyses of soil samples for the purpose of mine prospecting [Hal18] or investigations of environmental pollution [Rei+14]. Depending on the specific goal and application, various tasks need to be carried out on such a spatial dataset, like dimensionality reduction (DR), or finding meaningful linear combinations of involved variables [BK12; Wac03]. Spatial blind source separation (SBSS) [Nor+15; Bac+20; Mue21] is specifically designed for multivariate spatial data and reveals linear combinations of such data. It brings various advantages compared to alternative methods (Section 4.3.1), e.g., it keeps the well known loading-scores scheme from principal component analysis and properly accounts for spatial dependence due to its model-based approach. Therefore, latent dimensions identified with SBSS often correspond to the physical reality where data was collected, making it a superior analysis tool for spatial data. When irrelevant dimensions are discarded, SBSS serves as DR method as well. SBSS has been successfully applied to a geochemical dataset [Nor+15] and may be potentially used in any application domain that involves multivariate quantitative measurements at different locations.

However, a challenge to the effective use of SBSS in practice are two spatial tuning parameters that need to be set: A partition of the spatial domain into non-overlapping regions, and a configuration of non-overlapping ring-shaped kernels (Figure 4.1, see Section 4.3.1). The performance of SBSS depends largely on the choice of these tuning parameters, but the size of the parameter space is overwhelming for analysts. Theoretical guidelines about the optimal choice of tuning parameters exist (see Section 4.4.2), but they leave plenty of room for human judgement and automatic optimization does not seem feasible. Further complicating the issue, the current tool of analysts is text-based and not well suited to support them in their tasks.

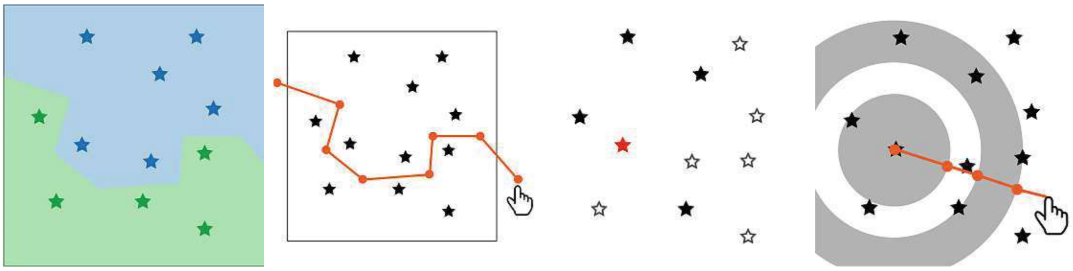

(a) (b) (c) (d)

Figure 4.1: SBSS parameters illustrated on the same locations. A regionalization (a) into a green and blue region. Two ring-shaped kernels (c) as applied to the red location. Black locations are the red location's neighbourhood. Our prototype allows setting those parameters with direct manipulation by splitting a region along a polyline (b) and choosing kernel radii (d).

As with many data analysis methods, the expertise of the human analyst is vital to SBSS parameter selection. We believe that visual analytics (VA) [TC05; Kei+08] can enable the effective use of SBSS in practice. VA pairs automatic data analysis with visual methods to combine and thereby enhance the computer's and human's individual strengths. To this end, we designed and developed interactive visualizations in collaboration with SBSS experts, who are co-authors of this paper, and an expert in geochemistry. We evaluated our approach with five visualization experts using a heuristic for value-driven visualization [Wal+19] and one domain as well as two external SBSS experts.

Our contributions are the following:

- A task description for SBSS parameter selection,
- a visualization design to support parameter selection for SBSS, including novel and existing interactions and visualizations,
- an evaluation of the design with experts in visualization, SBSS, and geochemistry.

Our contributions represent an important first step to enable the use of SBSS, a desirable multivariate spatial analysis method. Blind source separation and geostatistics in general and SBSS especially have been little explored in the visualization literature so far. In the context of visual parameter analysis our scenario is notable because of spatial tuning parameters.

4.3 Related Work

In the following we describe spatial statistics (Section 4.3.1), visualizations for geospatial point data (Section 4.3.2) and interactions with parameter spaces (Section 4.3.3) to contextualize our approach.

4.3.1 Spatial Statistics

Geostatistics is concerned with the analysis of data that show a natural order in space. Typically, many measurements at different sample locations are taken and the main source of information for proper statistical analysis of such multivariate spatial data is given by spatial dependence (cf. Tobler's law). Geostatisticians are faced with a wide variety of tasks, e.g., predicting the data at unobserved sample locations, dimensionality reduction, or finding meaningful linear combinations [BK12; Wac03]. Proper modelling of the spatial dependence is crucial for them. In the geostatistical framework it is assumed that the spatial data at hand are generated by a family of p-variate random vectors $\boldsymbol{x}(\boldsymbol{s}) = (x_1(\boldsymbol{s}), \ldots, x_p(\boldsymbol{s}))^\top$ indexed by elements $\boldsymbol{s}$ of the so-called spatial domain $S \subseteq \mathbb{R}^2$, e.g., longitude-latitude coordinates. Such a family of random vectors is referred to as multivariate random field. Spatial dependence is characterized by the so-called spatial covariance matrix which evaluates the covariance between the random field at two different sample locations. Often, the semi-variogram (covariance of the difference processes) is used in favor of the spatial covariance as it avoids the estimation of the mean but is usually harder to interpret. Modeling of proper covariance matrix functionals is a demanding task and usually simplified by further assumptions [GK15]. The second-order stationary assumption yields that the spatial covariance is invariant under shifts, i.e., the spatial covariance is the same for the whole field and only dependent on the distance between two sample locations. In contrast, the spatial covariance function of a non-stationary random field depends on specific locations and distances between locations and is therefore usually much more demanding to model.

We will outline the advantages of SBSS over principal component analysis (PCA) and its spatial variants [Jol86; Dem+13], because they are well known and widely used. For an overview of geostatistical methods see, e.g., [BK12; Wac03]. The classical PCA finds orthogonal directions of the data that maximize variance. It does so by the eigen-decomposition of the covariance matrix **Cov**, yielding the orthogonal loadings matrix **U** and uncorrelated principal components (scores) **Ux**. Two variants for spatial data are considered in the literature, both use the same methodology as classical PCA, but adapt **Cov**. The so-called geographically weighted PCA [FBC02; Har+15] uses spatial information implicitly as it computes multiple **Cov** for each sample location based on the neighbors. This leads to local PCA solutions and different loadings for each sample location, which is very time-consuming to interpret. Another variant diagonalizes the product of **Cov** and a measure of spatial dependence (Moran's I) [Jom+08], which leads to a trade-off between maximum spatial dependence and maximum marginal covariance in components. It is not clear which properties in terms of spatial/marginal dependence these components actually show. Generally, the advantage of PCA are feasible interpretations of the results in terms of the loading-scores scheme. However, PCA and both its spatial variants lack a statistical model, therefore, is not clear which spatial and/or marginal dependence properties the results actually have. SBSS, on the other hand, provides both and can find physically meaningful processes which generated the data that also have certain well-defined statistical properties.

In recent literature [Nor+15; Bac+20; Mue21] the methodology of blind source separation (BSS) [CJ10] was combined with principles of stationary/non-stationary spatial data analysis, resulting in spatial blind source separation (SBSS) for stationary and non-stationary source separation (SNSS) for non-stationary spatial data. For simplicity, both versions are referred to as SBSS in this paper. The SBSS framework is appealing as it keeps the advantageous loading-scores interpretation scheme but finds the solution by specifically accounting for spatial dependence, as it is mainly designed to find physically meaningful components. Moreover, SBSS does not restrict the loadings matrix to be orthogonal as PCA does. More meaningful components of a geochemical dataset were found in comparison to PCA by a domain expert [Nor+15], and pre-processing the data with SBSS in spatial prediction tasks simplifies the task but keeps the performance compared to other methods [MNY20]. The SBSS loadings matrix **W**—often denoted as unmixing matrix—is found by jointly diagonalizing **Cov** and a number of so-called local covariance matrices **LCov** leading to the random field $\mathbf{W}\boldsymbol{x}(\boldsymbol{s})$ (latent field) consisting of uncorrelated and spatially uncorrelated components. Local covariance matrices are computed by a weighted average of the spatial covariance matrix for all pairs of sample locations in a part of the spatial domain (regionalization, compare Figure 4.1a). The weights are determined by a kernel which only accounts for sample location pairs that are at least separated by r_{in} and at most separated by r_{out} (compare Figure 4.1c). A regionalization is needed to account for non-stationarity of the random field, while the kernels specify local proximity and attempt to measure spatial dependency. Thus, for stationary data one region is sufficient and if there is no spatial dependency present the kernels are not informative.

The crucial point which determines the performance of the SBSS methods is the choice of **LCov** matrices or more precisely choosing a set of radii parameters (kernels) and a suitable domain subdivision (regionalization). Theoretical guidelines hint that theses parameters should be chosen such that the spatial dependence of the latent field components is as different as possible. However, the practical usefulness of this statement is limited as the latent field is unknown a-priori, which opens the door for parameter selection supported by sophisticated visual analytic methods.

Spatial Data Analysis with Statistics and Visualization

After the influential work by Cleveland and McGill on graphical perception and dynamic graphics in the 1980s, researchers started to apply these ideas to spatial data. Haslett et al. [Has+91] used coordinated multiple views with interactive highlighting to find anomalies in a geochemical dataset. The linked views in question included dynamic statistical graphics, such as a variogram cloud [Cre93], histograms, and a scatterplot matrix, as well as geographic views (a map). GeoVISTA Studio [Gah+02], a visual programming environment for spatial data analysis, extended this approach and combined state-of-the-art visualizations with statistical methods for, e.g., classification. Demšar et al. [DFC08] used similar dynamic graphics but to explore spatially varying parameters of geographically weighted regression instead of the original spatial dataset. Dykes and Brunsdon [DB07] suggested adjustments to well known statistical graphics to make them work in a geographically weighted setting and for multiple spatial scales. The latter was also a focus for Goodwin et al. [Goo+16], who, more recently, used local regression coefficients to guide the analysis of a spatial dataset. To summarize, previous efforts have been put into using visualization to enable spatial exploration of the *outcome* of statistical methods. While that is future work we plan, this paper aims to enable the use of a spatial analysis method in the first place.

4.3.2 Visualization of Geospatial Point Data

As we see it necessary to visualize multivariate 2D spatial point data to facilitate SBSS parameter selection, the visualization of spatial point data is related to our work. Point data is quite common in geospatial visualizations. When the interest is in a variable's value, dot maps are often used. In those, each point is represented by one visual mark, like a circle. Other visual variables are used to encode the actual value, such as area or color. Issues may occur, e.g., when the data distribution has long tails (common in geochemistry), as a few extreme values then reduce perceptual accuracy for the majority of data points. Zhou et al. proposed the point grid map [Zho+17], in which visual marks are aligned on a grid such that directional relations are preserved. Typographic properties, like font weight, as visual channels have been explored by Brath and Bassini [BB17]. When there is little space for individual marks, pixel maps [Kei00] are an option. However, these approaches distort the location of points, which is crucial information in our case. Heatmaps and isocontours are employed when the number of points is too big for individual marks. On irregular points, like in our case these do, however, require some preprocessing as variable values need to be interpolated or resampled onto a regular grid.

There are also approaches to present point value without per-point marks on a map. Turkay et al. [Tur+14] proposed attribute signatures, in which the analyst draws a path through space and connected small multiple line charts show the value of variables along the path. Their approach scales to many variables, but only shows a small portion of variable values. Bouts et al. [Bou+16] warped the geographic space such that points with similar value are moved near each other, an idea from DR. While an interesting idea, we believe it would be unintuitive for our anticipated users.

Heatmaps are also useful to show the density of points, when individual marks tend to overlap. In this case, some abstraction is necessary. When the points are located along a road network, visual marks can be encoded along the streets with bristle maps [Kim+13]. If no natural regularization is available, it can be enforced with quadtrees [CM17], grids [GB20] or merged areas [ML19]. Finally, Phoenixmap [Zha+21] uses concave hulls for each category and encodes density along the outline. However, point densities are less of a concern for SBSS than point values.

4.3.3 Parameter Space Interactions

As we present interactive visualizations to set spatial parameters, we see interactive visualizations for other parameter spaces as related work and discuss them here. When the parameter space is multidimensional with a manageable amount of dimensions, parallel coordinate plots (PCPs) are highly popular [JF16]. They show dimensions as parallel axes and data points in the multidimensional space are encoded as polylines. Each vertex coincides with an axis where the respective value of the dimension is. Common interactions with PCPs are reordering and brushing. PCPs have also been explored as an ideation tool to quickly create new design options [MW20]. If data points exist in multiple sets, nested PCPs [Wan+17] are an option. PCPs were, in an immersive environment, also combined with scatterplots into parallel planes [Bru+16]. Each plane is a scatterplot of two variables, and polylines pass through these planes. This may help when the number of dimensions grows, but they may at some point be too many. In this case, users might still insist on sliders [Haz+20] or one could persuade them to work with a dimensionally-reduced view [Orb+19].

PCPs are great for multidimensional non-spatial data and have, therefore, been applied in combination with spatial data visualizations to enable multidimensional spatial data analysis [Gah+02; DFC08; MH18; OR18]. But different approaches are needed when the parameters have a spatial or temporal dimension. World Lines [Was+10] is an interaction paradigm to steer a flooding simulation while it happens. At different points in time, analysts may want to, e.g., place sand bags to protect an area from water, and explore the parallel universes (with and without sand bags). It preserves this branching temporal structure in the interface. In the spatial case it is popular to provide the analyst options, e.g., in the form of a spreadsheet metaphor [JM00], where possible parameter settings and their effect on the outcome are arranged next to each other. In such cases, the analyst often interacts with the parameter space through a selection in the output space, like in DreamLens [Mat+18]. A constrained editing mode that can optimize an objective interactively, e.g., the flight distance of a model airplane [Ume+14], is another useful interaction idiom. Finally, obtaining outputs by randomly precomputing large numbers of parameter settings [Sed+14] may be the simplest approach, but gets less useful and more computationally expensive the larger the parameter space is.

4.4 Background

4.4.1 Data Definition

As touched upon in Section 4.3.1, a multivariate spatial dataset in our case consists of p-dimensional vectors $\boldsymbol{x} \in \mathbb{R}^p$ at spatial locations $S \subseteq \mathbb{R}^2$. The vector at the i-th location is denoted as $\boldsymbol{x}(\boldsymbol{s}_i)$.

A parameter setting (r, k) consists of a partition, or regionalization, r and a point neighbourhood, or kernel, k. A kernel k is a set of non-overlapping rings with inner and outer radius ($0 \leq c_{in} < c_{out}$). The location of a kernel does not need to be set, as a kernel will be evaluated for every location in each region. A regionalization r partitions the spatial domain into a set of regions such that each location $\boldsymbol{s} \in S$ is contained in exactly one region. Hence, there are neither overlaps between regions nor leftover locations.

A kernel k, applied to n locations $\boldsymbol{s}_i \in S$, defines a symmetric $n \times n$ neighbourhood matrix $\mathbf{K}$. If for the distance d_{ij} between two locations $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ and any ring in k $c_{in} \leq d_{ij} \leq c_{out}$ holds, the i-th and j-th row/column of $\mathbf{K}$ contain the value 1, 0 otherwise.

4.4.2 Considerations for Selecting Parameters

There are several requirements and considerations to take into account when selecting a parameter setting (r, k) for SBSS.

Technical Requirements. From assumptions in SBSS theory follows, as already touched upon in Section 4.4.1, that the regions in r must not overlap. To further simplify finding regions, we require that their union must contain all locations in S. These are easy to enforce automatically, but two other considerations require the human analyst.

Balance Region and Kernel Size. A guideline by our collaborators to reduce the estimation error in the weighted average (Section 4.3.1) is that each region in r should contain a reasonably large amount of locations. The same is true for a kernel k, which should capture reasonably many locations in each region. Hence, r and k are not chosen independently. If a region contains sparsely distributed locations, the kernel needs to be bigger than for a denser region to capture the same number of locations. It clearly is also not useful if, e.g., the inner radius of the kernel is bigger than a region, as no locations would be captured. In practice, analysts should first select regions and kernels based on the guidelines below and afterwards verify that no region/kernel is "too small," based on a threshold that is appropriate for the dataset and application. In our evaluations (Section 4.7.1), participants initially set this threshold to 5% of data points. If too small regions/kernels are identified, analysts may proceed regardless or merge/expand regions/kernels, again following guidelines below.

Reconcile With Domain Knowledge. Another recommendation from SBSS theory is that regions should be selected such that they enclose areas where variables behave, or can be expected to behave, very differently from the other regions. This, however, depends on the concrete dataset SBSS is applied to, and prior knowledge about the physical reality it represents. As an example, if the measured variables are about air quality, it may make sense to distinguish between urban and rural regions in the data, but in case the measured variables are elements in soil, different soil types could guide the regionalization. Similarly, a kernel should be selected such that it encapsulates the spatial dependence of different latent processes in the dataset, i.e., a kernel should cover the distance within which a process may be noticeable. Such a latent process might be, e.g., emissions from driving cars, which influence air quality up to a distance of a few hundred meters [LCX19]. In the same way as a regionalization, the kernel parameter also clearly requires the domain knowledge of the analyst. Such considerations are difficult to quantify, but may be supported by others that are easier to (data-driven considerations). For instance, which latent processes can be expected in the dataset depends on which variables were measured and how far apart. The spatial dependence of a variable, important for kernel selection, can be expressed with a variogram [Cre93]. It is possible to automatically partition a spatial domain [Guo08], which could be an initial suggestion for this complex parameter.

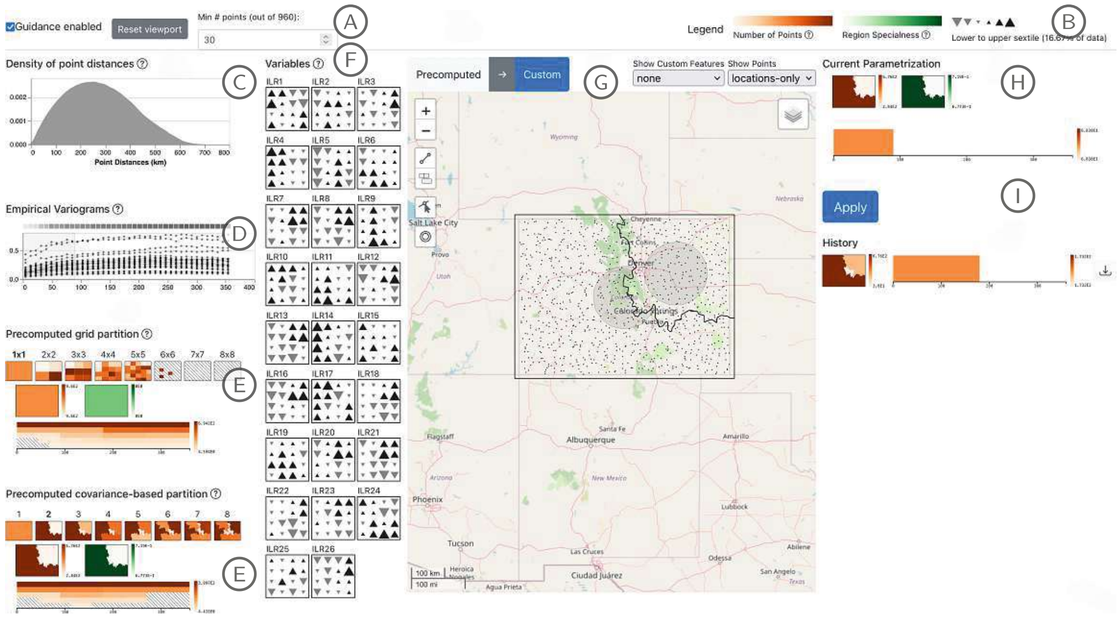
To summarize, SBSS parameter selection is characterized by a small set of rules that can be easily verified automatically, and a larger fuzzier set of guidelines that require human reasoning and domain knowledge. How our visual analytics prototype supports both is the topic of Section 4.6.

4.5 Task Description

We describe users and their tasks using the design triangle by Miksch and Aigner [MA14]. The data is described in Section 4.4.1.

Users. As SBSS is a relatively novel statistical method, our users are for now SBSS experts who want to investigate their method on real data instead of the usual simulation studies. While SBSS experts have formal education in mathematics/statistics and are knowledgeable in spatial statistics, we paid attention that this is not a requirement for our visual designs. We anticipate that as interest in SBSS grows in the future, domain experts without such qualifications will require our interactive visualizations, too. Our users' main tool is RStudio, an integrated development environment (IDE) for R [R C23], a language for statistical computing. RStudio is text-based and allocates one part of its interface to show a non-interactive visualization (which has to be programmed by the user with, e.g., ggplot2 [Wic16]).

Tasks. User tasks emerge from parameter setting considerations described earlier (Section 4.4.2). First and foremost, users need to be able to *quickly and efficiently enter parameter settings* (T1), also complex ones. As can easily be imagined, this is not possible with a text interface. For this reason, users currently favor parameter settings that can be easily described with code, such as regionalizations that are grids or regular slices in a particular direction, although these may not correspond to the spatial reality in the data. Furthermore, they have to *balance region and kernel size* (T2) and *reconcile possible regions and kernels with their domain knowledge* (T3). The former is currently difficult as regions and kernels are selected without a direct manipulation paradigm, and the latter because only a single visualization is visible at a time.



Min # points (out of 960): 30

Show Custom Features: none

Show Points: locations-only

Figure 4.2: Screenshot of our prototype (*Colorado* dataset). It shows the toolbar including legends (B) and a filter for number of locations (A), visualizations supporting data-driven considerations (C and D, Section 4.6.4), precomputed regionalizations and kernels (E, Section 4.6.2), variables as small multiples (F, Section 4.6.3), an interactive map (G, Section 4.6.1), the analyst's current selection (H) and past selections (I).

We obtained the necessary tasks in a user-centered design process with experts in SBSS and geostatistics. We also asked an expert in geochemistry for feedback on our visualizations during the design phase. He underlined the importance of task T1, that the system should be highly interactive and allow to produce many parameter settings in little time.

4.6 Visualizations and Interactions

In this section we describe the interactive visualizations of our prototype (Figure 4.2) and relate them to the task description (Section 4.5). We implemented those as part of a client-server architecture with the client being a JavaScript application and the server written in R. The latter is mainly to use the `SpatialBSS` R package [MNV21] that provides necessary functions. Both client and server carry out time-intensive computations once and re-use results, thereby allowing fluid interactions. The software is available online [Pic21]. We describe and show the design with changes made *after* our evaluations. We resized some elements and removed the guidance previously encoded in the blue colorscale (cf. Section 4.6.2).

4.6.1 Interactive Map (T1, T2, T3)

The SBSS parameters—regions and kernels—are complex spatial objects. It is time-intensive, error-prone, and frustrating to define these in an indirect manner by textual commands. A direct manipulation interface for both of them seemed therefore promising for task T1. We achieve this in an interactive map, which supports the usual pan and zoom interactions. Not only can the analyst define regions and kernels directly in their spatial context (tasks T2, T3), with an interactive map it is also possible for us to show supporting data to guide the parameter selection (task T3).

Notably the map has two modes. One is the "precomputed" mode, which allows to view precomputed guidance suggestions (Section 4.6.2). If the analyst wants to build their own parameter setting or modify a precomputed one, they need to switch to the "custom" mode. In this mode they can split a region in two, merge two adjacent regions, and define a kernel directly in the map (Section 4.6.1). Any precomputed setting can be copied to the "custom" mode for modification.

Visualization of Regions and Kernels (T2)

As per common convention in cartography, we show regions as polygons. They are not filled to not occlude the underlying map tiles, which provide important information.

We show the current kernel configuration as shaded concentric circles at the geometric center of each region. This is for two reasons: First, a single kernel configuration is used for all regions, which was an acceptable simplification to our collaborators. Hence we may copy it as soon as a new region is defined. Second, there is no single center for a kernel, as it will be evaluated at all locations. Shown at a region's center we can expect that analysts will be able to reason well about a kernel and region's relative size (task T2). We use the continuity Gestalt principle to encode which region a kernel belongs to and crop the rings by the region's boundary.

Direct Manipulation of Regions and Kernels (T1)

At first, the interactive map just shows the bounding box of all locations, and neither locations nor variables, to not clutter it from the start. This is important because we do not know in advance how many locations the dataset contains.

In discussions with our collaborators we learned that they expect regions to be coarse and few. This is partly because a region must not be too small (task T2, Section 4.4.2). As we further required all locations to be assigned to a region, a *regionalization* is shaped by splitting an existing region in two along a user-defined border, which is provided by drawing a polyline through a region on the map (Figure 4.1b). To merge adjacent regions it is sufficient to select them. This allows to quickly define also complex regions, while maintaining correctness (task T1).

A kernel configuration is a set of concentric rings. They are defined as follows. First, the analyst picks a center point anywhere on the map. From there, the analyst has to choose alternatingly the outer and inner radius of a ring (Figure 4.1d). The process is terminated and kernel definition complete when the kernel selection mode is turned off. With this direct manipulation approach and supporting views it is easily noticeable when there are overlapping kernels. Hence, we support kernel definitions in a quick and correct way (task T1). Please refer to the video in the supplemental material (linked in the appendix) for all visual feedback we provide.

Additional Data (T3)

Several additional spatial objects may be shown on the map to support selection of the parameters (task T3).

Custom Annotations. Our approach offers the ability for custom map annotations. The analyst may provide and overlay any GeoJSON feature collection [But+16]. This way, their domain knowledge can be externalized and visually encoded to support parameter selection. In a geochemical setting, it could be a soil atlas [HMZ09]. The cursor then snaps to the boundary of features, further simplifying the process.

Locations and Variables. Analysts may choose to show just the locations in the map encoded as points. This is a compromise between no locations and showing a spatial variable. They may, however, also overlay any single spatial variable of the dataset instead. These are encoded in the same way as in the small multiples explained in Section 4.6.3: A colored triangle of differing size shows the sextile (1/6 or 16.67% of the data).

Base Layer. Finally, we provide several base layers of the map to choose from. The default is OpenStreetMap and OpenTopo, Thunderforest Landscape [Gra] and Satellite are also available. We expect that these cover most commonly needed information as they provide layers optimized for both rural/natural and urban areas.

4.6.2 Guidance (T1, T2, T3)

The set of possible regionalizations and kernel definitions is vast and it is difficult for analysts that lack deep domain knowledge, to find a starting point. They do not know what possible parameter settings look like and how they compare. Hence, they need guidance [Cen+17]. We provide *orienting* and *directing* guidance in the following way (Figure 4.2, E). Possible regionalizations are precomputed, using a current strategy of analysts (grid-based) and one that matches SBSS experts' recommendations (covariance-based). Similarly, possible kernel settings are precomputed. We show these as suggestions (*directing*, Section 4.6.2) and color-code them by quantification measures (*orienting*, Section 4.6.2).

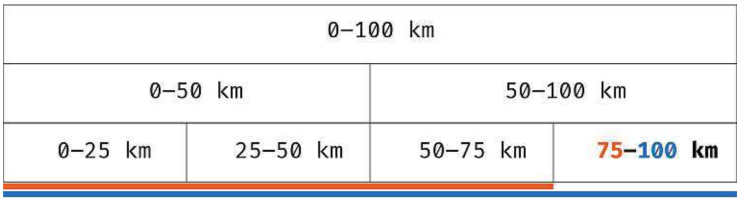


Figure 4.3: Structure of the visualization for kernels (Figure 4.2, E). Each kernel is a rectangle representing one part of a largest kernel.

Finding Regions and Kernels

A regionalization is visualized as choropleth map, with univariate color scales as defined in Section 4.6.2. We use two strategies to provide suggestions for regionalizations.

Grid-based Regionalization. For lack of better tooling, grids are currently a popular setting for the regionalization parameter. These can be quickly precomputed in a straightforward manner. We use square $n \times n$ grids with n from 1 to a user-defined granularity.

Covariance-based Regionalization. Recall that in Section 4.4.2 we described that regions should be selected such that the variable interactions are different. When we consider the covariance of variables as a measure, we can compute suggestions for a regionalization automatically. We first convert the point dataset to a polygon dataset using a Voronoi diagram. Then we group adjacent similar Voronoi cells using the REDCAP regionalization algorithm [Guo08]. In REDCAP's terms, we use $dist_{edge}$ as edge length and hg_r as region heterogenity:

$$\begin{aligned} dist_{edge}(\boldsymbol{s}_i, \boldsymbol{s}_j) &= \|\boldsymbol{x}(\boldsymbol{s}_i)\boldsymbol{x}(\boldsymbol{s}_i)^T - \boldsymbol{x}(\boldsymbol{s}_j)\boldsymbol{x}(\boldsymbol{s}_j)^T\|_F \\ hg_r &= \sum \|(\boldsymbol{x}(\boldsymbol{s}_i) - \overline{\boldsymbol{x}}_r)(\boldsymbol{x}(\boldsymbol{s}_i) - \overline{\boldsymbol{x}}_r)^T - \mathbf{Cov}_r\|_F \end{aligned}$$

i, j are indices of locations, $\mathbf{Cov}_r$ is the sample covariance matrix of all locations in the region and $\overline{\boldsymbol{x}}_r$ the means of variables in the region. $\| \cdot \|_F$ denotes the Frobenius norm. With these hyperparameters for REDCAP, we gain regionalizations for a user-defined number of regions. This approach was very successful in our evaluation (Section 4.7).

Kernels. For kernel suggestions we consider only kernels with a single ring, as the rings have no influence on each other. We obtain smaller rings by a recursive binary partition of a largest ring. To visualize the precomputed rings, we show them as stacked bars (Figure 4.3), where the Y axis encodes ring thickness and the X axis distance. The left edge of a bar marks the inner radius of a ring, the right edge the outer radius. The bars are colored according to a color scale described in Section 4.6.2. Single rings can be selected to be viewed on the map, but any combination of rings may be defined manually.

Quantifying Regions and Kernels

Number of Locations. For analysts it is important to know how many locations are contained in a region and captured by a kernel (Section 4.4.2). Hence, for any region in a regionalization, we simply count the number of locations in it. For a kernel in a region, we compute the neighbourhood matrix $\mathbf{K}$ (Section 4.4.1) and define the number of locations captured by the kernel as the mean of column sums in $\mathbf{K}$. As one column in $\mathbf{K}$ contains the neighbourhood for a single location defined by a kernel, it is the average neighbourhood size. This metric is encoded in the orange color scale. It may be used to inform, e.g., if a region may be split further or if two adjacent regions should be merged (tasks T2, T3).

Insufficient Number of Locations. A pattern of diagonal stripes appears when the number of locations in a region or kernel neighbourhood is smaller than a custom threshold. This way, analysts can easily detect too small regions or kernels (task T2).

Region Covariance Difference. In Section 4.4.2 we outlined that regions should be selected such that the variable interactions are different. One way to describe those are by the sample covariance matrix $\mathbf{Cov}_r$ of all $\boldsymbol{x}(\boldsymbol{s}_i)$ in a region. The difference of each $\mathbf{Cov}_r$ to the global sample covariance matrix $\mathbf{Cov}$ can then the quantified by the Frobenius norm: $\|\mathbf{Cov} - \mathbf{Cov}_r\|_F$. Higher values indicate more locally different variable interactions. This metric is encoded in the green color scale. This should be used to identify as many as much locally different regions as possible, as long as they are also reasonable for a domain expert (task T3).

Eigenvalue Difference. SBSS theory states that high quality recovering of the latent field is achieved if the eigenvalues of the local covariance matrices (Section 4.3.2) evaluated on the *latent* field are as different as possible [Bac+20; Mue21]. Hence, a promising parameter setting maximizes the difference between these eigenvalues. Unfortunately, the latent field is unknown beforehand. However, in this spirit, our collaborators suggested that the eigenvalue difference of the local covariance matrices evaluated on the *input data* might be a useful metric to suggest the latent field recovery quality of a parameter setting. In the version of the prototype we used for our evaluations (Section 4.7), this metric was encoded in a blue color scale. As it was not well accepted among study participants, probably due to its unreliability, we removed it from the final design presented in this paper (Figure 4.2).

4.6.3 Summary of One Spatial Variable (T1)

We decided to show all involved variables separately to the interactive map as small multiples (Figure 4.2, F). Following parameter selection considerations in Section 4.4.2, analysts need to identify areas of the spatial domain in which many variables have consistent values, which makes it necessary to show all variables at once. This is effectively a manual regionalization (Section 4.6.2) and the small multiples simultaneously provide an overview of all variables.

A single spatial variable is summarized by aggregating it to a grid. The size of the grid can be interactively changed by the analyst (semantic zoom). For each grid cell, the median value of the variable is encoded by a triangle symbol showing the percentile it falls in (Figure 4.2, B). This design was preferred by our collaborators over a heatmap or isocontours. We divide the data in sextiles (1/6 or 16.67% of the data). The lower three sextiles are gray and upside-down triangles, the upper three are black and upright triangles. This double encoding is redundant, but allows to perceive contiguous regions due to the shared color and also intuitively indicates which percentiles are shown: Downward-pointing triangles show lower values, upward-pointing triangles higher values. As for our collaborators the extreme values are of interest, values away from the 3rd and 4th sextiles are shown with bigger triangles. The relative sizes were chosen based on [Den96].

4.6.4 Distance Distribution and Variograms (T3)

We show two plots to support the data-driven selection considerations (Section 4.4.2).

Distance Distribution. To know how far away locations are from each other we show a density plot of all pairwise distances (Figure 4.2, C). From that an analyst can easily see if the spatial scale of the dataset is on hundreds of meters or thousands of kilometers. This is in addition to the interactive map (Section 4.6.1).

Variograms. The empirical variogram is an established plot in spatial statistics [Cre93, Chapter 2] that shows the spatial dependence of a variable, i.e., how its value changes with increased distance. With the (binned) distance on the X axis, the Y axis encodes the average squared difference between any point pair whose distance falls in that particular bin. We combine variograms of all variables in the dataset by superpositioning them (Figure 4.2, D). In Section 4.4.2 we explained that kernels can be selected such that they encapsulate dissimilar spatial behavior of variables. To support this assessment, we add a grayscale-coded square on top of each bin that encodes the variance. Hence, darker squares point to bins with more dissimilar spatial behavior. When the analyst selects a kernel, its current extent is interactively shown in the variogram view.

4.7 Evaluation

In previous sections we described users and their tasks (Section 4.5) and presented our interactive visualizations (Section 4.6). In this section we describe our efforts to evaluate these visualizations. We were interested in the following research questions:

- (RQ1) Do our interactive visualizations enable more *efficient* parameter selection? I.e., can analysts enter complex settings in less time?

- (RQ2) Do our interactive visualizations enable more *effective* parameter selection? I.e., do they allow analysts to enter previously-impractical settings?
- (RQ3) Is our designed guidance effective, i.e., is it semantically meaningful and accepted by users?

Evaluations were carried out with three groups of participants. We presented our prototype to five visualization experts (Section 4.7.2), who judged its value using a questionnaire [Wal+19]. This is to confirm that we did not make gross mistakes in the visualization design phase. After that, we invited two external SBSS experts (Section 4.7.1), who did not take part in the design phase, to a user study. Here we were interested in how our prototype can improve their parameter selection process. Finally, we showed the covariance-based regionalization guidance (Section 4.6.2) and latent dimensions (output from a parameter setting made by the second author) to an expert in geochemistry (Section 4.7.3), to judge how meaningful suggested partitions and acquired results are.

Hence, we combined quantitative and qualitative approaches. However, we did not deem it useful to compare RStudio and our visualizations in a quantitative way involving time and error. The two are based on completely different interaction paradigms and provide wildly differing levels of support to the analyst. From the discussion with SE2 we think we were right in that decision.

Datasets Used in Evaluation. For visualization experts we exclusively used the *GEMAS* [Rei+14] dataset (2 108 locations / 18 variables), because it covers most of Europe and we expected it therefore to be somewhat relatable. SBSS experts preferred the *Kola moss* [Rei+98] (594 / 31) and *Colorado* [SEK10] dataset (960 / 27). For guidance judgement we again used the *GEMAS* and *Kola moss* datasets, because the domain expert is one author of them and intimately familiar. All datasets are publicly available.

4.7.1 SBSS Experts

Regarding our research questions of efficiency and effectivity, we interviewed two people who work a lot in RStudio and are SBSS experts. We introduced them (SE1 and SE2), who have at least one publication on SBSS, to our visualizations. They used the prototype on a dataset they chose. These were different datasets. We asked them to produce a few parameter settings using our prototype. We did not provide any requirements to this task, to not constrain their exploration and ideas. Of course, we helped them if they did not remember visual encodings or interactions. We asked them to vocalize their plans and intentions ("think aloud"). After they were done or the time ran out, we discussed the visualizations and interactions in an unstructured fashion. The sessions took around 75 minutes each. SE2 even provided us beforehand parameter settings they made in RStudio.

In the beginning, SE1 had difficulties using the prototype. Especially the distinction between the "precomputed" view-only map mode and the "custom" editable mode was confusing, as both looked similar but edit controls were missing in one of them. However, after 15–20 minutes, SE1's interactions became quite fluid. SE2 had no problems from the start. This suggests that there is little training time necessary to use our prototype.

Both experts, being knowledgeable about SBSS but not the application domain of geochemistry, relied in their selection process heavily on the guidance of our prototype.

SE1 browsed through many suggestions, but had trouble to commit to any particular setting. It is possible that we provided too many options, or at least should not have shown them all at once. For lack of better judgement, SE1 settled for four parameter settings from our guidance system, with minor modifications.

SE2 went about it in a more structured way, but produced only a single setting in the end. At first SE2 also mostly browsed through (grid-based) suggestions and inspected them in the interactive map. SE2 also paid attention to the colormaps of the suggestions, although more on the orange and green one. At some point SE2 decided to find the most locally different region and clicked through grid-based regionalizations. These tended to show regions in the center as darkest, to confirm this SE2 inspected the variable summaries. There SE2 noticed that many variables had consistently similar values in the top right square and in the left-most column of the map. We pointed out that this observation matches the covariance-based regionalizations, and SE2 used this guidance more from that point on. SE2 combined the covariance-based maps and the variable summaries to decide for one regionalization, specifically the most fine-grained one that did not split the regions of interest identified earlier. Then a process of fine-tuning began, where SE2 split and merged regions to distribute locations evenly, while keeping as much of the baseline regionalization intact as possible. Finally, kernels were selected with support of the variograms. In the end, SE2 selected a parameter setting with much more complex regions than in prior attempts made using RStudio. Judging from our conversations with the domain expert (Section 4.7.3), this setting is likely more realistic, too.

While we had plenty of time with SE2, it was not the case with SE1, with whom we were not able to discuss drawbacks and benefits deeply. SE1 raised no improvement suggestions for the prototype, but mentioned that the purpose of the Eigenvalue guidance in the blue colormap is not clear. This was reinforced by SE2, who also did not look at it that much. We believe that this is because it relates only to the output and its impact on that is unclear. Therefore, this guidance is unreliable and we removed it in the final prototype. Regarding using our prototype vs. RStudio, SE1 mentioned that "the precomputations are extremely useful." This was also echoed by SE2, but suggested that it would have been nice if similar suggestions existed for the kernel parameter, too. SE2 also noted that being able to see the original variables would have been useful. This is related to a technical detail with geochemical data: Since these are measured as part of a whole (e.g., mg/kg in a soil sample), it is necessary to apply some data transformations first [Ait82] and our prototype showed variables only after these transformations. We

Component	Mean	Std.dev.
Insight	6.35	0.98
Time	6.52	0.65
Essence	6.05	0.89
Confidence	6.00	2.98

Table 4.1: Results of the ICE-T evaluation with visualization experts.

suggested SE2 to recreate one of their existing parameter settings with our prototype. However, SE2 declined with an interesting answer: It would “probably be faster” but pointless, as they would “very likely not be interested in choosing the same settings,” given the fewer constraints and additional supporting views of our prototype. We take this as strong evidence for our initial assumption, that providing tailored interactive visualizations change the selection process, and for our first two research questions. As both experts made use of our guidance suggestions, we see this as supporting evidence also for our third research question, that our guidance is effective.

4.7.2 Visualization Experts

We asked visualization experts to judge our visualization design. While good design does not automatically entail a more efficient/effective selection process, bad design most likely prohibits it. We used the heuristic value of visualization approach by Wall et al. [Wal+19] (ICE-T), because it is a good compromise between insight gained for us and time required for participants. We introduced five visualization experts from two universities, who are Ph.D. students or postdocs in visualization, to the SBSS problem domain and our prototype (Section 4.6). Five experts are sufficient according to the power analysis by Wall et al. The experts were allowed to use the prototype on their own and ask as many questions as necessary, until they felt confident enough to fill out the questionnaire. We discussed the terminology beforehand. The sessions took around one hour each and were conducted solely by the first author. The results are depicted in Table 5.1.

It can be seen that our approach was rated very well across ICE-T components. For our purpose, we see Time and Insight, in that order, as the most important components, which also were rated highest on average. Wall et al. [Wal+19] state that a visualization design can be considered successful if the mean score is greater than five, which we clearly achieved. We provide the raw questionnaire results as supplementary material.

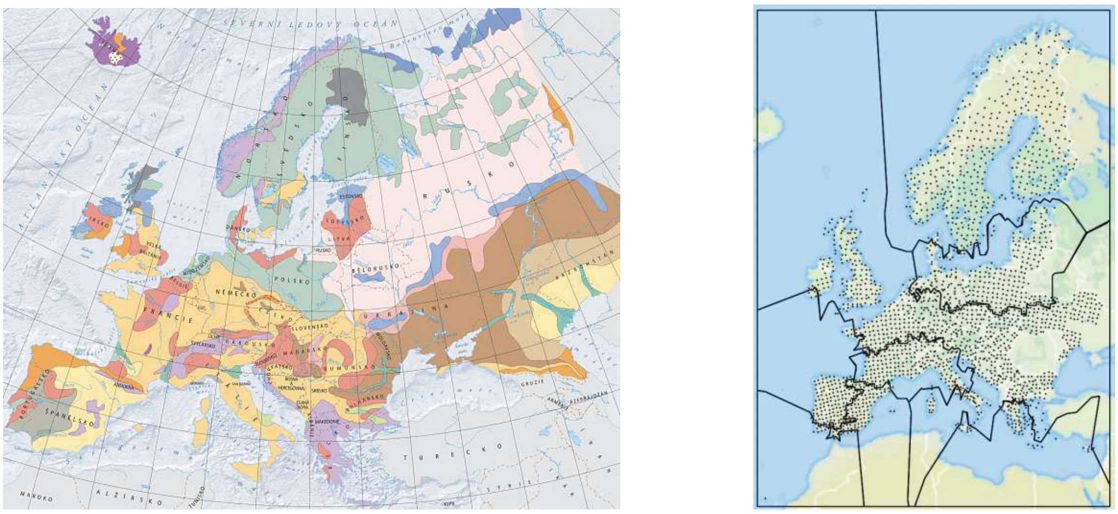
However, the standard deviation in the Confidence component is very high. The reason for this is that two out of four statements were often either deemed not applicable or rated badly by our participants. These pertain to facilitating learning more broadly about the data domain and helping to understand data quality. The former is not important to our prototype, as it is designed to support parameter selection only, and not general data analysis capabilities. The latter partly is: SBSS requires complete data, therefore good data quality can be seen as a precondition. On the other hand, duplicate or outlier entries could still exist, but would be invisible due to occlusion and the percentile summary. This could be solved, e.g., by a layout that avoids occlusion, by annotations when occlusion happens or by highlighting symbols if their value is greater than a user-defined number of standard deviations.

Several points were raised in the open discussion. The variogram was unknown to all participants, but thought to be a good way to show spatial dependence of variables. That there should be more space for the map, also because the other views could be shown conditionally, was raised by two participants. Our triangle symbols were deemed both intuitive and not ideal because it is not easy to see where the center is. Participants also suggested to make it possible to analyze custom groups of variables.

4.7.3 Domain Expert

We showed the automatic covariance-based regionalizations for the *GEMAS* dataset (Figure 4.4b) to the geochemistry expert. He immediately recognized geologically and, therefore, geochemically distinct areas that are characterized by their soil (Figure 4.4a), such as eastern Spain (Calcisol), Central Europe (Cambisol), Southern Baltic region (Albeluvisol), or the Nordic countries (Podzol). He further mentioned that such an automatic regionalization based on multivariate data would likely be helpful for geologists and geographers as an initial estimation of homogeneous regions. This is often necessary, e.g., because non-spatial methods, like PCA, must not be used on inhomogeneous data [Rei+08, Chapter 14]. The domain expert was not able to reconcile the automatic regionalizations with known processes in the *Kola moss* dataset. In our opinion, even this negative assessment is useful, as it suggests a stationary SBSS setting (i.e., no regionalization required). Overall, we take this as evidence that our regionalization suggestions can reflect real processes and be a starting point even for domain experts.

To further test the applicability of our interactive visualizations, the second author used them to define a parameter setting on the *GEMAS* dataset. This is difficult because of the dataset's complexity (2 108 locations covering Europe, 18 variables), especially for someone unfamiliar with the application domain. Yet it took him only a few minutes. We then plotted static maps of the resulting latent components and showed them to the geochemistry expert, who noticed familiar, surprising and unknown patterns. Unexpected was a structure in the area of North France, Belgium and Germany (Figure 4.5a). The expert speculated it is caused by sediments, but then the pattern would extend east- instead of westwards. While there are possible explanations, like population density, more research is necessary to confirm them. More unexpected patterns were identified near known sites of mining activity in Seville (Rio Tinto) and Almadén (Figure 4.5b). This was insofar surprising to the expert as Almadén is a mercury deposit and Rio Tinto copper/zinc, yet neither mercury nor copper were part of the dataset we used. The expert generally was impressed that a lot of known processes, like historic geological events (e.g., Oslo rift, glacial period), were so well visible, even though our dataset did not include the "most interesting elements." Revealing the same interesting patterns with fewer variables



(a) Main soil types in Europe. Reproduced from [HMZ09].

(b) 8 automatic regions for the *GEMAS* dataset.

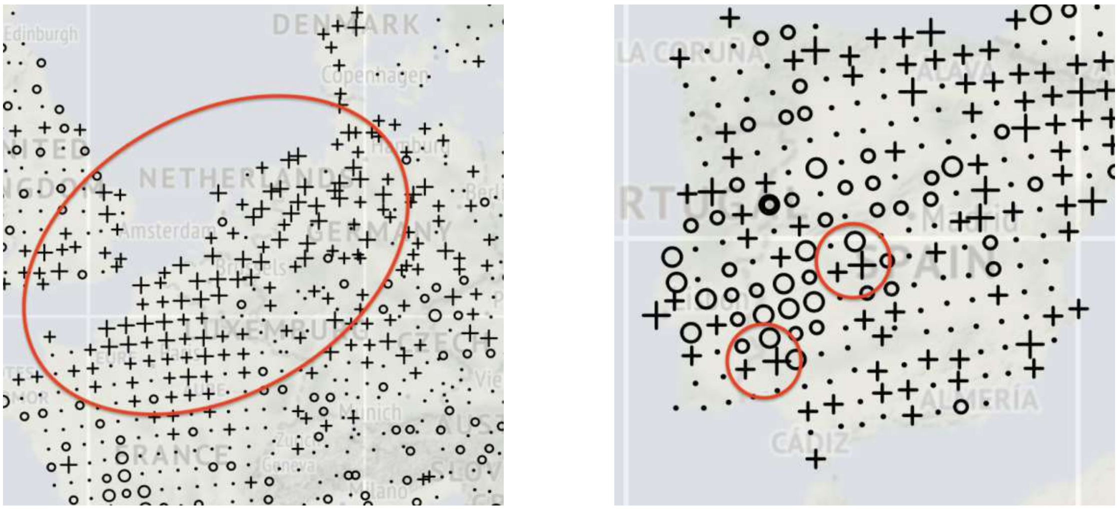
Figure 4.4: Comparison of a) a map of soil types in Europe and b) our regionalization guidance. While not perfect, as it is limited here to 8 regions and the dataset captures many more latent processes than just soil type, our guidance suggests similar boundaries, such as East/West Spain or North/South Baltic region.

has monetary implications in geochemistry, as some elements are expensive to measure within useful detection limits (and could be excluded). These insights show how useful SBSS can be for multivariate analysis of spatial data, and how accessible they became to novice users with our interactive visualizations.

4.7.4 Limitations and Discussion

Being a research prototype, our VA approach does have its limitations (see also Section 4.7.1 and Section 4.7.2). The computational demand increases with the number of locations in the dataset ($\mathcal{O}(n^2)$ per regionalization with REDCAP), hence the precomputations may take several minutes. Further, any region currently must be a single contiguous area without holes.

Our research questions pertained to the efficiency (RQ1) and effectivity (RQ2) of parameter settings with our interactive visualizations and the effectiveness of the guidance we designed (RQ3). To answer RQ1 and RQ2 we performed, on the one hand, a heuristic evaluation with visualization experts. Our prototype scored particularly well in the Time component, as participants strongly agreed that it provides efficient interactions. Visualizations were also deemed appropriate, except to find data quality issues. The latter is a minor issue as, in practice, SBSS expects a properly preprocessed dataset. On the other hand, we introduced our prototype to two external SBSS experts, who used it to select parameters on a dataset of their choice. Little training time was necessary



(a) Unidentified process causes pattern in France and Germany.

(b) Patterns near mercury (top circle) and copper mines.

Figure 4.5: Insights into the *GEMAS* dataset with SBSS. Images show high (crosses) and low (circles) values of a latent dimension [Rei+08]. Zoom, crop and red annotation by the authors.

and the visualizations and guidance suggestions were considered useful. One expert stressed how our prototype allows to set previously practically impossible parameter settings. Therefore, we think RQ1 and RQ2 can be answered positively. Our third research question (RQ3) was about the effectiveness of our guidance. The availability of regionalization suggestions was considered very useful by SBSS experts. A novice in geochemistry (the second author) was quickly able to select parameter settings that lead to surprising insights for a domain expert. We therefore think that also this research question can be answered positively.

4.8 Conclusion

SBSS is a desirable tool for multivariate spatial data analysis. It requires setting complex spatial tuning parameters: a partition of the spatial domain (regionalization) and a spatial neighbourhood configuration (kernels). In this paper, we presented a visual-interactive prototype that supports and guides analysts in finding appropate settings, thereby rendering it more usable in practice. We developed it in close collaboration with experts in SBSS, geostatistics, and geochemistry. The prototype contains several interactive capabilities to modify parameters and guiding visualizations. We evaluated the prototype quantitatively using a heuristic evaluation with five visualization experts and qualitatively with two external SBSS experts, who were not part of the design process, and a geochemistry expert. Our evaluations show that

- our visualizations are appropriate and the prototype allows highly interactive exploration of possible parameter settings,
- our prototype allows SBSS and visualization experts to select parameters more flexibly, efficiently, and realistically,
- our guidance suggestions can be semantically meaningful to a domain expert and are considered helpful by SBSS experts.

During this study we discovered partial results that we think can be transferred between the domains of geostatistics, geochemstriy, and visual analytics for mutual benefit. For instance, the ideas of variograms and regionalizations rarely occur in visual analytics literature for spatial data. Flexible and interactive variograms [Has+91] and regionalizations are useful for exploratory analysis of spatial data, which in turn can support geostatistical modeling. It would be interesting to further explore, how these can be combined with state-of-the-art interactive visualizations. Similarly, automatic regionalizations may be useful for geochemists, as they suggest homogeneous areas from multivariate data, which are interesting by themselves and can be analyzed by other methods, such as PCA. To further improve upon the concept of regionalizations it would be beneficial to make them uncertainty-aware, as we learned that region boundaries in practice may not be crisp and clear-cut when multiple influencing processes overlap.

Our contributions present a first step towards the effective practical use of SBSS. In the future we could look into more quantitative approaches at several stages of the design study, e.g., comparing SBSS results obtained with our prototype to a ground truth, investigating an objective-oriented parameter selection approach, or conduct experiments to find out which visualizations are best for the tasks we identified. Further topics arise as a result from our focus on visual-interactive parameter *selection*. Because several parameter settings are tried in practice, it raises the question how common visual parameter analysis tasks, like sensitivity analysis, can be possible with spatial parameters. Finally, it would be beneficial if the exploration of multiple SBSS results are supported by interactive visualizations.

Acknowledgements

This work was funded by the Austrian Science Fund (FWF) under grant P31881-N32. We sincerely thank Clemens Reimann for his support and advice. We also thank Helwig Hauser and our anonymous study participants for their time and valuable discussions.

4.9 Bibliography

[Ait82] J. Aitchison. “The Statistical Analysis of Compositional Data”. In: *Journal of the Royal Statistical Society* 44.2 (1982), pp. 139–177 (cit. on p. 159).

[Bac+20] F. Bachoc, M. G. Genton, K. Nordhausen, A. Ruiz-Gazen, and J. Virta. "Spatial Blind Source Separation". In: *Biometrika* 107.3 (2020), pp. 627–646. DOI: 10.1093/biomet/asz079 (cit. on pp. 144, 147, 156).

[BK12] T. Bailey and W. Krzanowski. "An Overview of Approaches to the Analysis and Modelling of Multivariate Geostatistical Data". In: *Mathematical Geosciences* 44 (2012), pp. 381–393. DOI: 10.1007/s11004-011-9360-7 (cit. on pp. 144, 146).

[Bou+16] Q. W. Bouts, T. Dwyer, J. Dykes, B. Speckmann, S. Goodwin, N. H. Riche, S. Carpendale, and A. Liebman. "Visual Encoding of Dissimilarity Data via Topology-Preserving Map Deformation". In: *IEEE Transactions on Visualization and Computer Graphics* 22.9 (2016), pp. 2200–2213. DOI: 10.1109/tvcg.2015.2500225 (cit. on p. 148).

[BB17] R. Brath and E. Banissi. "Multivariate Label-Based Thematic Maps". In: *International Journal of Cartography* 3.1 (2017), pp. 45–60. DOI: 10.1080/23729333.2017.1301346 (cit. on p. 148).

[Bru+16] N. Brunhart-Lupo, B. W. Bush, K. Gruchalla, and S. Smith. "Simulation Exploration through Immersive Parallel Planes". In: *2016 Workshop on Immersive Analytics (IA)*. 2016, pp. 19–24. DOI: 10.1109/immersive.2016.7932377 (cit. on p. 149).

[But+16] H. Butler, M. Daly, A. Doyle, S. Gillies, S. Hagen, and T. Schaub. *The GeoJSON Format*. Rfc7946. RFC Editor, 2016, Rfc7946. DOI: 10.17487/rfc7946 (cit. on p. 154).

[Cen+17] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H. Schulz, M. Streit, and C. Tominski. "Characterizing Guidance in Visual Analytics". In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 111–120. DOI: 10.1109/tvcg.2016.2598468 (cit. on p. 154).

[CM17] A. Chua and A. V. Moere. "BinSq: Visualizing Geographic Dot Density Patterns with Gridded Maps". In: *Cartography and Geographic Information Science* 44.5 (2017), pp. 390–409. DOI: 10.1080/15230406.2016.1174623 (cit. on p. 148).

[CJ10] P. Comon and C. Jutten. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Oxford: Academic Press, 2010. DOI: 10.1016/c2009-0-19334-0 (cit. on p. 147).

[Cre93] N. A. C. Cressie. *Statistics for Spatial Data: Revised Edition*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 1993. ISBN: 978-1-119-11515-1 978-0-471-00255-0. DOI: 10.1002/9781119115151 (cit. on pp. 148, 151, 157).

[DFC08] U. Demšar, S. A. Fotheringham, and M. Charlton. "Combining Geovisual Analytics with Spatial Statistics: The Example of Geographically Weighted Regression". In: *The Cartographic Journal* 45.3 (2008), pp. 182–192. DOI: 10.1179/000870408x311378 (cit. on pp. 148, 149).

[Dem+13] U. Demšar, P. Harris, C. Brunsdon, A. S. Fotheringham, and S. McLoone. “Principal Component Analysis on Spatial Data: An Overview”. In: *Annals of the Association of American Geographers* 103.1 (2013), pp. 106–128. DOI: 10.1080/00045608.2012.689236 (cit. on p. 146).

[Den96] B. D. Dent. *Cartography: Thematic Map Design*. 4th ed. Wm. C. Brown Publishers, 1996. 434 pp. ISBN: 978-0-697-22970-0 (cit. on p. 157).

[DB07] J. Dykes and C. Brunsdon. “Geographically Weighted Visualization: Interactive Graphics for Scale-Varying Exploratory Analysis”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1161–1168. DOI: 10.1109/tvcg.2007.70558 (cit. on p. 148).

[FBC02] A. S. Fotheringham, C. Brunsdon, and M. Charlton. *Geographically weighted regression: the analysis of spatially varying relationships*. John Wiley & Sons, 2002. ISBN: 0-471-49616-2 (cit. on p. 146).

[Gah+02] M. Gahegan, M. Takatsuka, M. Wheeler, and F. Hardisty. “Introducing GeoVISTA Studio: An Integrated Suite of Visualization and Computational Methods for Exploration and Knowledge Construction in Geography”. In: *Computers, Environment and Urban Systems* 26.4 (2002), pp. 267–292. DOI: 10.1016/s0198-9715(01)00046-1 (cit. on pp. 148, 149).

[GK15] M. G. Genton and W. Kleiber. “Cross-Covariance Functions for Multivariate Geostatistics”. In: *Statistical Science* 30.2 (2015), pp. 147–163. DOI: 10.1214/14-sts487 (cit. on p. 146).

[Goo+16] S. Goodwin, J. Dykes, A. Slingsby, and C. Turkay. “Visualizing Multiple Variables Across Scale and Geography”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 599–608. DOI: 10.1109/tvcg.2015.2467199 (cit. on p. 148).

[Gra] Gravitystorm. *Thunderforest Landscape*. URL: https://www.thunderforest.com/maps/landscape/ (visited on 11/15/2021) (cit. on p. 154).

[GB20] M. Gröbe and D. Burghardt. “Micro Diagrams: Visualization of Categorical Point Data from Location-Based Social Media”. In: *Cartography and Geographic Information Science* 47.4 (2020), pp. 305–320. DOI: 10.1080/15230406.2020.1733438 (cit. on p. 148).

[Guo08] D. Guo. “Regionalization with Dynamically Constrained Agglomerative Clustering and Partitioning (REDCAP)”. In: *International Journal of Geographical Information Science* 22.7 (2008). DOI: 10.1080/13658810701674970 (cit. on pp. 151, 155).

[Hal18] S. K. Haldar. “Chapter 5 - Exploration Geochemistry”. In: *Mineral Exploration (Second Edition)*. Ed. by S. K. Haldar. Elsevier, 2018, pp. 85–101. ISBN: 978-0-12-814022-2. DOI: 10.1016/b978-0-12-814022-2.00005-8 (cit. on p. 144).

[Har+15] P. Harris, A. Clarke, S. Juggins, C. Brunsdon, and M. Charlton. “Enhancements to a Geographically Weighted Principal Component Analysis in the Context of an Application to an Environmental Data Set”. In: *Geographical Analysis* 47.2 (2015), pp. 146–172. DOI: `https://doi.org/10.1111/gean.12048` (cit. on p. 146).

[Has+91] J. Haslett, R. Bradley, P. Craig, and A. Unwin. “Dynamic Graphics for Exploring Spatial Data with Application to Locating Global and Local Anomalies”. In: *The American Statistician* 45.3 (1991) (cit. on pp. 147, 164).

[Haz+20] S. Hazarika, H. Li, K.-C. Wang, H.-W. Shen, and C.-S. Chou. “NNVA: Neural Network Assisted Visual Analysis of Yeast Cell Polarization Simulation”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2020), pp. 34–44. DOI: `10.1109/tvcg.2019.2934591` (cit. on p. 149).

[HMZ09] T. Hrnčiarová, P. Mackovčin, and I. Zvara. *Atlas Krajiny České Republiky*. Praha, Průhonice: Ministerstvo životního prostředí ČR, Výzkumný ústav Silva Taroucy pro krajinu a okrasné zahradnictví, 2009. 332 pp. ISBN: 978-80-85116-59-5 (cit. on pp. 154, 162).

[JM00] T. Jankun-Kelly and K.-L. Ma. “A Spreadsheet Interface for Visualization Exploration”. In: *Proceedings Visualization 2000. VIS 2000 (Cat. No.00CH37145)*. 2000, pp. 69–76. DOI: `10.1109/visual.2000.885678` (cit. on p. 149).

[JF16] J. Johansson and C. Forsell. “Evaluation of Parallel Coordinates: Overview, Categorization and Guidelines for Future Research”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 579–588. DOI: `10.1109/tvcg.2015.2466992` (cit. on p. 149).

[Jol86] I. Jolliffe. *Principal Component Analysis*. New York: Springer, 1986 (cit. on p. 146).

[Jom+08] T. Jombart, S. Devillard, A.-B. Dufour, and D. Pontier. “Revealing cryptic spatial patterns in genetic variability by a new multivariate method”. In: *Heredity* 101 (2008), pp. 92–103. DOI: `https://doi.org/10.1038/hdy.2008.34` (cit. on p. 146).

[Kei00] D. Keim. “Designing Pixel-Oriented Visualization Techniques: Theory and Applications”. In: *IEEE Transactions on Visualization and Computer Graphics* 6.1 (2000), pp. 59–78. DOI: `10.1109/2945.841121` (cit. on p. 148).

[Kei+08] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. “Visual Analytics: Definition, Process, and Challenges”. In: *Information Visualization*. Ed. by A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North. 4950. Berlin, Heidelberg: Springer, 2008, pp. 154–175. ISBN: 978-3-540-70955-8 978-3-540-70956-5. DOI: `10.1007/978-3-540-70956-5_7` (cit. on p. 145).

[Kim+13] S. Kim, R. Maciejewski, A. Malik, Y. Jang, D. S. Ebert, and T. Isenberg. "Bristle Maps: A Multivariate Abstraction Technique for Geovisualization". In: *IEEE Transactions on Visualization and Computer Graphics* 19.9 (2013), pp. 1438–1454. doi: 10.1109/tvcg.2013.66 (cit. on p. 148).

[LCX19] S. V. Liu, F.-l. Chen, and J. Xue. "A Meta-Analysis of Selected near-Road Air Pollutants Based on Concentration Decay Rates". In: *Heliyon* 5.8 (2019), e02236. doi: 10.1016/j.heliyon.2019.e02236 (cit. on p. 151).

[Mat+18] J. Matejka, M. Glueck, E. Bradner, A. Hashemi, T. Grossman, and G. Fitzmaurice. "Dream Lens: Exploration and Visualization of Large-Scale Generative Design Datasets". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Montreal QC, Canada: ACM Press, 2018, pp. 1–12. isbn: 978-1-4503-5620-6. doi: 10.1145/3173574.3173943 (cit. on p. 149).

[ML19] L. McNabb and R. S. Laramee. "Multivariate Maps—A Glyph-Placement Algorithm to Support Multivariate Geospatial Visualization". In: *Information* 10.10 (2019), p. 302. doi: 10.3390/info10100302 (cit. on p. 148).

[MH18] M. E. Meseery and O. Hoeber. "Geo-Coordinated Parallel Coordinates (GCPC): Field Trial Studies of Environmental Data Analysis". In: *Visual Informatics* (2018), p. 14 (cit. on p. 149).

[MA14] S. Miksch and W. Aigner. "A Matter of Time: Applying a Data-users-tasks Design Triangle to Visual Analytics of Time-oriented Data". In: *Computers & Graphics* 38 (2014), pp. 286–290. doi: 10.1016/j.cag.2013.11.002 (cit. on p. 151).

[MW20] A. Mohiuddin and R. Woodbury. "Interactive Parallel Coordinates for Parametric Design Space Exploration". In: *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. Honolulu HI USA: ACM, 2020, pp. 1–9. isbn: 978-1-4503-6819-3. doi: 10.1145/3334480.3383101 (cit. on p. 149).

[Mue21] C. Muehlmann. "Advances in Blind Source Separation for Spatial Data". PhD thesis. Vienna, Austria: TU Wien, 2021, p. 100574. doi: 10.1016/j.spasta.2021.100574 (cit. on pp. 144, 147, 156).

[MNV21] C. Muehlmann, K. Nordhausen, and J. Virta. *SpatialBSS: Blind Source Separation for Multivariate Spatial Data*. R package version 0.12-0. 2021 (cit. on p. 152).

[MNY20] C. Muehlmann, K. Nordhausen, and M. Yi. "On Cokriging, Neural Networks, and Spatial Blind Source Separation for Multivariate Spatial Prediction". In: *IEEE Geoscience and Remote Sensing Letters* 18 (2020), pp. 1931–1935. doi: 10.1109/lgrs.2020.3011549 (cit. on p. 147).

[Nor+15] K. Nordhausen, H. Oja, P. Filzmoser, and C. Reimann. “Blind Source Separation for Spatial Compositional Data”. In: *Mathematical Geosciences* 47.7 (2015), pp. 753–770. DOI: 10.1007/s11004-014-9559-5 (cit. on pp. 144, 147).

[OR18] T. Opach and J. K. Rød. “Augmenting the Usability of Parallel Coordinate Plot: The Polyline Glyphs”. In: *Information Visualization* 17.2 (2018), pp. 108–127. DOI: 10.1177/1473871617693041 (cit. on p. 149).

[Orb+19] D. Orban, D. F. Keefe, A. Biswas, J. Ahrens, and D. Rogers. “Drag and Track: A Direct Manipulation Interface for Contextualizing Data Instances within a Continuous Parameter Space”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 256–266. DOI: 10.1109/tvcg.2018.2865051 (cit. on p. 149).

[Pic21] N. Piccolotto. *npiccolotto/sbss-vis: Visual Parameter Selection for Spatial Blind Source Separation*. GitHub. 2021. URL: https://github.com/npiccolotto/sbss-vis (visited on 02/25/2022) (cit. on p. 152).

[R C23] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2023 (cit. on p. 151).

[Rei+14] C. Reimann, M. Birke, A. Demetriades, P. Filzmoser, and P. O’Connor. *Chemistry of Europe’s Agricultural Soils. Part A: Methodology and Interpretation of the GEMAS Data Set*. Vol. Heft 102. Reihe B. Hannover: Bundesanstalt für Geowissenschaften und Rohstoffe (BGR), 2014 (cit. on pp. 144, 158).

[Rei+98] C. Reimann, M. Äyräs, V. A. Chekushin, I. V. Bogatyrev, R. Boyd, P. de Caritat, R. Dutter, T. E. Finne, J. H. Halleraker, J. Øystein, G. Kashulina, H. Niskavaara, et al. *Environmental Geochemical Atlas of the Central Barents Region*. Trondheim: NGU Norges Geologiske Undersøkelse, 1998. ISBN: 82-7385-176-1 (cit. on p. 158).

[Rei+08] C. Reimann, P. Filzmoser, R. Garrett, and R. Dutter. *Statistical Data Analysis Explained*. First Edition. John Wiley & Sons, Ltd, 2008. ISBN: 978-0-470-98581-6. DOI: 10.1002/9780470987605 (cit. on pp. 161, 163).

[Sed+14] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. “Visual Parameter Space Analysis: A Conceptual Framework”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2161–2170. DOI: 10.1109/tvcg.2014.2346321 (cit. on p. 149).

[SEK10] D. B. Smith, K. J. Ellefsen, and J. E. Kilburn. *Geochemical Data for Colorado Soils: Results from the 2006 State-Scale Geochemical Survey*. Data Series 520. 2010 (cit. on p. 158).

[TC05] J. J. Thomas and K. A. Cook. *Illuminating the Path: An R&D Agenda for Visual Analytics*. National Visualization and Analytics Center, Department of Homeland Security, IEEE Computer Society, 2005. ISBN: 0-7695-2323-4 (cit. on p. 145).

[Tur+14] C. Turkay, A. Slingsby, H. Hauser, J. Wood, and J. Dykes. "Attribute Signatures: Dynamic Visual Summaries for Analyzing Multivariate Geographical Data". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2033–2042. DOI: `10.1109/tvcg.2014.2346265` (cit. on p. 148).

[Ume+14] N. Umetani, Y. Koyama, R. Schmidt, and T. Igarashi. "Pteromys: Interactive Design and Optimization of Free-Formed Free-Flight Model Airplanes". In: *ACM Transactions on Graphics* 33.4 (2014), pp. 1–10. DOI: `10.1145/2601097.2601129` (cit. on p. 149).

[Wac03] H. Wackernagel. *Multivariate geostatistics: an introduction with applications*. Berlin, Heidelberg: Springer Science & Business Media, 2003. DOI: `10.1007/978-3-662-05294-5` (cit. on pp. 144, 146).

[Wal+19] E. Wall, M. Agnihotri, L. Matzen, K. Divis, M. Haass, A. Endert, and J. Stasko. "A Heuristic Approach to Value-Driven Evaluation of Visualizations". In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 491–500. DOI: `10.1109/tvcg.2018.2865146` (cit. on pp. 145, 158, 160).

[Wan+17] J. Wang, X. Liu, H.-W. Shen, and G. Lin. "Multi-Resolution Climate Ensemble Parameter Analysis with Nested Parallel Coordinates Plots". In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 81–90. DOI: `10.1109/tvcg.2016.2598830` (cit. on p. 149).

[Was+10] J. Waser, R. Fuchs, H. Ribičič, B. Schindler, G. Blöschl, and E. Gröller. "World Lines". In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 1458–1467. DOI: `10.1109/tvcg.2010.223` (cit. on p. 149).

[Wic16] H. Wickham. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN: 978-3-319-24277-4 (cit. on p. 151).

[Zha+21] J. Zhao, X. Liu, C. Guo, C. Qian, and Y. V. Chen. "Phoenixmap: An Abstract Approach to Visualize 2D Spatial Distributions". In: *IEEE Transactions on Visualization and Computer Graphics* 27.3 (2021), pp. 2000–2014. DOI: `10.1109/tvcg.2019.2945960` (cit. on p. 148).

[Zho+17] M. Zhou, J. Tian, F. Xiong, and R. Wang. "Point Grid Map: A New Type of Thematic Map for Statistical Data Associated with Geographic Points". In: *Cartography and Geographic Information Science* 44.5 (2017), pp. 374–389. DOI: `10.1080/15230406.2016.1160797` (cit. on p. 148).

CHAPTER 5

Visual Sensitivity Analysis for Spatial Blind Source Separation

> The content of this chapter is published in:
>
> Nikolaus Piccolotto, Markus Bögl, Christoph Muehlmann, Klaus Nordhausen, Peter Filzmoser, Johanna Schmidt, Silvia Miksch: *Data Type Agnostic Visual Sensitivity Analysis*. IEEE Transactions on Visualization and Computer Graphics, vol. 30, no. 1, 2024. DOI: 10.1109/TVCG.2023.3327203.

Context. In the following publication, we consider again a non-stationary SBSS model (Section 1.1.3) and, after supporting *optimization* in the previous publication (Chapter 4), focus on the PSA task *sensitivity* analysis. SBSS parameters and outputs are not of the kind of multivariate data that many numerical sensitivity analysis techniques are designed for, so their applicability is uncertain. Together with our collaborators (Christoph Muehlmann, Klaus Nordhausen, and Peter Filzmoser) we propose a visual-interactive approach that builds on a simple sensitivity index derived from the cluster diameter difference in parameter and output space. The main proposed visualization is the *Discrepancy Dendrogram*, which indicates by color if clusters are wider or narrower in the compared data space. We evaluated the prototype heuristically with visualization experts and in interviews with SBSS experts. Since our approach requires only dissimilarity information of data points, we could also show its transferability to another dataset and problem context. To this end, we interviewed a microclimate simulation expert together with Johanna Schmidt from VRVis GmbH. While the visualization experts were somewhat in disagreement about whether our VA prototype employs "good" visualizations, the other experts identified sensitive and stable parameter ranges with the help of our prototype.

RQ's Concerned: RQ1, RQ2, RQ3, RQ4.

5.1 Abstract

Modern science and industry rely on computational models for simulation, prediction, and data analysis. Spatial blind source separation (SBSS) is a model used to analyze spatial data. Designed explicitly for spatial data analysis, it is superior to popular non-spatial methods, like PCA. However, a challenge to its practical use is setting two complex tuning parameters, which requires parameter space analysis. In this paper, we focus on sensitivity analysis (SA). SBSS parameters and outputs are spatial data, which makes SA difficult as few SA approaches in the literature assume such complex data on both sides of the model. Based on the requirements in our design study with statistics experts, we developed a visual analytics prototype for data type agnostic visual sensitivity analysis that fits SBSS and other contexts. The main advantage of our approach is that it requires only dissimilarity measures for parameter settings and outputs. We evaluated the prototype heuristically with visualization experts and through interviews with two SBSS experts. In addition, we show the transferability of our approach by applying it to microclimate simulations. Study participants could confirm suspected and known parameter-output relations, find surprising associations, and identify parameter subspaces to examine in the future. During our design study and evaluation, we identified challenging future research opportunities.

5.2 Introduction

In many domains, data analysis requires dealing with multivariate measurements in space. For instance, mining corporations and public agencies may analyze geochemical soil samples for mine prospecting or investigating environmental pollution, respectively. Depending on the specific goal and application, various tasks, e.g., dimensionality reduction or finding meaningful linear combinations of variables, must be carried out on such datasets. Spatial blind source separation (SBSS) [Nor+15; Bac+20; MBN22] is designed explicitly for multivariate spatial data and reveals linear combinations of such data. SBSS offers various benefits compared to alternative methods, e.g., it keeps the well-known loading-scores scheme from principal component analysis and adequately accounts for spatial dependence due to its model-based approach. Therefore, latent dimensions identified with SBSS often correspond to the physical reality where data was collected, making it an excellent analysis tool for spatial data. A detailed description of SBSS is out of scope for this paper, and we refer interested readers to [Nor+15; Pic+22b; MBN22]. SBSS has been successfully applied to a geochemical dataset [Nor+15] and may be potentially used in any application domain that involves multivariate quantitative measurements at different locations.

SBSS requires setting two complex tuning parameters: A partition of the spatial domain in non-overlapping regions (regionalization) and a ring-shaped point neighborhood (kernel). On the other side of the model (Figure 5.2), SBSS yields a set of latent spatial dimensions (i.e., maps), where each is a linear combination of original dimensions with weights (loadings) given by the unmixing matrix. Consequently, parameter space analysis tasks

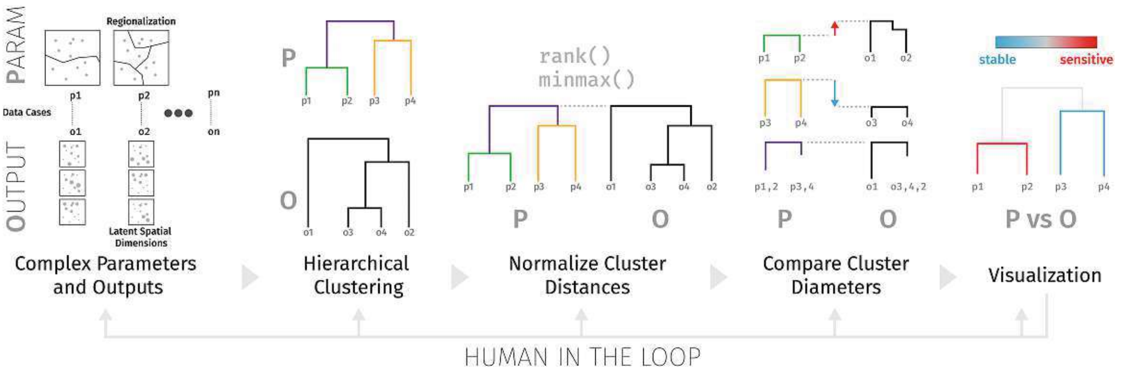


Figure 5.1: Illustration of the visualization pipeline used for the Discrepancy Dendrogram. The key idea of it is to use cluster diameters as a measure for variation in complex parameter settings and outputs. Given dissimilarity measures for each, we perform hierarchical clustering separately in each space. Distances are normalized by ranking or min-max normalization for comparison. For every cluster obtained through hierarchical clustering, we evaluate the difference in its diameter and visualize that by color.

[Sed+14] become relevant. Previous work [Pic+22b] focused on the *optimization* task, but *sensitivity analysis* (SA) is considered equally important for SBSS. SA compares the relative variation in parameter settings and output of the model, thus highlighting relevant/irrelevant parameters and their stable/sensitive ranges. This analysis is essential to obtain and communicate reliable results, i.e., those not a consequence of luck and coincidence. SA is especially important for SBSS as it lacks so far any goodness-of-fit criteria; hence deciding between alternative parameter settings is challenging. SA can help with this decision as in prior work on blind source separation [Pic+22a; Pic+22b], analysts noted that they find stable parameter settings more trustworthy and associated outputs more likely to be the "real" solution. SA may thus further strengthen the outcome of an *optimization* task and, additionally, inform geostatistical modeling: If, e.g., the regionalization parameter barely influences the output, analysts might reasonably suspect that the input dataset is spatially stationary (a geostatistical modeling decision).

SBSS is interesting for the visualization community primarily because of the mentioned affordances of its parameters and outputs: Parameter settings and outputs are spatial objects or otherwise complex in a way that a multivariate representation does not do them justice. While the literature contains many examples of visual parameter space exploration [Sed+14; PBM23], to the best of our knowledge, none of them support complex parameters and outputs without resorting to multivariate representation or feature derivation (Section 5.3). However, these requirements are not specific to SBSS, as many examples exist for models with complex parameters *and* outputs. For instance, spatial or time-varying inputs and outputs can arise in microclimate simulations [Vuc+22]. They predict meteorological variables (e.g., air temperature, humidity, or wind speed) in a small area, typically for a single street or building.

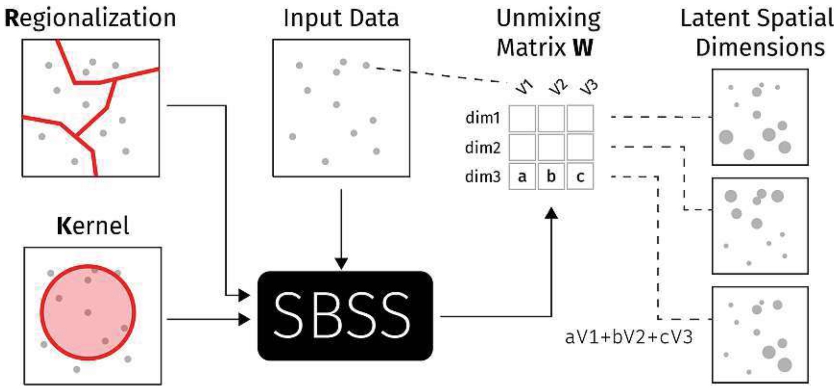


Figure 5.2: SBSS [Nor+15; Pic+22b; MBN22] takes a regionalization (R) and a kernel (K) as parameters and outputs a linear combination of input variables (latent spatial dimensions), described by the unmixing matrix (W).

We intend to close this gap with our paper. The core idea of our proposal is illustrated in Figure 5.1: We take a cluster's diameter as a measure of variation for the contained parameter settings or associated outputs (referred to as data cases, respectively). Then we can enable SA for SBSS in the following way. Given appropriate dissimilarity measures for data cases, we compute pairwise distances in each space (parameter and output), based on which a hierarchical clustering is produced. After normalizing distances, we compute the diameter difference of all clusters between one space and the another. This information is then presented in our main visualization, the Discrepancy Dendrogram. Supporting visualizations complete required user tasks. In particular, the contributions of our design study are that we

- propose a task abstraction for SA in the context of SBSS (Section 5.4);
- based on SBSS requirements, develop a visualization that supports SA and works on any data type (Figure 5.1, Section 5.5);
- integrate this and other visualizations in a visual analytics prototype (Section 5.6);
- evaluate the prototype with experts in visualization (Section 5.7.1) and SBSS (Section 5.7.2);
- show the transferability to other problems by applying our approach to microclimate simulations (Section 5.7.3).

5.3 Related Work

5.3.1 Sensitivity Analysis

Sensitivity Analysis (SA) is "the study of how the uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input [Sal02, p. 1].” SA allows analysts to determine how variations in the input influence the output. A broad distinction between various SA methods can be drawn at whether they are *local* or *global* [Sal+07]. Local methods are applicable when the model is linear as they yield, e.g., a partial derivative according to one parameter. An example of such local methods is the one-at-a-time approach, where one parameter is varied while the others are kept fixed. Global methods, on the other hand, are applicable to non-linear models, too. A well-known example is the Sobol index [Sob90], a variance-based global SA method. Several surveys exist [Ham94; IC04; CI04; IL15; BP16; Sal+19] that collect and discuss both local and global methods. Methods covered in these surveys mainly consider models with multivariate parameters, e.g., the output scalar y is a function of an input vector $\mathbf{x}$: $y = f(\mathbf{x})$. Spatially-varying parameters [LT09; Rai+19] or outputs [Mar+11; Lig13] have been considered as well. However, these methods do not fit to SBSS (Figure 5.2).

5.3.2 Visual Parameter Analysis

Visual parameter analysis (VPA) has a long history in the visualization literature, with seminal works published in the 1990s, like Design Galleries [Mar+97] or spreadsheet interfaces [JM00]. Sedlmair et al. [Sed+14] provided a common data flow model and a task taxonomy, such as optimization, uncertainty, or SA. Piccolotto et al. [PBM23] surveyed user interfaces and visualizations that support visual parameter space exploration. Several examples of VPA for multivariate parameters can be found in the literature [YBP21; Kni+21; Cib+20; Paj+17; Ber+13; Guo+11]. However, these approaches do not apply to SBSS parameters.

Many approaches have been used when it comes to visualizing parameter-output relations [PBM23]. When parameters are multivariate, visualizations that show correlations and trends can be used to carry out SA, such as histograms, scatterplots, or PCPs [Beh+14; Wan+17; Cib+23]. These visualizations are often *juxtaposed* and linked, such that selections in one view highlight the same data in other views [Mat+17]. Another option is to *embed* parameters and outputs in the same visualization, e.g., by encoding them as axes in the same PCP [Ste+13] or by color-coding a 3D model [Dor+15]. A consequence of juxtaposition is that general visualization-independent approaches may be used together. E.g., first grouping data cases by similarity, then inspecting properties of individual groups [BM10; AE20; Haz+20] is popular. Orban et al. [Orb+19] devised two linked dimensionally-reduced (DR) scatterplots, an approach that can generally be extended to complex data and SBSS parameters/outputs. However, our target users struggled with DR scatterplots in previous work [Pic+22a]. The difficulty was that the DR spatializations looked like scatterplots but did not show the same information and required a different way of reading, which was unintuitive to them. Therefore, we developed an alternative approach. A more specific form of juxtaposition is to *align* data cases in useful ways that highlight dependencies between parameters and outputs, e.g., as part of a spreadsheet [Lub+14; Lub+15; EST20]. The idea is that dependencies become visible when the spreadsheet is sorted by multiple columns. However, it requires a compact visual representation. *Superposition* may be possible if parameter and output refer to the same space, such as particle trajectories and their initial position [GT16]. *Sequential Superposition* leverages a system's interactivity. The analyst may rapidly browse between parameter/output pairs, and sudden visual jumps in the emerging animation point to sensitive parameter ranges [Sch+17; RGG18; He+20]. Parameter and output visualizations may also be *integrated* with explicit links drawn between them. E.g., a trapezoid that connects parameter and output histograms shows sensitivity by the relative length of horizontal segments [Wei+16]. Another option for composite visualizations of parameters and outputs for SA is *nesting*, i.e., putting visualizations inside the marks of another, like correlation matrices in an interval tree [EST20].

Data mining methods may also support visual SA. E.g., if regression analysis between parameter and output is possible, that information can be shown in the parameter visualization in the spirit of scented widgets [WHA07; KSI14; Des+19]. Correlation analysis between parameters and derived output features may also be done if they lend themselves to it [EST20]. Developing a surrogate model augmenting the original model with fast but inaccurate output predictions for new parameter settings is standard practice in VPA [Sed+14]. It may be possible to extract information from the surrogate to support SA, such as parameters in linear regression [Mat+17], or partial derivatives in neural networks [Haz+20].

Generally, in existing work, either the parameter (by multivariate representation) or the output (by feature derivation) must have multivariate characteristics. Our contribution to visual sensitivity analysis enables it in situations where both parameter and output are of complex data types, e.g., spatial objects.

5.3.3 Visual Cluster Analysis and Clustering Comparison

Clustering is an essential wide-spread class of data analysis methods, and various flavors were proposed over time [XT15]. Generally, clusterings partition data cases into coherent groups according to a distance function. Visual inspection of these groups may reveal previously hidden patterns. To visualize the whole clustering, nowadays, color-coded dimensionally-reduced scatterplots are commonly employed [Kwo+18; CD19; Xia+22]. However, these scatterplots are only approximate, as they contain projection errors [Jeo+22], and may require specialized knowledge to interpret [WVJ16]. Glyph-based visualizations [Cao+11] were proposed in the context of geospatial data. Dendrograms [SS02; Gal15] commonly depict hierarchical clusterings. Blanch et al. [BDB15] proposed the Dendrogramix, a combination of dendrogram and matrix visualization. The clustering outcome depends on the specific algorithm and parameters, so visualizations were proposed to compare these. However, they focus on the analysis of cluster members [CD19], comparison of clusterings concerning parameters [CD19] or algorithms [LYi+15; Kwo+18]. I.e., the definition of distance between data cases is fixed. Our work may be seen as comparing clusterings with alternative distances (Figure 5.1).

5.4 Users & Task Abstraction

As in previous work on SBSS [Pic+22b], our primary users are experts in statistics. We anticipate our user base to eventually include domain experts, e.g., from geochemistry. We conducted an extensive literature review [PBM23] to understand how visual VPA and, consequently, visual SA work in other contexts. Based on that, we distilled generic SA sub-tasks to enable SA on the SBSS-specific complex parameters with our clustering-based approach (T1–T5). We presented and discussed them with our collaborators (statistics/SBSS experts who are co-authors of this paper) to ensure their suitability. Based on these tasks, we developed the main visualization (Section 5.5).

Tasks. First, to start the analysis, analysts must *compare the association between parameters and outputs* (T1). Pairs of highly associated parameters and outputs are less interesting to investigate. For any given parameter/output, they must *assess its overall variation* (T2) to learn about contained similarity structures and outliers. Furthermore, analysts must *identify groups of data cases with low/high variation in a parameter/output* (T3) in order to *compare variation between parameters and outputs, both overall and for a group of data cases* (T4). To support analysts in reasoning why this variation happens, they must be able to *view individual data cases* (T5).

Guidelines. In addition to user tasks, we formulate three design guidelines for the visualizations. These were informed by evaluations conducted in our past work [Pic+22a; Pic+22b] and by widely used visualization guidelines. First, *visual marks of similar values should be adjacently arranged* (D1). This visual requirement suggests continuity that scalars exhibit naturally, but complex objects do not. It will make it easier to perceive stable/sensitive parameter ranges. *Occlusion must be avoided* (D2) to not clutter the display. The visualization should, if possible, *resemble a familiar graphic* (D3) that our target users are familiar with.

5.5 Discrepancy Dendrogram

We describe in this section how our main visualization, the Discrepancy Dendrogram, is constructed (also compare Figure 5.1). The complete VA prototype will be discussed in the following section. We aim for a visual-interactive approach for two reasons. First, we did not find numerical SA approaches that are applicable to our data (Section 5.3.1). Second, our approach needs configuration (e.g., Section 5.5.2 or Section 5.5.3), where each choice highlights different patterns (compare Figure 5.10), impacting the conclusions to draw. Thus, in an interactive setting, the analyst can quickly change between those configurations and thoroughly compare them (see, e.g., Section 5.7.2).

The core of SA is to compare the relative variation in parameter settings and outputs. It can readily be quantified for numbers (cf. variance-based SA approaches), but measuring *variation* for complex objects, like the spatial SBSS parameters, is not straightforward.

Our proposal's core idea (Figure 5.1) is to consider cluster diameters for that purpose: A cluster gets wider the more dissimilar contained data cases are. Conversely, the cluster diameter is zero when all contained data cases are the same. There are advantages to that approach. First of all, a clustering can be obtained when only pairwise similarity information (Section 5.5.1) is available. Thus a formal notion of variation need not exist for the data type at hand. Second, cluster analysis generally supports tasks T2 and T3 when one investigates global cluster structures (e.g., how many exist, how many data cases they contain) and local structures (e.g., finding outlier cases). Hence we propose to augment a visualization of cluster structures with the information required for SA, i.e., whether clusters shrink or expand when applying another dissimilarity measure to the data cases. This approach can be seen as *orienting* guidance [Cen+17] that points analysts to interesting data cases. The major available choices at this point are i) the type of visualization and ii) how to compute the augmenting information. The two choices are independent, and we focus on the latter before discussing the former in Section 5.5.5.

Sampling. Any parameter space analysis task requires a reasonable set of (parameter setting, output) tuples. Common desired sampling properties are that it is uniform and spans a large part of the parameter space, which is achieved via automated sampling techniques. These are hard problems for SBSS, where two random parameter settings are not a-priori equally reasonable. Domain knowledge critically informs parameter selection in SBSS [Pic+22b]. Single-execution runtimes measured in minutes or hours further complicate the issue. Thus, following study participants' current practices in SBSS and microclimate simulations, we rely on a few dozen, mostly manually selected, parameter settings and limit SA insights to that subspace. While not solving everything at once, our approach still improves their current situation.

5.5.1 Dissimilarity Measures

Dissimilarity measures, considerably the basic requirement for any analysis, exist for many data types. A dissimilarity measure is a function $d(\cdot,\cdot) \to \mathbb{R}^+$ that quantifies how similar two objects are. Generally, we expect that $d(a,b) = 0$ iff $a = b$ and that $d(a,b)$ is strictly monotonically increasing with the differences between a and b. We assume such a dissimilarity measure for every model parameter and output.

5.5.2 Hierarchical Clustering

Flat partitioning cluster algorithms, like k-means, divide the dataset into an a-priori specified number of groups while minimizing intra-group distances. On the other hand, hierarchical clustering algorithms retain all cluster structures in the dataset and, therefore, do not require a k parameter. Hierarchical clustering is thus preferable because it will contain all possible clusters the analyst might be interested in, and we can enumerate them. We chose a clustering by agglomerative nesting (AGNES) [Roh82] because bottom-up hierarchical clustering is easier to think about and, thus, easier to explain to analysts than the top-down variant. Further, many current alternatives, such as HDBSCAN [CMS13], require Euclidean distances and can not be used with just dissimilarities. The main parameter of AGNES is the linkage criterion, i.e., how to compute the distance between two clusters. Only some linkage criteria *can* be used in our case. E.g., centroid-based variants like Ward's method are not applicable as the concept of a centroid may not exist for complex data types, such as regionalizations. Consequently, we provide complete and average linkage as user-selectable hierarchical clustering parameters.

5.5.3 Normalize Cluster Distances

We aim to evaluate whether a given cluster shrinks or expands when an alternative dissimilarity measure $d_A()$ is applied. The obvious problem here is that $d()$ and $d_A()$ might have differing images, i.e., one maps to the unit interval $[0, 1]$ while the other maps onto $[0, 1312]$. We propose ranking or min-max normalization to solve this issue. Both operations work on a distance matrix. Ranking replaces values in all cells by their rank, while min-max normalization maps values onto the unit interval. When comparing ranks, the focus will naturally be on ordinal changes, ignoring magnitude. Min-max normalization, on the other hand, preserves magnitude. The analyst can switch between the two as both approaches have advantages and drawbacks (compare Figure 5.10).

5.5.4 Compare Cluster Diameters (Sensitivity Index)

Finally, we require a way to measure a cluster's diameter, which roughly corresponds to the linkage criterion in Section 5.5.2. To find candidates, we turn to internal clustering validation measures [Liu+10], as no external information exists in our case. These usually incorporate the *compactness* of clusters, which measures the variation within a cluster. Based on the selected linkage criterion, we use the largest distance between any two elements (complete linkage) or the average distance between all elements (average linkage).

Given two distance-normalized hierarchical clusterings P and O (e.g., one with distances of parameter settings and one with output distances) and a cluster diameter definition, we can compute by how much a cluster in P shrinks or expands in O, or the other way around, as P and O cluster the same data cases. We evaluate the *index()* function (Alg. 5.1) for every cluster, i.e., every horizontal line in a dendrogram. $D_{\{1,2\}}$ are the respective distance matrices of P and O. The subroutine `upperTri` returns the upper triangle of a square matrix, and `select` selects specified rows and columns of a square matrix. The function can be seen as a sensitivity index as it quantifies how much the variation differs between the parameter and output space.

5.5.5 Visualization

Two established visualization idioms for clusterings are dimensionally-reduced scatterplots and dendrograms. As our target users (statistics experts) found the former approach in previous work [Pic+22a] rather unintuitive, we chose the latter for our context, fulfilling design guideline D3 (Section 5.4). Additionally, a dendrogram supports many other

Algorithm 5.1: Pseudocode of sensitivity index computations.

```
Data: Cluster $C$ of data cases, normalized distance matrices $D_1$ and $D_2$, cluster
      diameter definition $diam()$.
Function $index(C, D_1, D_2, diam)$ is
    $D_1[C] \leftarrow$ upperTri(select($D_1, C$))
    $D_2[C] \leftarrow$ upperTri(select($D_2, C$))
    return $diam(D_1[C]) - diam(D_2[C])$
end
```

guidelines and user tasks. The leaves are juxtaposed (D2), and similar leaves, which are joined into clusters earlier than dissimilar leaves, naturally appear adjacent (D1). Optimal leaf orderings may be used [BGJ01]. Lines encode the diameter of every possible cluster that could be interesting (T2–T3). These lines do not overlap (D2). The open challenges are encoding the sensitivity index (Section 5.5.4) in the dendrogram (T4) and ensuring that visualizations of data cases are visible (T5).

The free visual channels in a dendrogram we could use to support T4 are line color (hue, saturation), line texture (e.g., dashed or dotted), and line thickness. We encoded the sensitivity index in color hue (compare Figure 5.1). The index diverges with 0 at the center. Hence, the direction is as important as the magnitude. Two-directional encodings are standard for color hue (diverging scales) but very uncommon for the other attributes and likely confusing for our target users. We use two diverging scales dependent on the choice of distance normalization (Section 5.5.3): Red–blue (ranked) and purple–green (min-max). By default, the color scale spans the whole theoretically possible index interval, but the analyst may instead use the interval as found in the dataset to highlight small-scale patterns.

To support task T5, we show customized space-efficient visualizations as leaves of the Discrepancy Dendrogram (Figure 5.5-A, bottom). There is little available space when the dendrogram shows many data cases. We combat this issue with several strategies. First, clusters of the dendrogram can be hidden. Second, when leaves are clicked, a tooltip containing a more detailed visualization appears. I.e., we show the regionalization parameter of SBSS as flat polygons in the dendrogram and as an interactive Leaflet map in tooltips. Any cluster can be selected to be shown in the Gallery (Figure 5.5-B). More interactions are described in Section 5.6.1.

5.5.6 Interpretation, Notation and Example

The choice of the color scale's orientation is arbitrary. We decided that red (purple) highlights an expanded cluster while blue (green) marks shrunk clusters in the alternative distance (O in Figure 5.1). Consequently, interpretations regarding stability or sensitivity depend on how parameters and outputs are assigned to primary and alternative distances (Figure 5.3). E.g., sensitive parameter settings are associated with wider clusters in the

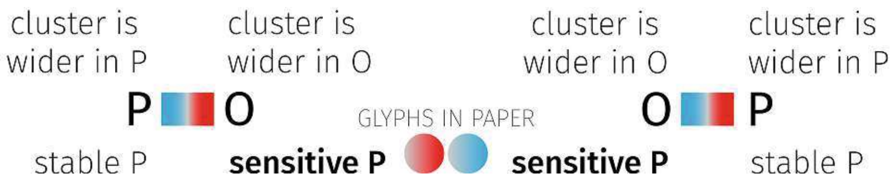


Figure 5.3: Parameter assessment changes depending on the assignment to primary and alternative distance in the Discrepancy Dendrogram. Glyphs in the document show the color of wider output clusters.

output space compared to the parameter space, which can appear as blue (parameter as primary distance) or red (parameter as alternative distance). In the remainder of the paper, we will use appropriate glyphs to denote the direction of sensitive parameters. A ●XY Discrepancy Dendrogram will thus i) compare X and Y, ii) show a dendrogram of clusters in X, iii) mark data cases with sensitive parameter settings as red.

Figure 5.4 shows a ●XY Discrepancy Dendrogram for the function $y = x^2$ sampled uniformly in the interval $[-4, 4]$. The dendrogram separates the parameter space into three clusters with X 1.4 to 4, -4 to -1.9, and -1.8 to 1.3 (from left to right). The lines' hue may be interpreted as the absolute gradient: Red lines mark wider clusters in Y (high) while the right-most cluster is gray (low). When plotted as a line chart, these patterns would refer to the parabola arms (red clusters) and the part between them (gray) as visible in the inset.

5.6 Visual Analytics Prototype

To facilitate SA of SBSS parameters and outputs, we propose a visual analytics prototype (Figure 5.5). We developed it in a user-centered design process in collaboration with statistics experts, who are co-authors of this paper. Links to a web version of the software are available in the supplemental material of the paper.

5.6.1 Discrepancy Dendrogram (T2–T5, D1–D3)

We discuss the construction of the Discrepancy Dendrogram in Section 5.5 and focus here on interactions. We provide several interactions with the Discrepancy Dendrogram to allow detailed investigation of clusters and to scale it to larger datasets. First of all, the user may choose between ranked and normalized distances (Section 5.5.3) and select the bounds of the color scale (Figure 5.5, left top). Further, they may choose the linkage criterion for the dendrogram (Section 5.5.2), which also affects cluster diameter computations (Section 5.5.4). Second, there might be multiple parameters and outputs in a given dataset. The analyst can thus select which parameter/output to build the dendrogram with (primary distance), which parameter/output to compare it to in the sensitivity index calculation (alternative distance), and which parameter/output to show in the dendrogram leaves (Figure 5.5-A1 to A3). As color hue is not a precise visual

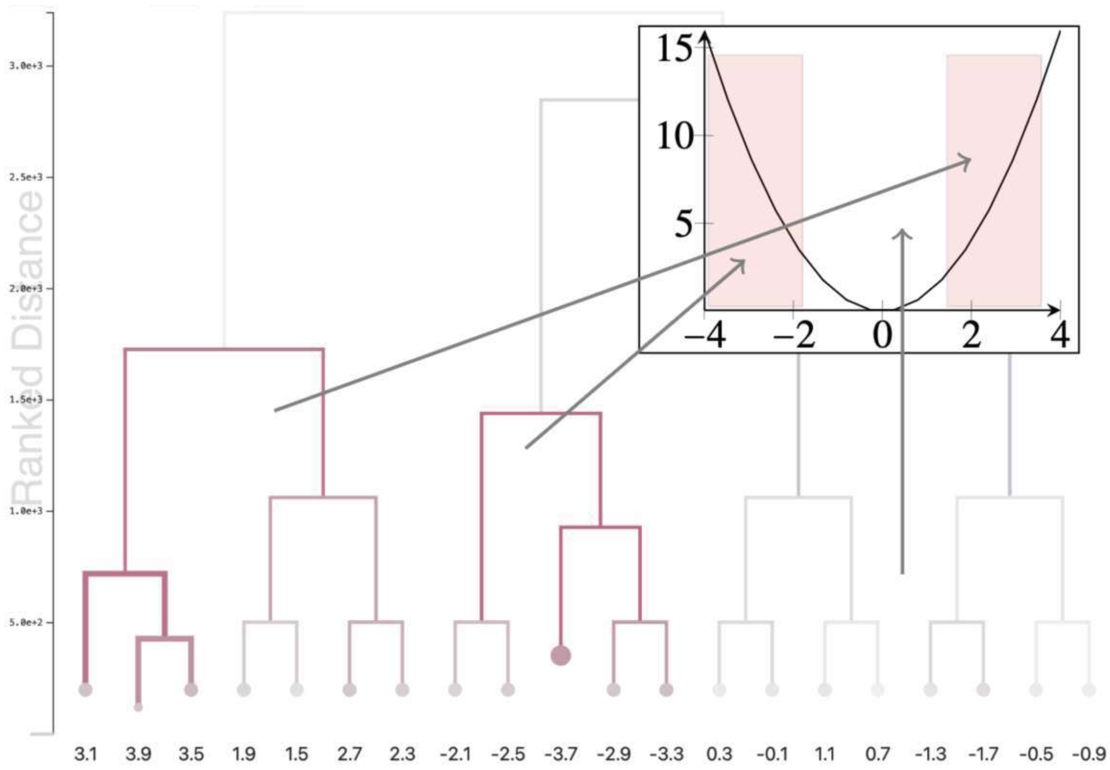


Figure 5.4: ●XY Discrepancy Dendrogram of the function $y = x^2$ (inset top right), with some clusters collapsed for readability. Red color highlights clusters that are wider in Y than X (=sensitive parameter ranges, i.e., marked parabola arms in inset).

channel, we encode a cluster's diameter difference additionally in the length of a vertical line segment next to the dendrogram's Y-axis legend. Other dendrogram interactions are more concerned with scalability. It is possible to collapse a cluster (shift + click), collapse all other clusters (meta + click), and collapse all clusters below a user-defined height (click on the Y-axis). These interactions free up additional display space for an area of interest. Colored circles replace collapsed clusters. The circle size is proportional to the amount of data cases in the cluster, while the color corresponds to the clicked line's color. The data cases of a collapsed cluster are replaced by a cluster representative. Finally, a cluster can be selected, after which contained data cases are shown in the Gallery (Figure 5.5-B).

5.6.2 Gallery (T5)

The Gallery shows data cases of a selected cluster in a grid (Figure 5.5-B). The number of columns and their width can be selected by the analyst, as can the sort order of data cases and which parameter or output they should show. It is possible, e.g., to sort parameter visualizations by output similarity, as is often done in visual parameter space analysis [Lub+14; EST20]. Thus, the Gallery can show complex patterns.

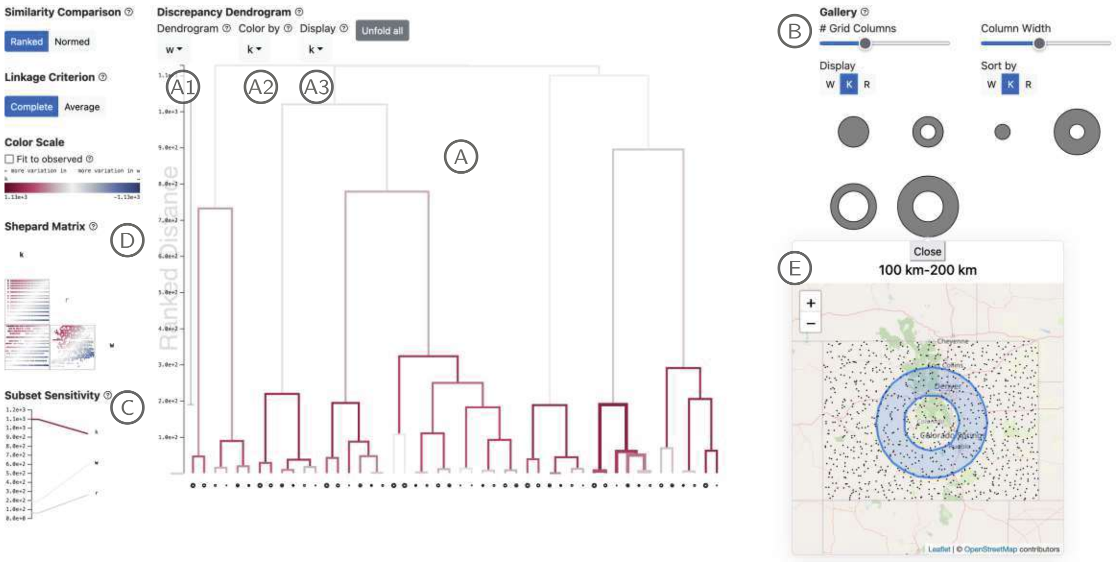


Figure 5.5: Screenshot of our prototype showing 48 SBSS parameters and outputs (Section 5.7.2). Components: (A) Discrepancy Dendrogram (Section 5.5, Section 5.6.1), (B) Gallery (Section 5.6.2), (C) Subset Sensitivity View (Section 5.6.3), (D) Shepard Matrix (Section 5.6.4), (E) tooltip.

5.6.3 Subset Sensitivity View (T4)

The Gallery shows data cases of a selected cluster in a grid (Figure 5.5-B). The number of columns and their width can be selected by the analyst, as can the sort order of data cases and which parameter or output they should show. It is possible, e.g., to sort parameter visualizations by output similarity, as is often done in visual parameter space analysis [Lub+14; EST20]. We obtain the sort order by a 1D multidimensional scaling projection. Thus, the Gallery can show complex patterns.

5.6.4 Shepard Matrix (T1)

We want to give analysts a way to judge which parameter-output relations to investigate (T1). To this end, we use a Shepard diagram [de 05] showing all pairwise distances of data cases in a scatterplot. Each axis is the distance according to one measure. A diagonal line in a Shepard diagram thus means a perfect correspondence between two distance measures, and a dispersed Shepard scatterplot may be more interesting to investigate. We use the same color hue as in the Discrepancy Dendrogram for dots in a Shepard diagram, i.e., the further away from the diagonal, the more color hue is used. As the dataset usually has more than two parameters/outputs, we adapt the scatterplot matrix to Shepard diagrams to show all possible combinations (Figure 5.5-D).

5.7 Evaluation

We evaluated our visualizations heuristically and with expert interviews. The TU Wien pilot ethics board assessed our methods. Thus, our research adheres to the highest ethical standards. Specifically, our research questions were:

- (RQ1) Does our visualization design allow efficient and effective SA for SBSS parameters/outputs?
- (RQ2) Is our designed guidance effective?
- (RQ3) Does our visualization design transfer to other contexts than SBSS?

For RQ1 and RQ2, we conducted a heuristic evaluation with five visualization experts (Section 5.7.1). Two SBSS experts used our visualizations on their own data (Section 5.7.2), which also informs RQ1 and RQ2. Finally, for RQ3, we discussed visualizations with a microclimate simulation expert using an appropriate dataset. In this section we use two-letter shortcuts for people: Just letters indicate authors (e.g., NP) and a trailing number refers to participants (e.g., ME1).

Procedure. All sessions started with a 30 minutes introduction where we explained our problem context and the visualizations independently from the available datasets in the prototype. The slides are available in the supplemental material in the appendix. After the introduction, visualization experts continued with the questionnaire. The other experts used the prototype on a dataset and parameter settings they were familiar with. A semi-structured interview followed for all participants.

5.7.1 Visualization Experts (RQ1, RQ2)

We evaluated our visualization design heuristically with visualization experts according to the ICE-T method [Wal+19]. While a good design does not imply that the visualizations are effective, we think the inverse most likely holds (bad design → ineffective). Our chosen method is a good compromise between insights gained and the time requested from participants. We asked five participants (four Ph.D. students and one post-doc) from various universities to join our evaluation. We mostly met them over Zoom, and the sessions took around one hour each. According to ICE-T guidelines, five people are sufficient. Participants were free to use the prototype with various datasets on their own computers. They could always return to the visualization while filling out the ICE-T questionnaire. ICE-T responses are on a 7-point Likert scale. We asked them to share their thought process to understand their critique better.

Table 5.1 holds the results of these questionnaires, split by ICE-T component. The complete responses are available as supplemental material in the appendix. Wall et al. [Wal+19] state that a visualization design is successful when the mean score exceeds

Component	Mean		Std.dev
Insight	6.26	1 ————●— 7	1.06
Confidence	5.11	1 ———●—— 7	2.03
Essence	5.32	1 ———●—— 7	1.45
Time	6.08	1 ————●— 7	0.91
Total	**5.83**	1 ———|●— 7	**1.50**

Table 5.1: Results of the ICE-T evaluation with visualization experts. Responses were on a 7-point Likert scale. A total mean greater than five (small bar) is considered a success.

five, which we clearly achieved with an overall mean of 5.83. Our visualization's worst-scoring component (mean 5.11) is Confidence, which is also the one with the highest standard deviation. While participants agreed that we use "meaningful and accurate visual encodings" (question Q18 in the ICE-T questionnaire) and "avoid misleading representations" (Q19), they mostly disagreed that our visualization "promotes understanding beyond individual data cases" (Q20) or highlights data quality issues (Q21). It would take some effort to detect duplicate or invalid data cases in our visualization, but that was a conscious design choice. The second-worst component is Essence, which also has the second-highest standard deviation, indicating disagreement between participants. In fact, the two most contested questions here were whether the visualization "facilitates generalizations and extrapolations" (Q16) or "helps understand how variables relate in order to accomplish different analytic tasks" (Q17). Low ratings in the former were, e.g., because the Discrepancy Dendrogram assesses individual clusters but does not indicate differences between elements. This issue could be tackled in the future by specially crafted comparison visualizations. In the latter question, some participants focused on the "*different* analytic tasks" and argued that our visualization does not fulfill this criterion due to its singular focus.

On the other hand, participants rated the Insight and Time components very well. Two questions of the former seemed somewhat controversial, as they are associated with higher standard deviations (1.79 and 1.64). One participant *somewhat disagreed* that the visualization "facilitates perceiving relationships in the data" (Q2). Their reasoning was as follows. We show data cases as leaves in the Discrepancy Dendrogram and also in a gallery to the side. However, all data cases are separate visualizations, so it would be akin to showing individual bars instead of a histogram. However, they also realized that this was not a goal of our visualization design. The other contested question was whether the visualization "helps identify unusual or unexpected, yet valid, data characteristics" (Q5). One participant *somewhat disagreed,* mentioning that data cases with unusual or unexpected features would be hard to spot if the distance metrics would not consider these. We do not see this as an issue because the chosen dissimilarity metrics might as well measure local differences.

5.7.2 SBSS (RQ1, RQ2)

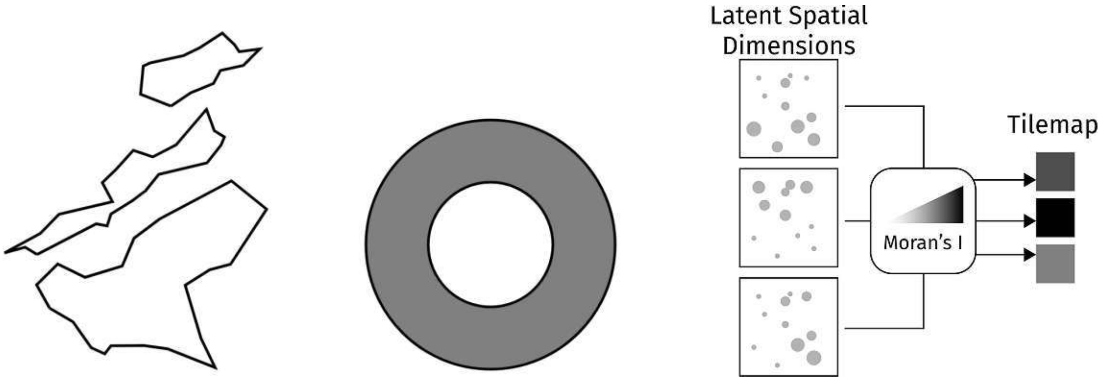
Two experts (SE1 and SE2) in statistics and SBSS, who were not part of the design process, used our visualizations on familiar datasets. They were recruited from the authors' professional network as they were required to have knowledge of SBSS. They both hold a Ph.D. in statistics and published on spatial data analysis. Sessions took around 2 hours. We guided them in the process as much as necessary, e.g., formulated possible analysis goals and answered any questions they had. After that, we continued with a semi-structured interview, inquiring about their confidence in findings, possible insights, and how these relate to prior expectations.

Datasets and Parameter Settings. The experts used two spatial datasets. SE2 worked on the *Colorado* dataset, which is a geochemical survey of 960 locations and 27 variables in Colorado, USA. Both SE2 and NP contributed parameter settings to investigate, as was agreed upon prior to the interview. SE2 provided an R script to obtain regionalizations (10 slices along four directions) and kernels (0–200 km radii). NP added regionalizations obtained in a prior study [Pic+22b]. SE1, on the other hand, worked on the meteorological *Veneto* dataset, which consists of 72 locations and 7 variables in Veneto, Italy. Parameter settings were obtained in a pilot session by SE1 and NP together using an existing prototype [Pic+22b]. We computed outputs for a full factorial of selected regionalizations and kernels for both datasets. In total, 42 settings were available for the *Veneto* and 48 for the *Colorado* dataset.

Dissimilarity Measures. We chose appropriate functions together with our collaborators. For the unmixing matrix W, we use the MD-Index [Ilm+10], a specialized comparison tool for unmixing matrices. For two kernels (K), we compute the difference of their so-called Spatial Kernel Matrix [MBN22]. We compare two regionalizations (R) by counting location pairs for which the region assignment is not identical.

Leaf Visualizations. We used three visualizations to represent R, K, and W (Figure 5.6). For R, we showed as multiple polygons representing the concave hull of regions (Figure 5.6a). In tooltips, these were integrated into interactive Leaflet maps. For K, we showed concentric circles representing the ring size (Figure 5.6b), also overlaying them to the spatial context with Leaflet in tooltips (Figure 5.5-E). We visualized W as a tilemap where each tile represented one latent dimension (Figure 5.6c). Tiles were colored in a univariate continuous gray color map showing Moran's I [Mor50], a measure for spatial autocorrelation. High values of that measure point to large-scale spatial patterns, which analysts might find easier to interpret. Tiles were ordered as the SBSS algorithm returned respective dimensions. Tooltips of tiles showed static plots of latent dimensions overlayed on OpenStreetMap.

SE1. NP guided SE1 to focus on SA because other than SE2, SE1 initially focused more on the spatial relationship between regionalizations (R) and locations in the dataset.



(a) R (b) K (c) W

Figure 5.6: Leaf visualizations for SBSS regionalization (R) and kernel (K) parameter, and output (W).

Regarding SA, SE1 was interested in the influence of the kernel (K) parameter on the output. NP pointed SE1 to a ●KW dendrogram configuration and explained that the red color points to sensitive parameter settings. Almost all K clusters were colored red. As they were wider in W, it indicated that the other parameter (R) exerts more influence on the output than K. SE1 switched to average linkage to account for any outliers that may skew the complete linkage criterion. Using this view (Figure 5.7), they found that K with a radius 0–60 km was the least red compared to others. Hence, this setting was most stable regarding the choice of R, with K=0–30 km a close second. SE1 explained that most locations in the dataset are within 75 km, so a kernel up to 60 km will likely capture most of the spatial dependency structure. SE1 also observed kernels up to 90 km radius (three big circles on the left in Figure 5.7) generally showing wider clusters in W than the smaller kernels due to their stronger red color. SE1 concluded that two levels of spatial variability exist in the dataset.

Next, a ●RW configuration of the Discrepancy Dendrogram, was investigated (Figure 5.8). Here, clusters of the 3-partitions chosen by altitude and precipitation were the most stable, meaning they were more independent of the choice of K than other partitions. This fact was initially surprising to SE1. However, SE1 reconciled it such that the two partitions are similar in that they both separate Veneto's mountainous and flat region. However, another separation in the plane seemed necessary. 2-partitions with just the mountain-flat separation were linked to wider clusters in the output, thus more sensitive to the choice of K.

In the interview, SE1 voiced many positive sentiments. They found the visualization "not difficult" to understand, and the construction of the Discrepancy Dendrogram was logical and easy to follow. SE1 liked the interactive maps and that "you can analyze the data by looking at different aspects in different ways." "Half of the work is made [with this tool]," so analysis time is saved compared to the "classical methods." In sum, SE1 found our visualizations "help evaluate the parameters" and identified an interesting

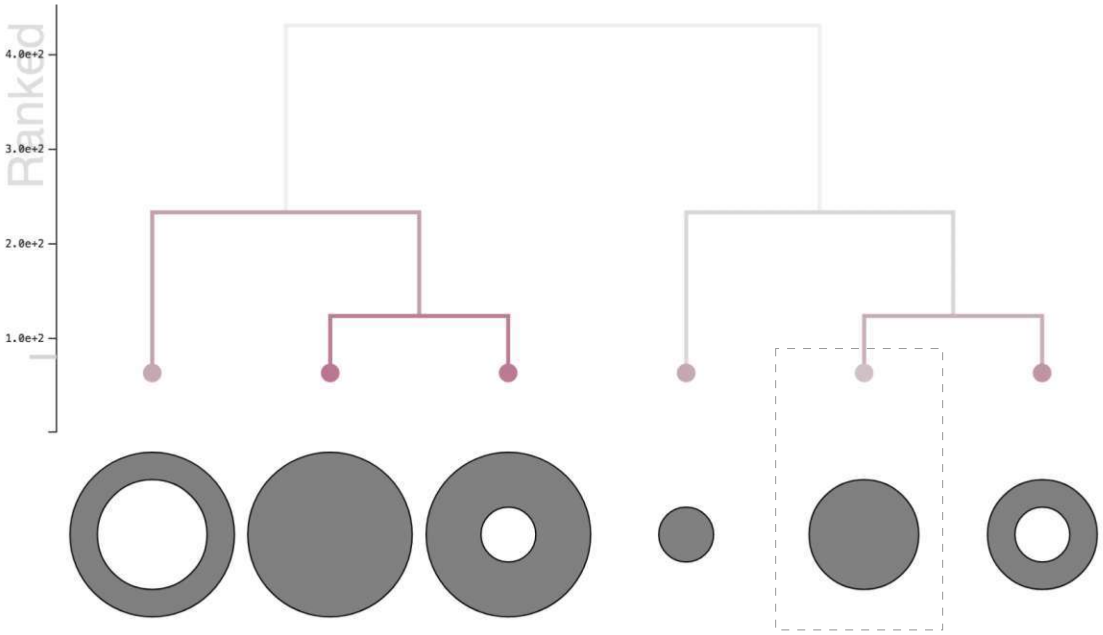


Figure 5.7: ●KW Discrepancy Dendrogram (average linkage with some clusters collapsed, cropped) for the *Veneto* dataset. The dashed box marks the most stable kernel setting identified by SE1.

parameter subspace to consider for future analysis: Smaller K in higher resolutions, as 0–60 km kernels were found to be most stable. SE1 could see our visualizations working for people who are "not completely expert [sic]" in SA. Based on these sentiments, we think RQ1 and RQ2 can be answered positively.

SE1 thought that the Discrepancy Dendrogram is not very easy to interpret but also attributed this to lack of familiarity with our approach and visualizations. Other than SE2, SE1 did not confirm or challenge expectations about parameter importance/sensitivity, as they find it necessary to compare multiple datasets before concluding anything. In the same spirit, SE1 remarked that a proper data analysis pipeline uses multiple complementing methods, prohibiting sweeping conclusions using our visualizations alone.

SE2. First, SE2 focused on a ●WK Discrepancy Dendrogram. SE2 observed many red lines and asked if it was correct to conclude that those outputs are less sensitive to kernel (K) choice, which it was. SE2 was then interested in regionalizations (R) and switched to ●WR. There, SE2 observed a very salient pattern (Figure 5.9): Most of the dendrogram was gray, indicating that cluster diameters match well between W and R. Thus, R is an important parameter for the *Colorado* dataset. A few clusters showed blue highlights, indicating clusters of sensitive R parameter settings. SE2 looked at one of the clusters (red arrow in Figure 5.9), saw the same R combined with various K, and considered the local dendrogram shape. SE2 concluded that two groups of W exist for

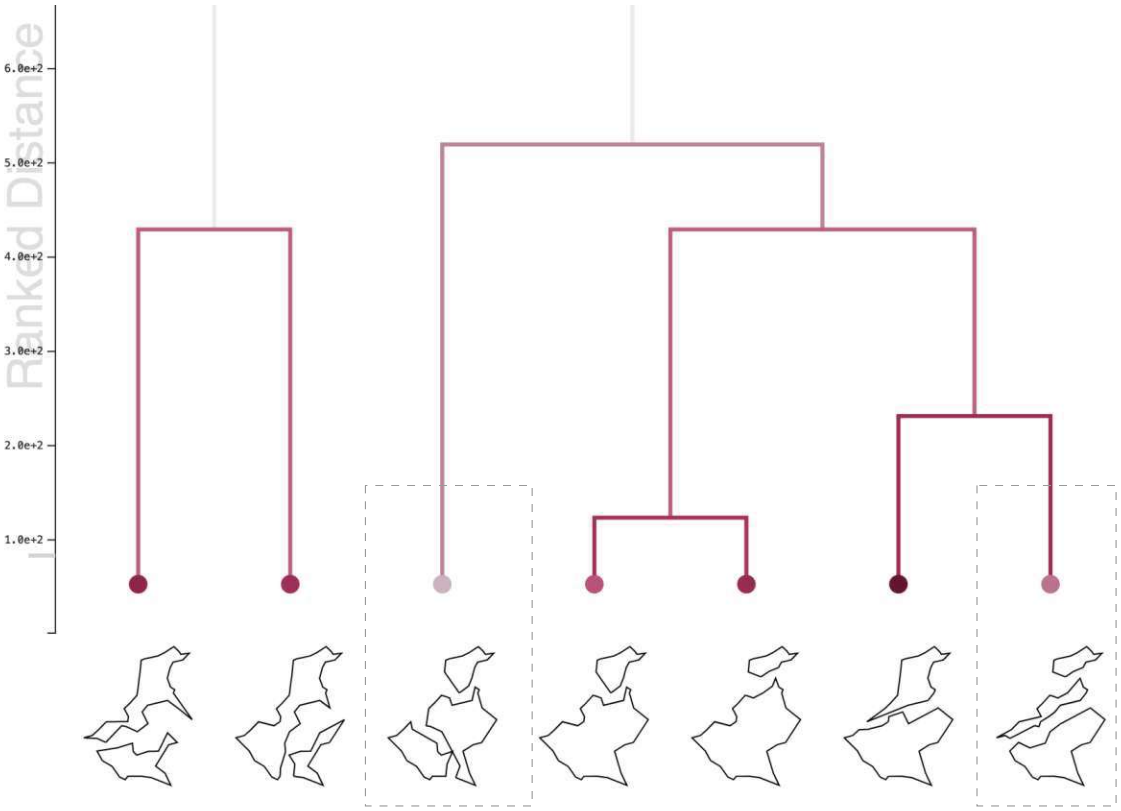


Figure 5.8: ●RW Discrepancy Dendrogram (complete linkage with some clusters collapsed, cropped) for the *Veneto* dataset. Dashed boxes mark the most stable regionalization settings identified by SE1.

this R setting (10 horizontal slices): One using very "un-local" kernels (K) with a 100 km hole and another group containing the dataset's remaining K settings. Hence, the choice of K matters a lot for this particular R setting. Other salient blue patterns were visible on the dendrogram's right side but not investigated by SE2. SE2 then returned to the ●WK configuration, but set the leaves to show R and investigated how these parameter settings were distributed in the dendrogram. They observed mostly neat clusters (by SBSS output W) of 6 data cases and identical R in each cluster, which was another hint that R is the more important parameter.

NP suggested looking at a parameter-focused dendrogram, after which SE2 changed it to ●KW. Here SE2 suggested that one K setting (0–50 km radius) is much more stable than the others due to its lighter color and wrongly concluded that R choice matters less for that. While their first assessment (more stable than others) was correct, the second part did not consider the magnitude of the cluster diameter difference in W. If SE2 would have used min-max normalized distances Discrepancy Dendrogram (Figure 5.10), they would have seen that also for that K, the cluster diameter difference in W was very high in absolute terms. Finally, SE2 also considered ●RW to investigate the stability

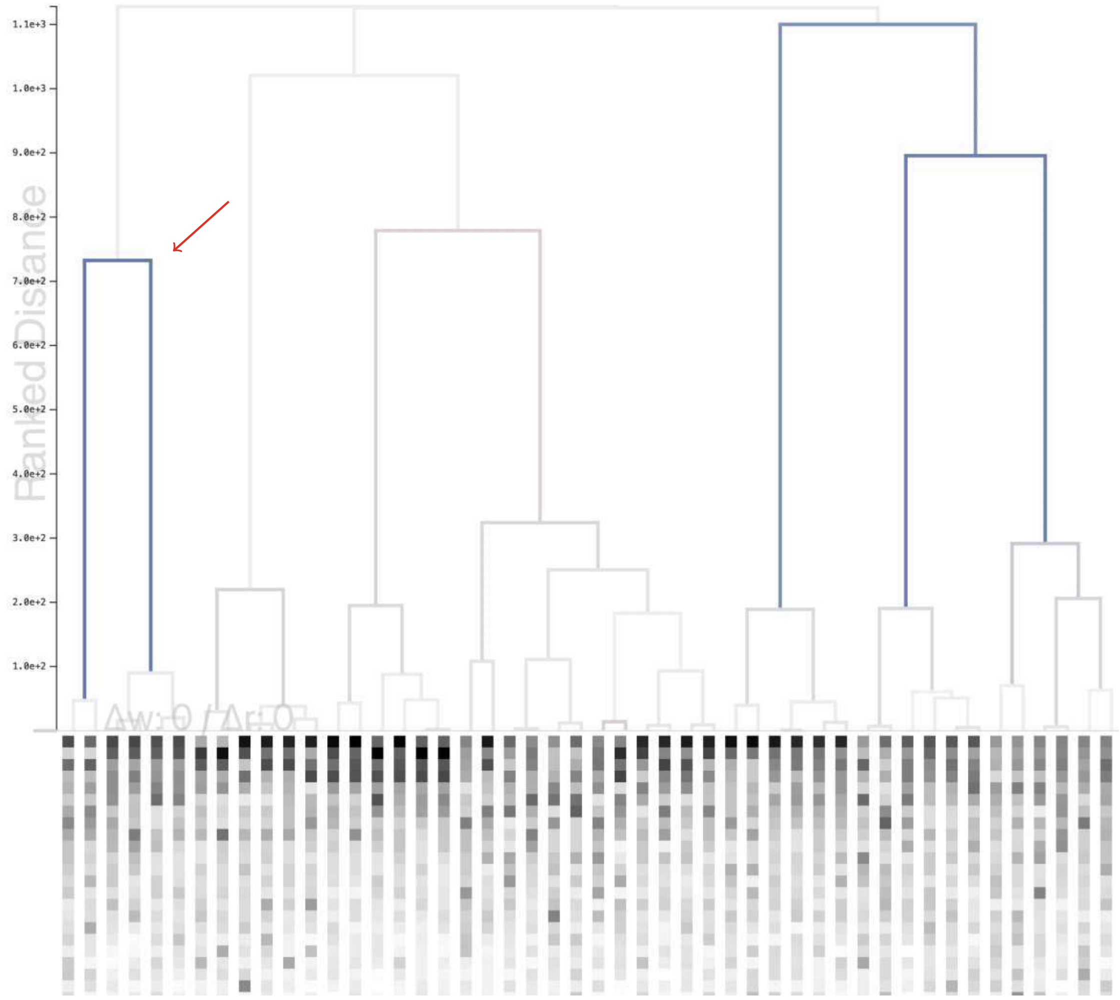
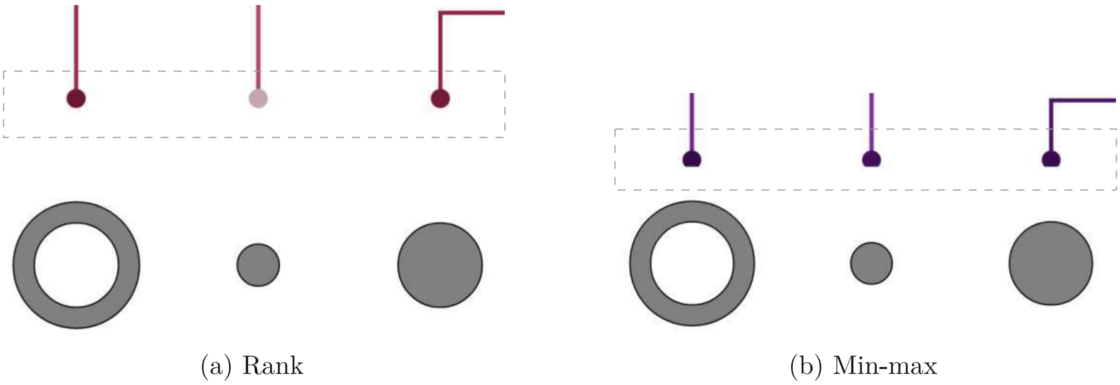


Figure 5.9: ●WR Discrepancy Dendrogram for the *Colorado* dataset. Blue lines mark data cases with variation in W despite similar R. Closer inspection revealed that the presence of a hole of at least 100 km size in associated K settings distinguishes these cases (red arrow).

of R. Here, a completely different picture than for ●KW emerged: The lines touching dendrogram leaves were gray instead of red, thus suggesting that less variation happens within R settings than between them. This image was consistent with ●WR, underlining the importance of the regionalization (R) parameter even more.

In the interview afterward, SE2 offered mainly positive comments. The visualization is "very intuitive to use", and it "speeds up analysis because one can see all parameter combinations at once." It "does exactly what it's supposed to do" because "the color pointed [them] to [data cases] where there was something going on" and, therefore, SE2 is "very confident in results obtained with this tool, I don't doubt it." They mentioned



(a) Rank

(b) Min-max

Figure 5.10: Rank and min-max distance normalization highlight different relations. Note the enclosed circles' color. With ranked distance (a), the small kernel is shown as most stable of the three (lighter color). Min-max distances (b) show that the absolute difference is low (saturated colors). Our visualizations offer more precise visual encodings in addition to color hue for such comparisons (see, e.g., Section 5.6.1).

that "many observations would not have been possible without this [visualization]" and that, therefore, it can be a qualitative complementary to the quantitative methods they use in their research. E.g., as a more systematic replacement for the trial & error they do now. SE2 confirmed their suspicion that R is the more important parameter using our visualizations. Again, we think these sentiments strongly support both RQ1 and RQ2.

SE2 also mentioned that the Discrepancy Dendrogram was not particularly easy to understand. I.e., the syntax, so to say, was clear (red, gray, and blue pointing in the direction of wider clusters), but translating that into actionable steps in parameter analysis was difficult. SE2 expects that this effect will get smaller with more familiarity with the visualizations. Finally, SE2 admitted they mostly looked at the tilemap in the leaves to judge W similarity. Since tiles contain a summary (Moran's I) of actual maps, the visualization may be misleading. A possible remedy could be a glyph design incorporating a derived feature and map similarity.

5.7.3 Microclimate Simulations (RQ3)

To demonstrate that the approach used in our prototype is transferable to other problem contexts (the goal of design studies [SMM12]), we applied it to microclimate simulation results [Vuc+22]. Such simulation models predict meteorological variables (e.g., air temperature or humidity) in a very small area, typically for a single street or a building. Microclimate simulations are critical nowadays as the climate crisis pressures cities and real estate developers to adapt to changing climate conditions. Usually, stakeholders, like city planners and architects, use existing simulation models and do not develop them themselves. Hence, parameter space analysis so far was mostly done by studying derived features (e.g., maximum temperature) with respect to grid size, often carried out with visual inspection, and computing relations (e.g., correlation) between individual variables.

Analysts have certain expectations about parameter relations. These come partially from known model limitations (e.g., the model does not perform well in extreme conditions) and partially from the modeled physical reality (e.g., the humidity of cold vs. hot air or wind chill effects).

In the conducted session, two authors (NP, JS) of this paper met with a microclimate simulation expert ME1, who has a Ph.D. in civil engineering and was recruited from the authors' professional network. JS controlled the prototype and suggested findings that ME1 assessed, while NP took notes.

Dataset and Parameter Settings. The experts' use case was to analyze the climatic conditions around a potential building (available as a 3D model) in several cities, seasons and meteorological conditions (called a *scenario*) to find the best location. The tested cities were Vienna, Helsinki, and Gothenburg in various seasons. ME1 computed a dataset containing 12 parameter settings and respective outputs. The low number of data cases follows the simulation model's computational demands as a single run takes several minutes to a couple of hours. The four outputs were wind speed (O_W), temperature on the surface (O_S) and in the air (O_A), and humidity (O_Q) at 6 am after a simulated interval of 24 h. The output values are spatially distributed on a grid. Parameters of the model were air temperature (P_A) and humidity (P_Q) as time series over 24 h, and wind speed and direction (P_W). We agreed to use Euclidean distance to measure similarity.

Leaf Visualizations. Three visualizations were used to show the model's parameters and outputs, both as leaf and tooltip visualizations. For the spatially distributed outputs (O_W, O_S, O_A, O_Q), we used heatmaps with univariate color scales of varying hue. Time series (P_A, P_Q) were shown as line charts. Wind speed and direction were shown as arrows, with speed as length and direction as rotation.

ME1. In the beginning, we asked ME1 about the most important output in the dataset, which ME1 answered to be the surface temperature (O_S). The goal was to identify a scenario where O_S is both low and stable so as to not be a threat to the human circulatory system. At the same time, general parameter-output relations were of interest. To achieve these tasks, JS set up a Discrepancy Dendrogram with O_S as primary distance and cycled through parameters as alternative distance. We started with a ● $\underline{O_S} P_W$ configuration, i.e., compared surface temperature output to the wind (direction and speed) parameter. The dendrogram showed many red lines, indicating wider clusters in P_W and thus generally no strong association between P_W and O_S. JS changed from ranked to min-max distances to see if the pattern persists when the magnitude is considered, which it did. This relation was expected for ME1. We also observed an O_S outlier with temperatures up to 36 °C, which seemed unexpected (red arrow in Figure 5.11). ME1 recalled that "the simulation model in question aims to capture extreme conditions in summer, like overheating, and there is really the question of how it performs in other conditions and different climates." ME1 concluded that the outlier might be a failure case of the model. Later analysis showed that the presumed model failure was related to extreme temperatures in the P_A

parameter. However, it became clear that wind alone “does not really make a difference” when it comes to surface temperature.

JS then switched to other parameters. Air temperature (P_A) was strongly correlated, as expected (Figure 5.12). A similar picture emerged for humidity (P_Q), except for a group of three scenarios (Figure 5.11-A) that arrived at similar O_S with significantly varying P_Q settings. ME1 noted that to determine the actual impact of P_Q here, one has to account for the different seasons and cities. This observation was noted as something to investigate later, as, at the time, season and city were not displayed in the prototype. JS then proceeded to compare other outputs with parameters. Our visualizations showed, and ME1 confirmed, the known relationship between humidity and air temperature. The next interesting observation came from the connection between wind and temperature. Wind parameter (P_W) and output (O_W) were not strongly correlated, and air temperature (P_A) was identified as another relevant factor (red arrow in Figure 5.12a). Regarding how temperature could influence wind, ME1 mentioned horizontal and vertical mixing effects but that those would be smaller than the wind-to-temperature effects. ME1 speculated that some correlations might come from the used 3D grid slices being on pedestrian level (1.8 m) while surface temperature is only valid for the slice at 0 m.

Asked about disadvantages or improvements, ME1 mentioned not picking a winning location for their use case because the city and season were missing in the visualization. NP checked together with JS later. Of all the surface temperature (O_S) clusters (A–C in Figure 5.11), O_S was least sensitive to air temperature (P_A) in the three scenarios enclosed by A. They belonged to Helsinki (2/3) and Vienna (1). Thus, Helsinki could be identified as the most suitable choice due to the more constant surface temperature. This choice is also consistent with the latest report of the Intergovernmental Panel on Climate Change [Gut+21], which predicts more stable mean temperature for Northern than Central Europe.

To summarize, we could apply our visualizations in a domain they were not originally designed for in the following way. We could find a suitable location for the building, which was the main goal for ME1, thus solving this domain’s SA task. ME1 could reconcile visualization images with domain knowledge and find interesting relations to investigate in the future, like the humidity parameter’s impact. We see this session as evidence to support RQ3, that our visualizations can be transferred to other contexts.

5.8 Limitations

As we rely on cluster diameters, the particular choice of nested partitions will greatly influence our sensitivity index, visualization image and, ultimately, the analysis outcome. The partitions are in turn influenced by the dataset, dissimilarity measure, clustering algorithm, and its parameters. We took care to select reasonable defaults, but they may not work for every situation. While it may be a demanding task, truthful clusterings can be obtained (cf. Section 5.3.3) and the particular groupings could be modifiable by the analyst. Another consequence of relying on relative cluster diameter differences for

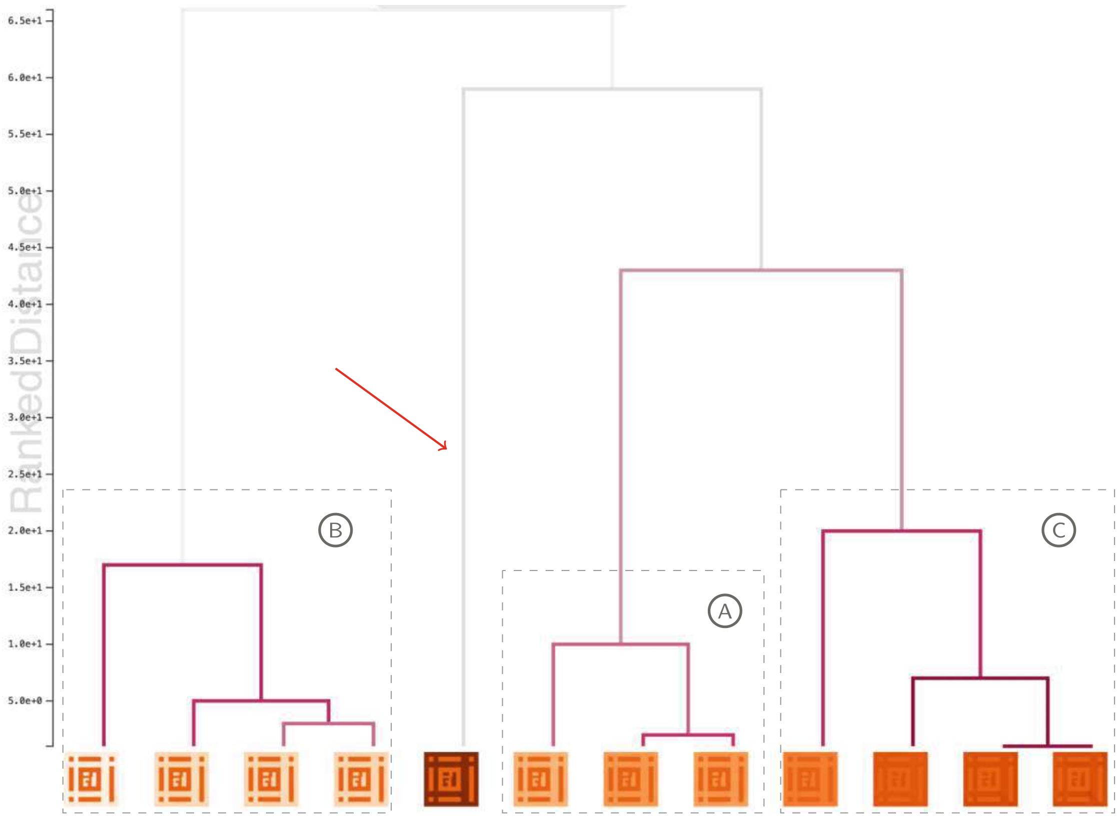
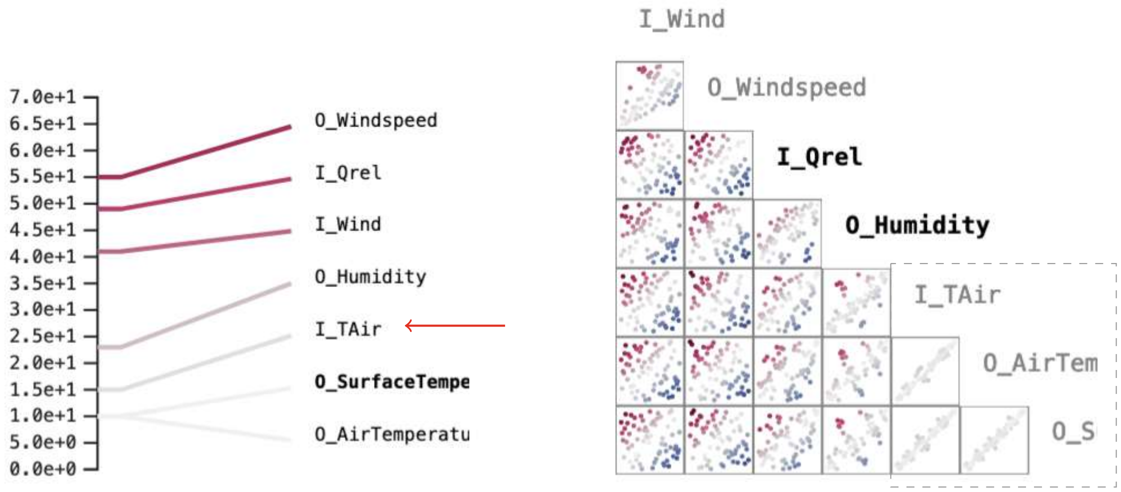


Figure 5.11: ●$\underline{O_S}P_W$ Discrepancy Dendrogram used for microclimate simulations. Red lines indicate wider clusters in P_W and thus little influence of that parameter on O_S. The red arrow marks a data case suspected to be a model failure. Data cases enclosed by A were also investigated with a ●$\underline{O_S}P_Q$ configuration. Data cases enclosed by A–C were considered for the final location choice.

SA is that the sensitivity index likely changes when new data is considered, thus the visualization image may be unstable with regard to additions to the underlying dataset. While that may seem like a big constraint, we argue that the same is true for visual SA of multivariate parameters: If they are sampled too coarsely or in too narrow intervals, then the analysis outcome may change a lot when the previously excluded parameter space is considered.

Our approach (Section 5.5.4) roughly corresponds to a one-at-a-time sensitivity index, i.e., a local method. Saltelli et al. [Sal+19] argue that local methods are only appropriate when the model under investigation is demonstrably linear. We did not confirm whether SBSS (Section 5.7.2) or the microclimate simulations (Section 5.7.3) are linear models. However, we do not see this as an issue for two reasons. First, local indices in the SA literature make precise quantitative statements for the whole parameter. As we defined our index only for subsets of data cases, it does not do that. Second, we developed



(a) Subset Sensitivity View (Section 5.6.3) of data cases in Figure 5.11-A.

(b) Shepard Matrix (Section 5.6.4) of microclimate dataset.

Figure 5.12: Subset Sensitivity View (a) of cluster in Figure 5.11 shows that air temperature P_A is be the driving parameter for surface temperature O_S, as expected. Shepard diagrams of air-related parameters/outputs in Shepard Matrix (b) show that this relation holds for all data cases.

the index for visual guidance in an interactive visualization. As all relevant data cases are visible in detail at any time, the analyst may consider much more context and existing domain knowledge than they would when interpreting only a single number, as demonstrated in Section 5.7.

5.9 Discussion and Conclusion

Based on requirements and observations in the context of SBSS, we developed a data type agnostic approach to visual SA. It only requires dissimilarity measures and thus works for complex parameters and outputs alike. The core innovation is measuring variation in parameter settings and outputs by cluster diameters. SA then becomes possible by looking at the difference of the same cluster's diameter in parameter and output space. Evaluation participants expressed high confidence in our visualizations. Future work may improve this paper's proposal by accounting for noise or simultaneously supporting multiple parameters.

The Discrepancy Dendrogram and supporting visualizations (Section 5.6) were also received very well by evaluation participants, especially considering the task complexity and short training time (around 30 minutes). The construction of the Discrepancy Dendrogram was logical for all participants, and the prototype provided sufficient interactions and levels of detail. The successful heuristic evaluation (Section 5.7.1) further supports this evidence. SBSS and microclimate simulation experts could confirm suspected or expected parameter-output relations with our visualizations, while mentioning the need to familiarize themselves more with our approach. E.g., the regionalization parameter R is more important for SBSS than the kernel configuration K (suspected by SE2), or that surface temperature mainly depends on air temperature (expected by ME1). Further, they could make high-level decisions (building location, ME1), find new relevant parameter subspaces (smaller kernels, SE1), or just obtain interesting observations (kernels with holes, SE1 and SE2). Considering the utility of the Discrepancy Dendrogram it will also be interesting to apply our approach to other visualization idioms, e.g., to DR scatterplots (Section 5.3.3).

We noted, e.g., during introductory explanations, that some participants found it mentally demanding to reason simultaneously about 1) groups of elements instead of single elements and 2) two distances within a group of elements. This issue is, to some extent, inherent to the problem we want to solve. On the other hand, we think rephrasing SA or finding visual representations so that analysts can reason about single elements instead of groups has much simplification potential. Achieving this would allow even more powerful SA visualizations potentially applicable to many contexts (Section 5.7.3).

Acknowledgments

This work was funded by the Austrian Science Fund (FWF) under grant P31881-N32 and the Vienna Science and Technology Fund (WWTF) under grant [10.47379/ICT19047]. We sincerely thank our evaluation participants for their time and expertise.

5.10 Bibliography

[AE20] A. M. Abuzuraiq and H. Erhan. “The Many Faces of Similarity”. In: *Design in the Age of Humans - Proceedings of the 25th CAADRIA Conference*. Vol. 1. Hong Kong: Assiciation for Computer-Aided Architectural Design Research in Asia, 2020, pp. 485–494. ISBN: 978-988-789-174-1. DOI: 10.52842/conf.caadria.2020.1.485 (cit. on p. 175).

[Bac+20] F. Bachoc, M. G. Genton, K. Nordhausen, A. Ruiz-Gazen, and J. Virta. “Spatial Blind Source Separation”. In: *Biometrika* 107.3 (2020), pp. 627–646. DOI: 10.1093/biomet/asz079 (cit. on p. 172).

[BGJ01] Z. Bar-Joseph, D. K. Gifford, and T. S. Jaakkola. “Fast Optimal Leaf Ordering for Hierarchical Clustering”. In: *Bioinformatics* 17.Suppl 1 (2001), S22–s29. DOI: 10.1093/bioinformatics/17.suppl_1.S22 (cit. on p. 180).

[Beh+14] M. Beham, W. Herzner, M. E. Gröller, and J. Kehrer. "Cupid: Cluster-Based Exploration of Geometry Generators with Parallel Coordinates and Radial Trees". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1693–1702. DOI: 10.1109/tvcg.2014.2346626 (cit. on p. 175).

[Ber+13] S. Bergner, M. Sedlmair, T. Moller, S. N. Abdolyousefi, and A. Saad. "ParaGlide: Interactive Parameter Space Partitioning for Computer Simulations". In: *IEEE Transactions on Visualization and Computer Graphics* 19.9 (2013), pp. 1499–1512. DOI: 10.1109/tvcg.2013.61 (cit. on p. 175).

[BDB15] R. Blanch, R. Dautriche, and G. Bisson. "Dendrogramix: A Hybrid Tree-Matrix Visualization Technique to Support Interactive Exploration of Dendrograms". In: *IEEE Pacific Visualization Symposium*. New York: IEEE, 2015, pp. 31–38. ISBN: 978-1-4673-6879-7. DOI: 10.1109/pacificvis.2015.7156353 (cit. on p. 176).

[BP16] E. Borgonovo and E. Plischke. "Sensitivity Analysis: A Review of Recent Advances". In: *European Journal of Operational Research* 248.3 (2016), pp. 869–887. DOI: 10.1016/j.ejor.2015.06.032 (cit. on p. 175).

[BM10] S. Bruckner and T. Möller. "Result-Driven Exploration of Simulation Parameter Spaces for Visual Effects Design". In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 1468–1476. DOI: 10.1109/tvcg.2010.190 (cit. on p. 175).

[CI04] D. G. Cacuci and M. Ionescu-Bujor. "A Comparative Review of Sensitivity and Uncertainty Analysis of Large-Scale Systems—II: Statistical Methods". In: *Nuclear Science and Engineering* 147.3 (2004), pp. 204–217. DOI: 10.13182/04-54cr (cit. on p. 175).

[CMS13] R. J. G. B. Campello, D. Moulavi, and J. Sander. "Density-Based Clustering Based on Hierarchical Density Estimates". In: *Advances in Knowledge Discovery and Data Mining*. Ed. by J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu. Vol. 7819. Berlin, Heidelberg: Springer, 2013, pp. 160–172. ISBN: 978-3-642-37456-2. DOI: 10.1007/978-3-642-37456-2_14 (cit. on p. 178).

[Cao+11] N. Cao, D. Gotz, J. Sun, and H. Qu. "DICON: Interactive Visual Analysis of Multidimensional Clusters". In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2581–2590. DOI: 10.1109/tvcg.2011.188 (cit. on p. 176).

[CD19] M. Cavallo and Ç. Demiralp. "Clustrophile 2: Guided Visual Clustering Analysis". In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 267–276. DOI: 10.1109/tvcg.2018.2864477 (cit. on p. 176).

[Cen+17] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H. Schulz, M. Streit, and C. Tominski. “Characterizing Guidance in Visual Analytics”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 111–120. DOI: 10.1109/tvcg.2016.2598468 (cit. on p. 178).

[Cib+23] L. Cibulski, T. May, J. Schmidt, and J. Kohlhammer. “COMPO*SED: Composite Parallel Coordinates for Co-Dependent Multi-Attribute Choices”. In: *IEEE Transactions on Visualization and Computer Graphics* (2023). in press. DOI: 10.1109/tvcg.2022.3180899 (cit. on p. 175).

[Cib+20] L. Cibulski, H. Mitterhofer, T. May, and J. Kohlhammer. “PAVED: Pareto Front Visualization for Engineering Design”. In: *Computer Graphics Forum* 39.3 (2020), pp. 405–416. DOI: 10.1111/cgf.13990 (cit. on p. 175).

[de 05] J. de Leeuw. “Shepard Diagram”. In: *Encyclopedia of Statistics in Behavioral Science*. Ed. by B. S. Everitt and D. C. Howell. Chichester: Wiley, 2005, bsa612. ISBN: 978-0-470-86080-9 978-0-470-01319-9. DOI: 10.1002/0470013192.bsa612 (cit. on p. 183).

[Des+19] R. Desai, F. Anderson, J. Matejka, S. Coros, J. McCann, G. Fitzmaurice, and T. Grossman. “Geppetto: Enabling Semantic Design of Expressive Robot Behaviors”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Glasgow Scotland Uk: ACM, 2019, pp. 1–14. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300599 (cit. on p. 176).

[Dor+15] H. Doraiswamy, N. Ferreira, M. Lage, H. Vo, L. Wilson, H. Werner, M. Park, and C. Silva. “Topology-Based Catalogue Exploration Framework for Identifying View-Enhanced Tower Designs”. In: *ACM Transactions on Graphics* 34.6 (2015), pp. 1–13. DOI: 10.1145/2816795.2818134 (cit. on p. 175).

[EST20] C. Eichner, H. Schumann, and C. Tominski. “Making Parameter Dependencies of Time-Series Segmentation Visually Understandable”. In: *Computer Graphics Forum* 39.1 (2020), pp. 607–622. DOI: 10.1111/cgf.13894 (cit. on pp. 175, 176, 182, 183).

[Gal15] T. Galili. “Dendextend: An R Package for Visualizing, Adjusting and Comparing Trees of Hierarchical Clustering”. In: *Bioinformatics* 31.22 (2015), pp. 3718–3720. DOI: 10.1093/bioinformatics/btv428 (cit. on p. 176).

[GT16] T. Günther and H. Theisel. “Inertial Steady 2D Vector Field Topology”. In: *Computer Graphics Forum* 35.2 (2016), pp. 455–466. DOI: 10.1111/cgf.12846 (cit. on p. 176).

[Guo+11] Z. Guo, M. O. Ward, E. A. Rundensteiner, and C. Ruiz. “Pointwise Local Pattern Exploration for Sensitivity Analysis”. In: *IEEE Conference on Visual Analytics Science and Technology (VAST)*. New York: IEEE, 2011, pp. 131–140. DOI: 10.1109/vast.2011.6102450 (cit. on p. 175).

[Gut+21] J. M. Gutiérrez, R. G. Jones, G. T. Narisma, L. M. Alves, M. Amjad, I. V. Gorodetskaya, M. Grose, N. A. B. Klutse, S. Krakovska, J. Li, D. Martínez-Castro, L. O. Mearns, et al. "Atlas". In: *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Ed. by V. Masson-Delmotte, P. Zhai, A. Pirani, S. L. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M. I. Gomis, M. Huang, K. Leitzell, et al. Cambridge, New York: Cambridge University Press, 2021, pp. 1927–2058. DOI: 10.1017/9781009157896.021 (cit. on p. 193).

[Ham94] D. M. Hamby. "A Review of Techniques for Parameter Sensitivity Analysis of Environmental Models". In: *Environmental Monitoring and Assessment* 32.2 (1994), pp. 135–154. DOI: 10.1007/bf00547132 (cit. on p. 175).

[Haz+20] S. Hazarika, H. Li, K.-C. Wang, H.-W. Shen, and C.-S. Chou. "NNVA: Neural Network Assisted Visual Analysis of Yeast Cell Polarization Simulation". In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2020), pp. 34–44. DOI: 10.1109/tvcg.2019.2934591 (cit. on pp. 175, 176).

[He+20] W. He, J. Wang, H. Guo, K.-C. Wang, H.-W. Shen, M. Raj, Y. S. G. Nashed, and T. Peterka. "InSituNet: Deep Image Synthesis for Parameter Space Exploration of Ensemble Simulations". In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2020), pp. 23–33. DOI: 10.1109/tvcg.2019.2934312 (cit. on p. 176).

[Ilm+10] P. Ilmonen, K. Nordhausen, H. Oja, and E. Ollila. "A New Performance Index for ICA: Properties, Computation and Asymptotic Analysis". In: *Latent Variable Analysis and Signal Separation*. Ed. by V. Vigneron, V. Zarzoso, E. Moreau, R. Gribonval, and E. Vincent. Berlin, Heidelberg: Springer, 2010, pp. 229–236. ISBN: 978-3-642-15995-4. DOI: 10.1007/978-3-642-15995-4_29 (cit. on p. 186).

[IC04] M. Ionescu-Bujor and D. G. Cacuci. "A Comparative Review of Sensitivity and Uncertainty Analysis of Large-Scale Systems—I: Deterministic Methods". In: *Nuclear Science and Engineering* 147.3 (2004), pp. 189–203. DOI: 10.13182/nse03-105cr (cit. on p. 175).

[IL15] B. Iooss and P. Lemaître. "A Review on Global Sensitivity Analysis Methods". In: *Uncertainty Management in Simulation-Optimization of Complex Systems: Algorithms and Applications*. Ed. by G. Dellino and C. Meloni. Vol. 59. Boston, MA: Springer US, 2015, pp. 101–122. ISBN: 978-1-4899-7547-8. DOI: 10.1007/978-1-4899-7547-8_5 (cit. on p. 175).

[JM00] T. Jankun-Kelly and K.-L. Ma. "A Spreadsheet Interface for Visualization Exploration". In: *Proceedings Visualization 2000. VIS 2000 (Cat. No.00CH37145)*. 2000, pp. 69–76. DOI: 10.1109/visual.2000.885678 (cit. on p. 175).

[Jeo+22] H. Jeon, H.-K. Ko, J. Jo, Y. Kim, and J. Seo. "Measuring and Explaining the Inter-Cluster Reliability of Multidimensional Projections". In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2022), pp. 551–561. doi: 10.1109/tvcg.2021.3114833 (cit. on p. 176).

[Kni+21] J. Knittel, A. Lalama, S. Koch, and T. Ertl. "Visual Neural Decomposition to Explain Multivariate Data Sets". In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (2021), pp. 1374–1384. doi: 10.1109/tvcg.2020.3030420 (cit. on p. 175).

[KSI14] Y. Koyama, D. Sakamoto, and T. Igarashi. "Crowd-Powered Parameter Analysis for Visual Design Exploration". In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. Honolulu Hawaii USA: ACM, 2014, pp. 65–74. isbn: 978-1-4503-3069-5. doi: 10.1145/2642918.2647386 (cit. on p. 176).

[Kwo+18] B. C. Kwon, B. Eysenbach, J. Verma, K. Ng, C. De Filippi, W. F. Stewart, and A. Perer. "Clustervision: Visual Supervision of Unsupervised Clustering". In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2018), pp. 142–151. doi: 10.1109/tvcg.2017.2745085 (cit. on p. 176).

[LYi+15] S. L'Yi, B. Ko, D. Shin, Y.-J. Cho, J. Lee, B. Kim, and J. Seo. "XCluSim: A Visual Analytics Tool for Interactively Comparing Multiple Clustering Results of Bioinformatics Data". In: *BMC Bioinformatics* 16.11 (2015), S5:1–s5:15. doi: 10.1186/1471-2105-16-s11-s5 (cit. on p. 176).

[Lig13] A. Ligmann-Zielinska. "Spatially-Explicit Sensitivity Analysis of an Agent-Based Model of Land Use Change". In: *International Journal of Geographical Information Science* 27.9 (2013), pp. 1764–1781. doi: 10.1080/13658816.2013.782613 (cit. on p. 175).

[LT09] L. Lilburne and S. Tarantola. "Sensitivity Analysis of Spatial Models". In: *International Journal of Geographical Information Science* 23.2 (2009), pp. 151–168. doi: 10.1080/13658810802094995 (cit. on p. 175).

[Liu+10] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu. "Understanding of Internal Clustering Validation Measures". In: *International Conference on Data Mining*. New York: IEEE, 2010, pp. 911–916. isbn: 978-1-4244-9131-5. doi: 10.1109/icdm.2010.35 (cit. on p. 179).

[Lub+15] M. Luboschik, M. Röhlig, A. T. Bittig, N. Andrienko, H. Schumann, and C. Tominski. "Feature-Driven Visual Analytics of Chaotic Parameter-Dependent Movement". In: *Computer Graphics Forum* 34.3 (2015), pp. 421–430. doi: 10.1111/cgf.12654 (cit. on p. 175).

[Lub+14] M. Luboschik, S. Rybacki, F. Haack, and H.-J. Schulz. "Supporting the Integrated Visual Analysis of Input Parameters and Simulation Trajectories". In: *Computers & Graphics* 39 (2014), pp. 37–47. doi: 10.1016/j.cag.2013.09.004 (cit. on pp. 175, 182, 183).

[Mar+97] J. Marks, P. Beardsley, B. Andalman, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, J. Seims, and S. Shieber. “Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation”. In: *Proceedings of SIGGRAPH*. New York: ACM, 1997, pp. 389–400. DOI: 10.1145/258734.258887 (cit. on p. 175).

[Mar+11] A. Marrel, B. Iooss, M. Jullien, B. Laurent, and E. Volkova. “Global Sensitivity Analysis for Models with Spatially Dependent Outputs”. In: *Environmetrics* 22.3 (2011), pp. 383–397. DOI: 10.1002/env.1071 (cit. on p. 175).

[Mat+17] K. Matković, H. Abraham, M. Jelović, and H. Hauser. “Quantitative Externalization of Visual Data Analysis Results Using Local Regression Models”. In: *1st International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Ed. by A. Holzinger, P. Kieseberg, A. M. Tjoa, and E. Weippl. Vol. 10410. Cham: Springer International Publishing, 2017, pp. 199–218. ISBN: 978-3-319-66807-9 978-3-319-66808-6. DOI: 10.1007/978-3-319-66808-6_14 (cit. on pp. 175, 176).

[Mor50] P. A. P. Moran. “Notes on Continuous Stochastic Phenomena”. In: *Biometrika* 37.1/2 (1950), pp. 17–23. DOI: 10.2307/2332142 (cit. on p. 186).

[MBN22] C. Muehlmann, F. Bachoc, and K. Nordhausen. “Blind Source Separation for Non-Stationary Random Fields”. In: *Spatial Statistics* 47 (2022), p. 100574. DOI: 10.1016/j.spasta.2021.100574 (cit. on pp. 172, 174, 186).

[Nor+15] K. Nordhausen, H. Oja, P. Filzmoser, and C. Reimann. “Blind Source Separation for Spatial Compositional Data”. In: *Mathematical Geosciences* 47.7 (2015), pp. 753–770. DOI: 10.1007/s11004-014-9559-5 (cit. on pp. 172, 174).

[Orb+19] D. Orban, D. F. Keefe, A. Biswas, J. Ahrens, and D. Rogers. “Drag and Track: A Direct Manipulation Interface for Contextualizing Data Instances within a Continuous Parameter Space”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 256–266. DOI: 10.1109/tvcg.2018.2865051 (cit. on p. 175).

[Paj+17] S. Pajer, M. Streit, T. Torsney-Weir, F. Spechtenhauser, T. Möller, and H. Piringer. “WeightLifter: Visual Weight Space Exploration for Multi-Criteria Decision Making”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 611–620. DOI: 10.1109/tvcg.2016.2598589 (cit. on p. 175).

[Pic+22a] N. Piccolotto, M. Bögl, T. Gschwandtner, C. Muehlmann, K. Nordhausen, P. Filzmoser, and S. Miksch. “TBSSvis: Visual Analytics for Temporal Blind Source Separation”. In: *Visual Informatics* 6.4 (2022), pp. 51–66. DOI: 10.1016/j.visinf.2022.10.002 (cit. on pp. 173, 175, 177, 179).

[PBM23] N. Piccolotto, M. Bögl, and S. Miksch. “Visual Parameter Space Exploration in Time and Space”. In: *Computer Graphics Forum* 42.6 (2023), e14785. DOI: 10.1111/cgf.14785 (cit. on pp. 173, 175, 177).

[Pic+22b] N. Piccolotto, M. Bögl, C. Muehlmann, K. Nordhausen, P. Filzmoser, and S. Miksch. “Visual Parameter Selection for Spatial Blind Source Separation”. In: *Computer Graphics Forum* 41.3 (2022), pp. 157–168. DOI: 10.1111/cgf14530 (cit. on pp. 172–174, 177, 178, 186).

[Rai+19] J. Raimbault, C. Cottineau, M. L. Texier, F. L. Néchet, and R. Reuillon. “Space Matters: Extending Sensitivity Analysis to Initial Spatial Conditions in Geosimulation Models”. In: *Journal of Artificial Societies and Social Simulation* 22.4 (2019), 10:1–10:22. DOI: 10.18564/jasss.4136 (cit. on p. 175).

[Roh82] F. J. Rohlf. “12 Single-link Clustering Algorithms”. In: *Classification Pattern Recognition and Reduction of Dimensionality*. Ed. by P. R. Krishnaiah and L. N. Kanal. Vol. 2. Elsevier, 1982, pp. 267–284. DOI: 10.1016/s0169-7161(82)02015-x (cit. on p. 178).

[RGG18] I. B. Rojo, M. Gross, and T. Günther. “Visualizing the Phase Space of Heterogeneous Inertial Particles in 2D Flows”. In: *Computer Graphics Forum* 37.3 (2018), pp. 289–300. DOI: 10.1111/cgf.13420 (cit. on p. 176).

[Sal02] A. Saltelli. “Sensitivity Analysis for Importance Assessment”. In: *Risk Analysis* 22.3 (2002), pp. 579–590. DOI: 10.1111/0272-4332.00040 (cit. on p. 175).

[Sal+19] A. Saltelli, K. Aleksankina, W. Becker, P. Fennell, F. Ferretti, N. Holst, S. Li, and Q. Wu. “Why so Many Published Sensitivity Analyses Are False: A Systematic Review of Sensitivity Analysis Practices”. In: *Environmental Modelling & Software* 114 (2019), pp. 29–39. DOI: 10.1016/j.envsoft.2019.01.012 (cit. on pp. 175, 194).

[Sal+07] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global Sensitivity Analysis: The Primer*. Chichester: Wiley, 2007. ISBN: 978-0-470-05997-5. DOI: 10.1002/9780470725184 (cit. on p. 175).

[Sch+17] A. Schulz, J. Xu, B. Zhu, C. Zheng, E. Grinspun, and W. Matusik. “Interactive Design Space Exploration and Optimization for CAD Models”. In: *ACM Transactions on Graphics* 36.4 (2017), pp. 1–14. DOI: 10.1145/3072959.3073688 (cit. on p. 176).

[SMM12] M. Sedlmair, M. Meyer, and T. Munzner. “Design Study Methodology: Reflections from the Trenches and the Stacks”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), pp. 2431–2440. DOI: 10.1109/tvcg.2012.213 (cit. on p. 191).

[Sed+14] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. “Visual Parameter Space Analysis: A Conceptual Framework”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2161–2170. DOI: 10.1109/tvcg.2014.2346321 (cit. on pp. 173, 175, 176).

[SS02] J. Seo and B. Shneiderman. “Interactively Exploring Hierarchical Clustering Results [Gene Identification]”. In: *Computer* 35.7 (2002), pp. 80–86. DOI: 10.1109/mc.2002.1016905 (cit. on p. 176).

[Sob90] I. M. Sobol’. “On Sensitivity Estimation for Nonlinear Mathematical Models”. In: *Matematicheskoe modelirovanie* 2.1 (1990), pp. 112–118 (cit. on p. 175).

[Ste+13] C. A. Steed, D. M. Ricciuto, G. Shipman, B. Smith, P. E. Thornton, D. Wang, X. Shi, and D. N. Williams. “Big Data Visual Analytics for Exploratory Earth System Simulation Analysis”. In: *Computers & Geosciences* 61 (2013), pp. 71–82. DOI: 10.1016/j.cageo.2013.07.025 (cit. on p. 175).

[Vuc+22] M. Vuckovic, J. Schmidt, T. Ortner, and D. Cornel. “Combining 2D and 3D Visualization with Visual Analytics in the Environmental Domain”. In: *Information* 13.1 (2022), 7:1–7:21. DOI: 10.3390/info13010007 (cit. on pp. 173, 191).

[Wal+19] E. Wall, M. Agnihotri, L. Matzen, K. Divis, M. Haass, A. Endert, and J. Stasko. “A Heuristic Approach to Value-Driven Evaluation of Visualizations”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 491–500. DOI: 10.1109/tvcg.2018.2865146 (cit. on p. 184).

[Wan+17] J. Wang, X. Liu, H.-W. Shen, and G. Lin. “Multi-Resolution Climate Ensemble Parameter Analysis with Nested Parallel Coordinates Plots”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 81–90. DOI: 10.1109/tvcg.2016.2598830 (cit. on p. 175).

[WVJ16] M. Wattenberg, F. Viégas, and I. Johnson. “How to Use T-SNE Effectively”. In: *Distill* 1.10 (2016), e2. DOI: 10.23915/distill.00002 (cit. on p. 176).

[Wei+16] J. Weissenböck, A. Amirkhanov, E. Gröller, J. Kastner, and C. Heinzl. “PorosityAnalyzer: Visual Analysis and Evaluation of Segmentation Pipelines to Determine the Porosity in Fiber-Reinforced Polymers”. In: *IEEE Conference on Visual Analytics Science and Technology*. New York: IEEE, 2016, pp. 101–110. DOI: 10.1109/vast.2016.7883516 (cit. on p. 176).

[WHA07] W. Willett, J. Heer, and M. Agrawala. “Scented Widgets: Improving Navigation Cues with Embedded Visualizations”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1129–1136. DOI: 10.1109/tvcg.2007.70589 (cit. on p. 176).

[Xia+22] J. Xia, Y. Zhang, J. Song, Y. Chen, Y. Wang, and S. Liu. "Revisiting Dimensionality Reduction Techniques for Visual Cluster Analysis: An Empirical Study". In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2022), pp. 529–539. doi: 10.1109/tvcg.2021.3114694 (cit. on p. 176).

[XT15] D. Xu and Y. Tian. "A Comprehensive Survey of Clustering Algorithms". In: *Annals of Data Science* 2.2 (2015), pp. 165–193. doi: 10.1007/s40745-015-0040-1 (cit. on p. 176).

[YBP21] H. Yang, R. Ballester-Ripoll, and R. Pajarola. "SenVis: Interactive Tensor-based Sensitivity Visualization". In: *Computer Graphics Forum* 40.3 (2021), pp. 275–286. doi: 10.1111/cgf.14306 (cit. on p. 175).

CHAPTER 6

Discussion

In this chapter we will discuss how our work advances the research questions formulated in Chapter 1. Section 6.1 focuses on the research questions, Section 6.2 on the overarching contributions, and Section 6.3 on limitations and gaps of our research in general. To repeat, these are our first-author publications that this thesis builds on:

P1 **Nikolaus Piccolotto**, Markus Bögl, Theresia Gschwandtner, Christoph Muehlmann, Klaus Nordhausen, Peter Filzmoser, Silvia Miksch: *TBSSvis: Visual Analytics for Temporal Blind Source Separation*. Visual Informatics, vol. 6, no. 4, 2022. DOI: 10.1016/j.visinf.2022.10.002. Chapter 3 of this thesis.

P2 **Nikolaus Piccolotto**, Markus Bögl, Christoph Muehlmann, Klaus Nordhausen, Peter Filzmoser, Silvia Miksch: *Visual Parameter Selection for Spatial Blind Source Separation*. Computer Graphics Forum, vol. 41, no. 3, 2022. DOI: 10.1111/cgf14530. Chapter 4 of this thesis.

P3 **Nikolaus Piccolotto**, Markus Bögl, Christoph Muehlmann, Klaus Nordhausen, Peter Filzmoser, Johanna Schmidt, Silvia Miksch: *Data Type Agnostic Visual Sensitivity Analysis*. IEEE Transactions on Visualization and Computer Graphics, vol. 30, no. 1, 2024. DOI: 10.1109/TVCG.2023.3327203. Chapter 5 of this thesis.

6.1 Research Questions

(RQ1) What are the characteristics of effective guidance for Temporal and Spatial BSS parameter selection? We tackled this research question in one way or another in all our publications. In publication P1, we tackled the question for one TBSS method (gSOBI), where the tuning parameter space included one weight $(0, 1)$ and two lag sets. More guidance was required for the latter than the former as the weight parameter describes a trade-off between the two methods at its extreme points (vSOBI and SOBI, respectively). From the involved equations and simulation studies it was already known that vSOBI dominates SOBI below a weight of around 0.9. However, the choice of lags depends on the dataset at hand. To this end, we proposed a multiple-view system consisting of a PCP to filter lags, a multivariate autocorrelation function plot to investigate the autocorrelation of input variables at a lag, a line chart and a scatterplot. The PCP allowed to filter lags by their correspondence to calendar granules, such as days, weeks or months. Several derived variables per lag were encoded in the PCP axes, such as the maximum absolute autocorrelation of any input variable. Analysts could thus approach the problem in smaller steps: First they select the calendar granule where they expect most autocorrelation, thus filtering lags down to those. Second, the PCP allows to further reduce the candidate lags via filtering to an amount of granules (e.g., 1–6 months) or to some range of the derived variables. The remaining visualizations (line graph, scatterplot) allowed to investigate individual lags in detail as a last step, if necessary. In the expert interviews, especially the interactivity and linked views were considered very useful. Although sometimes challenging in the beginning, the interactive visualizations allowed to dig through the available parameter space faster than it would have been possible in R. However, that alone was not enough for every participant, as one in particular would have preferred the strongest possible guidance degree. The reason for that is likely that the per-lag derived variables are only a proxy for what the guidance should really be based on, which are properties of the unobserved latent dimensions. In addition, the ability to compare competing parameter settings and associated outputs in detail was seen as very useful. We summarize how that was achieved as part of the next research question.

In publication P2, we considered guidance for SBSS parameter selection. In that non-stationary model (Section 1.1.3), the tuning parameters were a regionalization (a partition of locations) and a kernel (a neighborhood definition). As in the TBSS case, any guidance metrics should actually be computed on latent dimensions than input variables, as far as BSS theory goes, because a kernel should be selected such that it encapsulates the spatial dependence of different latent processes. As a non-stationary scenario requires the analyst to partition the spatial dataset into coherent regions, automatic regionalizations into a pre-specified number of regions (2–8) were considered extremely helpful. Especially so because geochemical surveys often consist of several hundred locations and dozens of variables, and thus manual regionalizations are time-consuming. Analysts still had to choose the number of regions. Whether a region was too small (especially in combination with a particular kernel choice) could inform that choice as well as how close the covariance of variables in a region is to the sample (i.e., non-spatial) covariance matrix. While not part of our design nor suggested by any study participant, region homogenity/heterogenity measures as used in various regionalization algorithms would have been another avenue towards that end. Similarly, analysts had to choose the number and extent of kernels. As the latent processes are a-priori unavailable, we again used the input variables as a proxy for them. To guide the kernel selection process, we employed superpositioned variograms. Variograms are an established visualization in spatial statistics, showing how much a variable changes with distance. Selecting kernels was thus reduced to visually segmenting the variograms. Again, although it was not part of our design nor suggested by study participants, we may have found suggestions for kernel sizes by repurposing a time series segmentation algorithm. In expert interviews, SBSS experts praised the regionalization guidance and wished for something similar for the kernel parameter. A geochemist confirmed boundaries of the regionalization guidance for one of the datasets we used and was impressed and excited by patterns found in latent dimensions.

In publication P3, we focused on sensitivity analysis for SBSS parameters. It relates to parameter selection guidance insofar as less effort needs to be put into selecting parameters that do not influence the output much. Another way how sensitivity analysis can influence parameter optimization is that often analysts will prefer the output obtained from a stable parameter range. As it was unclear to what extent numerical sensitivity analysis approaches are applicable to the spatial data found in SBSS parameters and outputs, we used cluster diameter differences to quantify stable (cluster is smaller/narrower in output space) and sensitive (cluster is smaller/narrower in parameter space) parameter settings. These diameter differences were encoded in a colored dendrogram and supported by other visualizations. The VA prototype was evaluated with two SBSS experts and one expert in microclimate simulations. They were able to identify stable/sensitive parameter ranges, more and less important parameters, and could make high-level decisions or find parameter ranges to investigate next.

To summarize, what were the characteristics of effective guidance? Based on P1–P3 we can provide another confirmation for thirty years of visualization research, that **interactivity** is a major factor; the ability to sift through large amounts of data "speeds up analysis because you see [everything] at once" [Pic+24, p. 8]. In P1 specifically, this could be achieved by rapidly changing between multiple levels of detail while in P3 we quite literally showed everything at once. **Comparing alternative parameter settings and their outputs** was also very helpful as it allowed to investigate the effect of a single lag or kernel, if one was inclined to do so (P1, P3). In P2, **sensible parameter settings suggestions** (regionalizations) were considered very helpful. For the kernel parameter, we reduced a tough abstract task (define number of kernels and their extent) to a **simpler task on a concrete visualization** (segment the variograms). Seeing the parameter settings in their **spatial context**, i.e., next to country borders and overlaid on satellite images, was also advantageous. Finally, the prototype suggested in P3 was as a whole intended to guide parameter selection, so **combining multiple PSA tasks** in order to support one is very likely beneficial.

(RQ2) Which VA methods can be utilized to explore ensembles of temporal and spatial BSS components? In publication P1, we employed several strategies to that end. We used DR scatterplots to show similarities of whole ensembles and individual components. To ensure that every data point in the scatterplot is visible and clickable, we removed overlaps by regularizing everything onto a grid. To show projection errors, mouse-over on a particular data point triggered a size change in the marks of other data points based on their similarity. The two lag set parameters were shown in separate DR scatterplots, so analysts could use both the output and the parameters to select a TBSS method. The other strategy was the *component overview*, which is based on a set-aware k-medoids clustering. Cannot-link constraints ensure the set-awareness by always grouping components of the same parametrization into separate clusters. Therefore, only components of different methods are grouped together and the resulting medoids can represent all existing components well (depending on the choice of k and available TBSS methods). To find differences between a couple of methods and their components, we suggested i) sorting components by a user-selected degree-of-interest function, ii) stacking components of a method vertically, and iii) connecting components of two methods by lines whose thickness encoded similarity (*time series slope graph*). Analysts could always zoom into specific time intervals. This visualization was deemed "easier [to use] than looking at a correlation matrix" [Pic+22, p. 63].

Publication P2 focused completely on the parameter selection. The developed VA prototype did not show components. For our evaluation we mapped them using R and `ggplot` [Wic16] and printed them on paper.

In publication P3, component ensembles are shown on a coarse level with a hierarchical clustering (according to the unmixing matrix, the regionalization or the kernel parameter) and concise leaf visualizations in the dendrogram. Specifically, the leaf visualization was a tilemap depicting each component's derived feature (Moran's I, a measure for spatial autocorrelation). Similar-looking tilemaps thus pointed to similar SBSS outputs, although an evaluation participant remarked that two ensembles with similar Moran's I but different spatial distributions of data will look the same. In addition to the tilemap, a detailed plot of a component was available as tooltip. Our work in P3 thus allowed to explore component ensembles through the similarity structures in SBSS parameters and outputs, similar in spirit to the DR scatterplots in P1.

Thus, **using the similarity structures found in SBSS parameters and outputs** (P1, P3) suggests VA methods to explore BSS component ensembles. For example, they can be visualized directly (time series slope graph) or they can be fed into appropriate data mining algorithms first (set-aware clustering, DR embedding). Also for this research question, the ability to **interactively** switch between multiple levels of detail was vital to the analysis process.

(RQ3) How can we characterize tasks BSS analysts carry out, especially to explain latent temporal and spatial dimensions? Since publications P1–P3 are design studies, we published a task characterization in each of those papers. In P1 we suggested rather low-level tasks, using Brehmer & Munzner's task typology to describe what data BSS analysts need to see and compare. The important parts are, unsurprisingly, the parameter settings, latent dimensions and their loadings (i.e., the unmixing matrix). While the specific characteristics of latent dimensions may vary (e.g., temporal, spatial, spatio-temporal, or other data types altogether) we believe that the task characterization is valid for many BSS models and may be transferred to other latent variable models, such as PCA.

In publication P2, we described necessary tasks for non-stationary SBSS parameter selection, which arise from the theoretical and practical considerations in that process. For instance, the mathematical formulae in SBSS assume disjoint regions and kernels, respectively, so that property must be ensured. Further, their sizes have to complement each other, because a kernel suitable to a small and densely populated region may not capture enough of a point's neighborhood in a more sparsely populated region. Finally, they have to be sensible for the application domain and the data at hand. For instance, a region containing locations on both sides of a sea could be pointless in a geological setting, as the soil may have different properties on the respective shores. Consequently, the tasks we proposed are *quickly and efficiently enter parameter settings*, *balance region and kernel size* and *reconcile possible regions and kernels with [the analyst's] domain knowledge.*

In publication P3, we proposed a novel way to carry out visual sensitivity analysis, based on a sensitivity index derived from a hierarchical clustering. The tasks we proposed in that publication are naturally confined to this particular analysis process. Specifically, to do it successfully, analysts have to *compare the association between parameters and outputs* to find parameter/output pairs that are worth investigating. For any particular parameter/output, analysts have to *assess its overall variation*, i.e., see contained similarity structures such as outliers or tight clusters. This helps *identifying groups of data cases with low/high variation in a parameter/output* to *compare variation between parameters and outputs, both overall and for a group of data cases.* Groups of data cases with more/less variation in the parameter/output space are exactly the stable/sensitive settings analysts are looking for.

Regarding necessary tasks to explain latent dimensions, i.e., relating patterns in latent dimensions to (possible) physical processes causing them, or tasks around the DR process itself we did, unfortunately, not propose much so far. We suggested a quantification framework to measure the association, e.g., between patterns and parameter settings [PBM23], which may be used for that purpose, but more research in that direction is necessary.

(RQ4) Can we adapt approaches suggested to explain multivariate DR to temporal and spatial latent dimensions? Unfortunately could only partially investigate this research question. To some limited extent, our work in existing publications can support it. E.g., BSS experts expect that a stable output is more likely representative of the "real" underlying processes. Thus, outputs and patterns sensitive to particular parameter settings may be excluded from the DR process to begin with (P1, P3). We also only scratched the surface regarding the prevalence of input variables in time and space. It should be possible to add input variables to the framework presented in [PBM23], in which case the association between patterns in the output and input variables could be quantified. In publication P1, we built simple comparative visualizations between input and latent temporal dimensions, which allows to visually match patterns. However, we have to leave specialized visualizations for these tasks for future work (Chapter 7).

(Overall RQ) How can we use VA best practices to aid usage of BSS techniques in time and space? Summarizing the answers obtained for individual sub-questions, we can state that seeing all relevant data in context and the ability to interactively manipulate them proved incredibly useful. Experts mentioned this as an advantage without exception. While not a particularly exciting result, due to its lack of novelty (cf. Section 1.1.1), it shows that also BSS can be tackled with the principles first laid out a few decades ago. Exchanging complex tasks on data with simple tasks on visualizations is another way to phrase the benefits of visualization/VA, and it was very successful, e.g., in the context of BSS parameter selection. Putting the two together allowed to efficiently compare BSS parameter settings and outputs, thus leveraging existing similarity structures in the data both through comparison visualizations and data mining algorithms.

6.2 Contributions

In this thesis, we investigated how BSS, a latent variable model, may support the analysis of multivariate temporal and spatial datasets. A limitation of our research in that regard, which we discuss in more detail in Section 6.3, was the lack of front-line analysts in most evaluations. In P2, we could collaborate with a geochemistry expert; therefore, that publication carries the most information to that end. In components identified with our prototype, the expert was able to identify patterns that were especially interesting considering that the dataset did not contain the chemical elements that should be involved in said pattern (e.g., mineral reservoirs) or, generally, considering that the most interesting elements were missing in the dataset. What BSS was able to do, therefore, is highlight known patterns using fewer variables, which has monetary implications when it comes to, e.g., sample collection and chemical analysis.

Much of our research involved solving challenges associated with BSS itself, such as selecting proper tuning parameters or exploring latent components. Doing so was a prerequisite to tackling the previous question. We covered these in the introduction and related work chapters (1 and 2, respectively). It was necessary to support vPSA tasks [Sed+14] for BSS tuning parameters, which we could explicitly achieve for the tasks *optimization* and *sensitivity* for SBSS (P2 and P3, respectively). For TBSS (P1), participants found that the interactive visualizations help selecting parameters, but we have less evidence that they also lead to suitable outputs than, e.g., in P2. The task to find *outliers* is, in our opinion, handled by the various clustering approaches we devised (P1, P3). While *partitioning* would be likely valuable for BSS analysis as well, that task is impeded by the fact that automatic and representative random sampling of the tuning parameter space is challenging (or even undesired) and was not an avenue we pursued. A reasonably accurate surrogate model for BSS would likely be extremely helpful in investigating the tuning parameter space, but building such a model is also hampered by the difficulty in automatic sampling. A surrogate model would allow a broader range of parameter space navigation strategies and enable the *fitting* PSA task. The *uncertainty* task would become relevant when we consider BSS as part of a larger data pipeline for prediction. Regarding exploring latent components, a helpful framing of the issue was in terms of set relation tasks [Als+16]. The main obstacle to those is that latent components are not perfectly comparable, i.e., in some aspects, they are more alike and less so in others. We proposed an analysis framework based on fuzzy sets [PBM23] to that end but did not implement it in any of our VA prototypes. Instead, we focused on visual comparison and other data mining approaches, such as the set-aware clustering in P1. In our research, we partially touched upon other challenges, like explaining latent components.

A common theme in our research is leveraging similarity information in the data in various ways to support analytic tasks. In P1, the slope graph shows the Pearson correlation between time series and thus allows to verify the similarity of individual components and their sets simultaneously. The set-aware clustering uses similarity information about components to cluster them, but with additional constraints necessary to respect group structures. It thus highlights the main trends and outliers in the components. In P2, the regionalization guidance uses similarity information between measurements at locations to form larger regions. To select a kernel size, the analyst considers the similarity of the variables' spatial dependence shown in the variogram. In P3, we use similarity information to enable sensitivity analysis on parameters and outputs that are semantically more complex objects than numbers. These are our success stories. An example where using similarity data did not prove advantageous was the multidimensional projections in P1, which evaluation participants rarely used. While the amount of data they considered may be part of an explanation of why that happened, another possible theory is that the plots did not show the data well enough. Each dot in the projection represented a *set* of components but did not depict the actual components' data. A similarity-based layout using one glyph for each component may be more concise but comes at the cost of scalability.

Finally, we discuss the transferability and scalability of our contributions. The task description in P1 likely applies to other latent variable models or BSS of other data types. It may guide, e.g., the development of a VA solution using a geographic PCA variant. We expect the proposed visualizations and data mining techniques in P1 to work when the data are groups of time series that must be compared. Our work in P2 is rather specific to SBSS. The automatic regionalization may be applied to geographic datasets in general, but the technique itself is not our contribution as we adapted an existing algorithm. The way how kernels are selected by using the superpositioned variograms as a guideline may be helpful for other geographically weighted techniques, such as the scalograms by Dykes and Brunsdon [DB07], which also often depend on a spatial neighborhood definition. The work proposed in P3 can transfer to other contexts, as we showed in the evaluation, as long as the dissimilarity measures are appropriate for the data. However, when suitable features can be derived, we recommend trying that first, as more widely known visualization idioms (e.g., scatterplots, histograms) and numerical approaches (e.g., linear regression) exist for multivariate data that may be leveraged. Our proposed approaches were designed for the real-world datasets we had available. In P1, the financial dataset had 3 139 time steps and 23 variables, while the medical dataset consisted of 2 500 time steps and 8 variables. In P2, the two datasets were the GEMAS geochemical survey [Rei+98], which consists of 2 108 locations and 18 variables, and the Colorado survey (960 / 27) [SEK10]. While these limits do not accommodate all possible datasets, we expect them to cover a reasonable subset. For P3, we considered 42, 48 (SBSS), and 12 (microclimate) parameter/output combinations, respectively. We added interactions to the main dendrogram visualization that should scale it better to larger datasets, but there will be a point where it will be too cluttered. Other visualization idioms, possibly based on DR scatterplots, could help in this regard.

6.3 Limitations

We discuss limiting aspects to our thesis' contributions in this section. We use the Design Study Methodology pitfalls and the Nested Model's threats to validity (cf. Section 1.1.2) as our main guidelines. Sedlmair et al. [SMM12] propose 32 possible pitfalls that may happen in various stages of a design study. These range from collaborating with the wrong people (*PF-3*) to writing the paper chronologically instead of focusing on results (*PF-31*). Tamara Munzner [Mun09; Mun14] suggests threats to each level's validity in the Nested Model along with possible ways to counter them. In informal terms [Mun09, p. 921] the threats are "they [the target users] do not do that" (L1), "you show them the wrong thing" (L2), "how you show it does not work" (L3), and "your code is too slow" (L4).

6.3.1 Participants

User-centered design is the prevailing design paradigm in visualization, although others exist (e.g., algebraic design [KS14]). Therefore, guidelines emphasize the importance of choosing the right people to collaborate with and ensuring that an important and recurring problem of theirs is solved with visualization research. Munzner's L1 threat ("they do not do that") and pitfall *PF-10* ("no real/important/recurring task") of the Design Study Methodology are two examples of that. There is little doubt that visualization research was necessary and that the problems we tackled are important for someone employing BSS. However, one could argue that we did not actually work with "front-line analysts" [SMM12]. Our collaborators in this thesis, when it came to designing and evaluating proposed VA approaches, were mainly researchers in statistics and mathematics who developed novel methods, such as the discussed BSS methods. Their main job was not the analysis of data that they collected themselves but to propose, develop, prove, and implement statistical approaches. We think that working with statistics researchers was appropriate to answer our VA research questions. However, if the goal was to develop interactive visualizations that facilitate using BSS in general, then the question remains whether our selection of participants is sufficiently representative of the expected end users.

6.3.2 Evaluation, Long-Term Usage, and Gaps

As discussed in the introduction (Section 1.1.1), visualizations are a tool for data analysis. Tool designers are generally interested in whether their proposal is effective. So, to validate against the L1 threat, Munzner [Mun09, Sec. 3.3] proposes to report the designed tool's adoption rates. To validate against the L2 threat ("you show them the wrong thing"), we chose expert interviews for their practicality, but Munzner and also others [SP06] call for long-term field studies. In such studies, researchers observe how front-line analysts use the designed tool over a longer time frame, using ethnographical observation methods, interviews, surveys, and automated activity logs.

Following the previous discussion, it is a bit unclear who should be observed in such a manner (L2). As we designed with statistics researchers, they would be the natural choice, but BSS-supported data analysis is not something they need to do frequently (pitfall *PF-10*).

Adoption rates are also tricky to measure (L1, L2). Our research team published the statistical software on CRAN[1], a platform to host R packages, and the VA prototypes on GitHub[2], a platform to host open source software. The only signal we can get from those platforms is the number of downloads, which is not equal to the number of installations or the number of times someone started the tool. Telemetry, i.e., collecting more detailed usage data in situ and sending them back to the developers, is a strategy to obtain activity logs but comes with privacy concerns. One may also only measure *what* is done, not *why*, so telemetry would need to be supplemented with online surveys and, if possible, interviews. It is not impossible, but long-term studies are challenging with open-source software for a diverse user base.

Another challenge to those is the technical implementation of our VA prototypes. RStudio[3], the popular development environment for R [R C23], offers support for plug-ins in the form of "Shiny Gadgets." Shiny[4] is a web application framework for interactive applications in R. Shiny Gadgets builds on Shiny to extend the functionality of RStudio. It would have been best to implement our concepts as Shiny apps or Gadgets to increase the likelihood of long-term adoption. As we were inexperienced with the Shiny framework, we expected that using it would slow us down (impeding rapid prototyping and development, *PF-22*). In the worst case, we may get stuck later when realizing that a crucial function does not exist or that we need a particular implementation of an algorithm in another programming language. In hindsight, the latter may have been

[1] https://cran.r-project.org/ (accessed 15th May, 2024)
[2] https://github.com/npiccolotto?tab=repositories (accessed 15th May, 2024)
[3] https://posit.co/products/open-source/rstudio/ (accessed 15th May, 2024)
[4] https://shiny.posit.co/ (accessed 15th May, 2024)

unfounded because it was more likely that no implementation existed at all. This was the case, e.g., for REDCAP [Guo08] (re-implemented) and CorrelatedMultiples [Liu+18] (passed). Nonetheless, our technology stack was a web server backend in R and a frontend in JavaScript for P1 and P2, while P3 did not have a backend. As we did not ship the prototypes integrated to RStudio, potential users must install, configure, and run, i.e., deploy, our VA prototypes on their own, which may be daunting and time-consuming.

The previous paragraph described a possible gap between visualization research and applying results in practice. This gap is discussed in the visualization community, e.g., in the *VisGap* workshops held since 2020 at the EuroVis conference. Topics included design approaches [Jän+20; Bra23] or how to provide visualization research as software [Hen23] and to prolong the life of software artifacts [Ise22]. In our research for this thesis, we encountered several gaps in addition to the previously described *deployment gap*.

Deploying software is one part, but since visualizations transform data into images, potential users must also convince the tool to read their dataset. This is likely not an issue when the tool is shipped as a plug-in, as it will either have access to the program's runtime environment or get the data passed as an object. If the tool is separate from the usual analysis environment, one can use well-defined open data exchange formats, such as comma-separated values (CSV) or JavaScript Object Notation (JSON). These two are text-based and may not be amenable to larger datasets, and a binary format would be more prudent. Either way, the onus is on the analyst to convert their dataset to the specified format. Automatic conversion software could be developed and provided, but the analyst's task is still to find, install, and run that converter. We call this the *transmission gap* and expect it to especially apply to P3, where our prototype reads an elaborate JSON file and requires many suitable images to be generated, too. The transmission gap may also work in the other direction: The derived data must get out of the tool and into the next, creating a new transmission gap.

Then, we see a certain *interactivity gap* in the sense that participants (statistics researchers) praised the interactivity of our prototypes (compare Section 6.1) but do not use interactive visualizations themselves in their work. It is not even about bespoke coordinated multiple-view systems but basic plots like a dynamic map with symbols. The main issue we see here is ergonomics. One *can* render an interactive scatterplot in RStudio via a Shiny app. But doing so [Hen23, Fig. 1] requires writing about 20 lines of code, using two libraries, and the result is displayed in a separate window. The example shown on the Shiny homepage at the time of writing even has around 60 lines. Certainly, that is too much to ask of someone who reports that they use `ggplot` [Wic16] only when they need "fancy images for the paper," as one participant in P1 phrased it. Compare this to calling `plot(iris)` that is one line, without libraries, and shows the result in RStuio's designated *Plots* area next to the R code, where most plots reside. Hence, to bring the advantages of interactivity to this user group, it seems promising to aim for the same affordances as R's built-in `plot` function provides:

1. The code to create the plot must fit in one line.

2. One function must cover most plotting needs. Multiple dispatch can facilitate this point, which is also how R's `plot` behaves: The rendered image depends on the data type passed to the function.

3. Ideally, it requires no other parameters than the object holding the data. That entails not only the visualization idiom (e.g., scatterplot matrix or PCP) must be selected but also sensible defaults for its parameters.

4. The output must be shown where all other plots are, too.

However, that indicates yet another gap. The above is not possible without changes to R (adding another function to the `base` package) or RStudio itself (showing Shiny in *Plots* area). RStudio is developed by Posit Software, PBC, and published under the GNU Affero General Public License (AGPL) in version 3. While one may fork the RStudio repository[5] and add the required changes, these changes need to be integrated back into the main repository to benefit everyone. RStudio is also a complex and large software project itself. Hence, we anticipate that (possibly significant) resources from Posit Software will be required to create the necessary modifications. If not, then they at least need Posit's approval. We call this the *jurisdiction gap*, a gap that visualization researchers cannot bridge just with their own efforts.

That being said, it is only point 4 in the list above that creates the gap and one may choose to not tackle it for practical reasons. The existing abstractions and interfaces can still be extended, which is mainly `ggplot`, given that visualization researchers do not have jurisdiction over the R `base` package. `ggdist` [PK20], the probabilistic grammar of graphics, or `ggnetwork` [TBH17], which extends `ggplot` to graphs, are great examples for that.

6.4 Bibliography

[Als+16] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers. "The State-of-the-Art of Set Visualization". In: *Computer Graphics Forum* 35.1 (2016), pp. 234–260. DOI: `10.1111/cgf.12722` (cit. on p. 211).

[Bra23] R. Brath. "Many Types of Design Needed for Effective Visualizations". In: *VisGap - The Gap between Visualization Research and Visualization Software*. The Eurographics Association, 2023. ISBN: 978-3-03868-226-4. DOI: `10.2312/visgap.20231118` (cit. on p. 214).

[DB07] J. Dykes and C. Brunsdon. "Geographically Weighted Visualization: Interactive Graphics for Scale-Varying Exploratory Analysis". In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1161–1168. DOI: `10.1109/tvcg.2007.70558` (cit. on p. 211).

[5] `https://github.com/rstudio/rstudio` (accessed 15th May, 2024)

[Guo08] D. Guo. “Regionalization with Dynamically Constrained Agglomerative Clustering and Partitioning (REDCAP)”. In: *International Journal of Geographical Information Science* 22.7 (2008). DOI: 10.1080/13658810701674970 (cit. on p. 214).

[Hen23] R. Henkin. “Better Information Visualization Software Through Packages for Data Science Ecosystems”. In: *VisGap - The Gap between Visualization Research and Visualization Software*. The Eurographics Association, 2023. DOI: 10.2312/VISGAP.20231115 (cit. on p. 214).

[Ise22] T. Isenberg. “Personal Experiences of Providing and Using Research Prototypes”. In: *VisGap - The Gap between Visualization Research and Visualization Software*. The Eurographics Association, 2022. DOI: 10.2312/VISGAP.20221059 (cit. on p. 214).

[Jän+20] S. Jänicke, P. Kaur, P. Kuzmicki, and J. Schmidt. “Participatory Visualization Design as an Approach to Minimize the Gap between Research and Application”. In: *VisGap - The Gap between Visualization Research and Visualization Software*. The Eurographics Association, 2020. ISBN: 9783038681250. DOI: 10.2312/VISGAP.20201108 (cit. on p. 214).

[KS14] G. Kindlmann and C. Scheidegger. “An Algebraic Process for Visualization Design”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2181–2190. DOI: 10.1109/tvcg.2014.2346325 (cit. on p. 212).

[Liu+18] X. Liu, Y. Hu, S. North, and H.-W. Shen. “CorrelatedMultiples: Spatially Coherent Small Multiples With Constrained Multi-Dimensional Scaling”. In: *Computer Graphics Forum* 37.1 (2018), pp. 7–18. DOI: 10.1111/cgf.12526 (cit. on p. 214).

[Mun09] T. Munzner. “A Nested Model for Visualization Design and Validation”. In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (2009), pp. 921–928. DOI: 10.1109/tvcg.2009.111 (cit. on pp. 212, 213).

[Mun14] T. Munzner. *Visualization Analysis and Design*. CRC Press, 2014. ISBN: 978-1-4987-5971-7 (cit. on p. 212).

[Pic+22] N. Piccolotto, M. Bögl, T. Gschwandtner, C. Muehlmann, K. Nordhausen, P. Filzmoser, and S. Miksch. “TBSSvis: Visual Analytics for Temporal Blind Source Separation”. In: *Visual Informatics* 6.4 (2022), pp. 51–66. DOI: 10.1016/j.visinf.2022.10.002 (cit. on p. 208).

[PBM23] N. Piccolotto, M. Bögl, and S. Miksch. “Multi-Ensemble Visual Analytics via Fuzzy Sets”. In: *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, 2023. ISBN: 978-3-03868-222-6. DOI: 10.2312/eurova.20231092 (cit. on pp. 209, 211).

[Pic+24] N. Piccolotto, M. Bögl, C. Muehlmann, K. Nordhausen, P. Filzmoser, J. Schmidt, and S. Miksch. “Data Type Agnostic Visual Sensitivity Analysis”. In: *IEEE Transactions on Visualization and Computer Graphics* 30.1 (2024), pp. 1106–1116. DOI: 10.1109/tvcg.2023.3327203 (cit. on p. 207).

[PK20] X. Pu and M. Kay. “A Probabilistic Grammar of Graphics”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Honolulu HI USA: ACM, 2020, pp. 1–13. ISBN: 978-1-4503-6708-0. DOI: 10.1145/3313831.3376466 (cit. on p. 215).

[R C23] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2023 (cit. on p. 213).

[Rei+98] C. Reimann, M. Äyräs, V. A. Chekushin, I. V. Bogatyrev, R. Boyd, P. de Caritat, R. Dutter, T. E. Finne, J. H. Halleraker, J. Øystein, G. Kashulina, H. Niskavaara, et al. *Environmental Geochemical Atlas of the Central Barents Region*. Trondheim: NGU Norges Geologiske Undersøkelse, 1998. ISBN: 82-7385-176-1 (cit. on p. 212).

[SMM12] M. Sedlmair, M. Meyer, and T. Munzner. “Design Study Methodology: Reflections from the Trenches and the Stacks”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), pp. 2431–2440. DOI: 10.1109/tvcg.2012.213 (cit. on p. 212).

[Sed+14] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. “Visual Parameter Space Analysis: A Conceptual Framework”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2161–2170. DOI: 10.1109/tvcg.2014.2346321 (cit. on p. 210).

[SP06] B. Shneiderman and C. Plaisant. “Strategies for Evaluating Information Visualization Tools: Multi-Dimensional in-Depth Long-Term Case Studies”. In: *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*. New York, NY, USA: ACM, 2006, pp. 1–7. ISBN: 978-1-59593-562-5. DOI: 10.1145/1168149.1168158 (cit. on p. 213).

[SEK10] D. B. Smith, K. J. Ellefsen, and J. E. Kilburn. *Geochemical Data for Colorado Soils: Results from the 2006 State-Scale Geochemical Survey*. Data Series 520. 2010 (cit. on p. 212).

[TBH17] S. Tyner, F. Briatte, and H. Hofmann. “Network Visualization with Ggplot2”. In: *The R Journal* 9.1 (2017), pp. 27–59. DOI: 10.32614/RJ-2017-023 (cit. on p. 215).

[Wic16] H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN: 978-3-319-24277-4 (cit. on pp. 208, 214).

CHAPTER 7

Future Work

In the following, we discuss some open challenges that we identified and would like to see pursued in the future.

Ensemble ∩ Set Visualization. BSS components represent complex temporal/spatial data and they appear in groups. While we proposed some VA methods to visualize such data (Chapter 3), we think the space of possible visualization designs at the intersection of the two disciplines (presented in Section 2.3 and Section 2.4) is barely explored. Ensemble and set visualization have more in common than they currently seem to realize. For instance, one only has to partition a "traditional" ensemble by ranges of a parameter to obtain complex data that also has categorical information attached. Coming from the other side, one only needs more data available for set elements than a uniquely identifying character sequence to, again, obtain complex data that also has categorical information attached. Possible visualization designs are likely found in the product of visualization approaches for ensemble datasets [Wan+19] and set-typed data [Als+16]. Alsallakh et al. [Als+16, p. 253] also stated "visualizing sets in the context of other data types", "comparing multiple set families", and "visualizing fuzzy and uncertain set memberships" as open challenges that directly relate to our problem context [PBM23]. Wang et al. [Wan+19], on the other hand, mention in their open challenges for ensemble data that the *ensemble* dimension has not been covered by many papers, i.e., that VA for multiple ensembles is understudied. Thus, figuring out what works will benefit both disciplines and also further advance our second research question.

VA for Regionalizations. During the work presented in Chapter 4, we learned how regionalizations, a SBSS tuning parameter, can be useful for geostatistical modeling as a whole. To our knowledge, while regionalizations[1] were discussed in the visualization

[1] While some works in visualization [Wu+17; WBL18; WBL20] discuss *regions* and their construction, they assume a weaker concept where regions are non-contiguous or overlapping.

literature [Guo09], only fully automatic algorithms were proposed so far (see, e.g., references in [DRS07; Ayd+21]). The geochemical expert in our evaluations had extensive knowledge about geological and chemical processes in the survey area that can inform a regionalization, but algorithms can only incorporate what was given to them in a computer-readable format. Further, algorithms often need parameters in advance that are difficult to come up with, such as the number of desired regions or a threshold value for a spatially summative variable [DAR12]. Very few report any kind of uncertainty about the result. For these reasons we think that VA would be well-applied to that problem, potentially impact many fields, and has the potential to answer other related interesting questions, such as by which criteria humans organize and partition spatial fields. These in turn may inform the design of future algorithms.

Visualization of Many Spatially Distributed Variables across Scales. In our work, we compared latent dimensions only globally, but finer-grained comparisons would allow more detailed assessments whether two dimensions can be considered equal. Goodwin et al. [Goo+16] suggested a framework for multivariate visual comparison, in which they distinguish the *number of variables* as *univariate*, *bivariate*, and two sizes of *multivariate* data. In addition, they consider *micro*, *macro* and *global* variation between variables. The authors suggest possible existing visualizations for each combination of the two dimensions, but mention as the limitations of their framework [Goo+16, p. 607] that it is “general, partially populated and contains only broad design guidelines.” Particular designs should be proposed and tested according to specified tasks, which could be related to interactive map tasks [Rot13]. This topic fits into persistent challenges in geovisualization [Çöl+17]: The dimensionality of datasets was mentioned to be challenging as well as developing guidelines to matching visualization types to task types. Progress in this direction will thus benefit geovisual analytics in general. One way to approach these problems is by collaborating with analysts in domains traditionally handling many variables, such as geochemistry. Design studies should surface necessary tasks, which can then be related to the mentioned existing literature. Multiple design alternatives to solve said tasks can be investigated in controlled experiments.

Multivariate Modeling & Prediction. So far we considered BSS detached from one of its stated practical purposes. BSS is promising in the context of multivariate modeling and, as a next step, prediction/forecasting. The advantages are expected to be especially noticeable in contexts with dozens of variables. Bögl et al. [Bög+13, p. 2245] mentioned as future work to “include the performance of the model for forecasting the diagnostic step.” They achieved this in later work [Bög+15], but still only for univariate time series. Sun et al. [Sun+20] integrate the forecasting performance and risk associated with different products and categories into the demand forecasting model selection process, but they also consider only univariate models. Multivariate spatial modeling is challenging, too. As touched upon in Section 1.1.3, to model a spatial field in terms of the second-order dependence, spatial covariance functions are required. Many of these functions are parametric and their parameters have to be estimated for the dataset at hand. Naturally, the number of parameters and their estimation complexity increases with higher dimensionality of the data. The prediction performance is not part of this estimation process. With latent dimensions obtained by SBSS it would be possible, since they are marginally and spatially uncorrelated, to model them separately with univariate approaches, predict seperately, and back-transform the result into the original data space — a potentially much simpler process. To support it properly, initial work in this thesis has to be further developed to be incorporated into a larger pipeline consisting of feature selection, BSS parameter selection, dimensionality reduction, modeling, and prediction. Many of the challenges for data processing pipeline design suggested by von Landesberger et al. [vLFR17, p. 2238–2239] become relevant, such as "un-breaking the user's workflow" (holistically assess the effect of decisions across the whole pipeline), "analyzing and visualizing the flow of uncertainty in pipelines," "assist in parameter choices," or "exploit derived measures." Results may thus advance visualization research in that regard, besides the practical improvements for domain experts.

DR Explainability for Geovisual Analytics. Our fourth research question about applying DR explainability approaches to temporal and spatial latent dimensions is already included in the previous paragraph. Interactive visualizations should support the analyst in deciding which dimensions to keep and which to remove while considering the source of the introduced error, its temporal and spatial distribution, and its impact on, e.g., prediction performance. Achieving these goals likely improves geostatistical modeling and the result can easily be translated to other latent variable models. Another possibly fruitful transfer from DR research to geoanalytics could be attribute-based explanations [Sil+15]. The idea is to highlight which (groups of) variables in the dataset contribute most to the local neighborhood's variance or high-dimensional distances. The approach seems directly applicable to multivariate spatial data. If done so, it could be an overview visualization and thus an alternative to, e.g., small multiples or a clustering. The former gets less efficient with many variables while the latter often introduces information loss via clear-cut categories.

7.1 Bibliography

[Als+16] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers. "The State-of-the-Art of Set Visualization". In: *Computer Graphics Forum* 35.1 (2016), pp. 234–260. DOI: `10.1111/cgf.12722` (cit. on p. 219).

[Ayd+21] O. Aydin, Mark. V. Janikas, R. M. Assunção, and T.-H. Lee. "A Quantitative Comparison of Regionalization Methods". In: *International Journal of Geographical Information Science* 35.11 (2021), pp. 2287–2315. DOI: `10.1080/13658816.2021.1905819` (cit. on p. 220).

[Bög+13] M. Bögl, W. Aigner, P. Filzmoser, T. Lammarsch, S. Miksch, and A. Rind. "Visual Analytics for Model Selection in Time Series Analysis". In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2237–2246. DOI: 10.1109/tvcg.2013.222 (cit. on p. 220).

[Bög+15] M. Bögl, W. Aigner, P. Filzmoser, T. Gschwandtner, T. Lammarsch, S. Miksch, and A. Rind. "Integrating Predictions in Time Series Model Selection". In: *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, 2015. DOI: 10.2312/eurova.20151107 (cit. on p. 220).

[Çöl+17] A. Çöltekin, S. Bleisch, G. Andrienko, and J. Dykes. "Persistent Challenges in Geovisualization – a Community Perspective". In: *International Journal of Cartography* 3 (sup1 2017), pp. 115–139. DOI: 10.1080/23729333.2017.1302910 (cit. on p. 220).

[DAR12] J. C. Duque, L. Anselin, and S. J. Rey. "The Max-P-Regions Problem*". In: *Journal of Regional Science* 52.3 (2012), pp. 397–419. DOI: 10.1111/j.1467-9787.2011.00743.x (cit. on p. 220).

[DRS07] J. C. Duque, R. Ramos, and J. Suriñach. "Supervised Regionalization Methods: A Survey". In: *International Regional Science Review* 30.3 (2007), pp. 195–220. DOI: 10.1177/0160017607301605 (cit. on p. 220).

[Goo+16] S. Goodwin, J. Dykes, A. Slingsby, and C. Turkay. "Visualizing Multiple Variables Across Scale and Geography". In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 599–608. DOI: 10.1109/tvcg.2015.2467199 (cit. on p. 220).

[Guo09] D. Guo. "Flow Mapping and Multivariate Visualization of Large Spatial Interaction Data". In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (2009), pp. 1041–1048. DOI: 10.1109/tvcg.2009.143 (cit. on p. 220).

[PBM23] N. Piccolotto, M. Bögl, and S. Miksch. "Multi-Ensemble Visual Analytics via Fuzzy Sets". In: *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, 2023. ISBN: 978-3-03868-222-6. DOI: 10.2312/eurova.20231092 (cit. on p. 219).

[Rot13] R. E. Roth. "An Empirically-Derived Taxonomy of Interaction Primitives for Interactive Cartography and Geovisualization". In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2356–2365. DOI: 10.1109/tvcg.2013.130 (cit. on p. 220).

[Sil+15] R. R. O. da Silva, P. E. Rauber, R. M. Martins, R. Minghim, and A. C. Telea. "Attribute-Based Visual Explanation of Multidimensional Projections". In: *EuroVis Workshop on Visual Analytics (EuroVA)* (2015), 5 pages. DOI: 10.2312/eurova.20151100 (cit. on p. 221).

[Sun+20] D. Sun, Z. Feng, Y. Chen, Y. Wang, J. Zeng, M. Yuan, T.-C. Pong, and H. Qu. "DFSeer: A Visual Analytics Approach to Facilitate Model Selection for Demand Forecasting". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2020, pp. 1–13. ISBN: 978-1-4503-6708-0. DOI: 10.1145/3313831.3376866 (cit. on p. 220).

[vLFR17] T. von Landesberger, D. W. Fellner, and R. A. Ruddle. "Visualization System Requirements for Data Processing Pipeline Design and Optimization". In: *IEEE Transactions on Visualization and Computer Graphics* 23.8 (2017), pp. 2028–2041. DOI: 10.1109/tvcg.2016.2603178 (cit. on p. 221).

[Wan+19] J. Wang, S. Hazarika, C. Li, and H.-W. Shen. "Visualization and Visual Analysis of Ensemble Data: A Survey". In: *IEEE Transactions on Visualization and Computer Graphics* 25.9 (2019), pp. 2853–2872. DOI: 10.1109/tvcg.2018.2853721 (cit. on p. 219).

[WBL18] Y. Wang, G. Baciu, and C. Li. "Visualizing Functional Regions". In: *EuroVis 2018 - Short Papers*. 2018 (cit. on p. 219).

[WBL20] Y. Wang, G. Baciu, and C. Li. "Visualizing Dynamics of Urban Regions Through a Geo-Semantic Graph-Based Method". In: *Computer Graphics Forum* 39.1 (2020), pp. 405–419. DOI: 10.1111/cgf.13882 (cit. on p. 219).

[Wu+17] W. Wu, Y. Zheng, N. Cao, H. Zeng, B. Ni, H. Qu, and L. M. Ni. "MobiSeg: Interactive Region Segmentation Using Heterogeneous Mobility Data". In: *IEEE Pacific Visualization Symposium*. 2017, pp. 91–100. DOI: 10.1109/pacificvis.2017.8031583 (cit. on p. 219).

List of Figures

List of Tables

List of Algorithms

Acronyms

SOS second-order separation. 18

t-SNE t-Distributed Stochastic Neighbor Embedding. 1, 16, 76

TBSS Temporal Blind Source Separation. ix, xi, 2, 3, 17–20, 22, 23, 25, 27, 69, 105–117, 127, 129, 132–134, 205, 206, 208, 210, 225

UMAP Uniform Manifold Approximation and Projection. 1, 16, 75

VA Visual Analytics. ix, xi, 3, 9, 10, 13, 14, 23–25, 27, 28, 65, 75–77, 105, 143, 171, 207, 208, 210–214, 219, 220, 225

vPSA Visual Parameter Space Analysis. 27, 39–43, 46, 48, 49, 55–60, 210, 225

vSOBI variant of Second-Order Blind Identification. 19, 20, 206

Appendix

Supplemental Material to Section 2.1.1

Papers included in Thematic Analysis

citekey	*title*	*year*	*outlet*	*identifier*
bruckner2010	Result-Driven Exploration of Simulation Parameter Spaces for Visual Effects Design	2010	IEEE TVCG	10.1109/TVCG.2010.190
waser2010	World Lines	2010	IEEE TVCG	10.1109/TVCG.2010.223
malik2010	Comparative Visualization for Parameter Studies of Dataset Series	2010	IEEE TVCG	10.1109/TVCG.2010.20
kerr2010	Toward evaluating material design interface paradigms for novice users	2010	ACM TOG	10.1145/1778765.1778772
matkovic2010a	Interactive Visual Analysis of Multiple Simulation Runs Using the Simulation Model View: Understanding and Tuning of an Electronic Unit Injector	2010	IEEE TVCG	10.1109/TVCG.2010.171
afzal2011	Visual analytics decision support environment for epidemic modeling and response evaluation	2011	VAST	10.1109/VAST.2011.6102457
matejka2018	Dream Lens: Exploration and Visualization of Large-Scale Generative Design Datasets	2018	CHI	10.1145/3173574.3173943
he2020	InSituNet: Deep Image Synthesis for Parameter Space Exploration of Ensemble Simulations	2020	IEEE TVCG	10.1109/TVCG.2019.2934312
bernard2019	Visual-Interactive Preprocessing of Multivariate Time Series Data	2019	EG CGF	10.1111/cgf.13698
torsney-weir2011	Tuner: Principled Parameter Finding for Image Segmentation Algorithms Using Visual Response Surface Exploration	2011	IEEE TVCG	10.1109/TVCG.2011.248
umetani2011	Sensitive Couture for Interactive Garment Modeling and Editing	2011	ACM TOG	10.1145/2010324.1964985
pretorius2011	Visualization of Parameter Space for Image Analysis	2011	IEEE TVCG	10.1109/TVCG.2011.253
pretorius2015	Visual parameter optimisation for biomedical image processing	2015	BMC Bioinfor	10.1186/1471-2105-16-S11-S9
bao2013	Generating and exploring good building layouts	2013	ACM TOG	10.1145/2461912.2461977
millward2013	An operational software tool for the analysis of coronagraph images: Determining CME parameters for input into the WSA-Enlil heliospheric model	2013	Wiley Space	10.1002/swe.20024
coffey2013	Design by Dragging: An Interface for Creative Forward and Inverse Design with Simulation Ensembles	2013	IEEE TVCG	10.1109/TVCG.2013.147
chaudhuri2013	Attribit: content creation with semantic attributes	2013	UIST	10.1145/2501988.2502008
bogl2013	Visual Analytics for Model Selection in Time Series Analysis	2013	IEEE TVCG	10.1109/TVCG.2013.222
ribicic2013	Visual Analysis and Steering of Flooding Simulations	2013	IEEE TVCG	10.1109/TVCG.2012.175
holbein2018	Parameter Space Comparison of Inertial Particle Models	2018	VMV	10.2312/vmv.20181254

khan2019	GenYacht: An interactive generative design system for computer-aided yacht hull design	2019	Elsevier Oce	10.1016/j.oceaneng.2019.106462
desai2019	Geppetto: Enabling Semantic Design of Expressive Robot Behaviors	2019	CHI	10.1145/3290605.3300599
orban2019	Drag and Track: A Direct Manipulation Interface for Contextualizing Data Instances within a Continuous Parameter Space	2019	IEEE TVCG	10.1109/TVCG.2018.2865051
swearngin2020	Scout: Rapid Exploration of Interface Layout Alternatives through High-Level Design Constraints	2020	CHI	10.1145/3313831.3376593
umetani2012	Guided Exploration of Physically Valid Shapes for Furniture Design	2012	ACM TOG	10.1145/2185520.2185582
koyama2014	Crowd-powered parameter analysis for visual design exploration	2014	UIST	10.1145/2642918.2647386
luboschik2014	Supporting the integrated visual analysis of input parameters and simulation trajectories	2014	Elsevier Com	10.1016/j.cag.2013.09.004
beham2014	Cupid: Cluster-Based Exploration of Geometry Generators with Parallel Coordinates and Radial Trees	2014	IEEE TVCG	10.1109/TVCG.2014.2346626
sorger2016	LiteVis: Integrated Visualization for Simulation-Based Decision Support in Lighting Design	2016	IEEE TVCG	10.1109/TVCG.2015.2468011
poco2014	Visual Reconciliation of Alternative Similarity Spaces in Climate Modeling	2014	IEEE TVCG	10.1109/TVCG.2014.2346755
doraiswamy2015	Topology-based catalogue exploration framework for identifying view-enhanced tower designs	2015	ACM TOG	10.1145/2816795.2818134
ruppert2014	Visual access to an agent-based simulation model to support political decision making	2014	i-KNOW	10.1145/2637748.2638410
matkovic2014	Visual Analytics for Complex Engineering Systems: Hybrid Visual Steering of Simulation Ensembles	2014	IEEE TVCG	10.1109/TVCG.2014.2346744
weissenbock201	PorosityAnalyzer: Visual analysis and evaluation of segmentation pipelines to determine the porosity in fiber-reinforced polymers	2016	VAST	10.1109/VAST.2016.7883516
yumer2015	Semantic shape editing using deformation handles	2015	ACM TOG	10.1145/2766908
frohler2016	GEMSe: Visualization-Guided Exploration of Multi-channel Segmentation Algorithms	2016	EG CGF	10.1111/cgf.12895
obermaier2016	Visual Trends Analysis in Time-Varying Ensembles	2016	IEEE TVCG	10.1109/TVCG.2015.2507592
luboschik2015	Feature-Driven Visual Analytics of Chaotic Parameter-Dependent Movement	2015	EG CGF	10.1111/cgf.12654

berseth2021	Interactive Architectural Design with Diverse Solution Exploration	2021	IEEE TVCG	10.1109/TVCG.2019.2938961
walch2020	LightGuider: Guiding Interactive Lighting Design using Suggestions, Provenance, and Quality Visualization	2020	IEEE TVCG	10.1109/TVCG.2019.2934658
liu2018b	Understanding the Relationship Between Interactive Optimisation and Visual Analytics in the Context of Prostate Brachytherapy	2018	IEEE TVCG	10.1109/TVCG.2017.2744418
hazarika2020	NNVA: Neural Network Assisted Visual Analysis of Yeast Cell Polarization Simulation	2020	IEEE TVCG	10.1109/TVCG.2019.2934591
liu2021b	Supporting the Problem-Solving Loop: Designing Highly Interactive Optimisation Systems	2021	IEEE TVCG	10.1109/TVCG.2020.3030364
bernard2018	Combining the automated segmentation and visual analysis of multivariate time series	2018	EuroVA	10.2312/eurova.20181112
schulz2017	Interactive design space exploration and optimization for CAD models	2017	ACM TOG	10.1145/3072959.3073688
umetani2014	Pteromys: interactive design and optimization of free-formed free-flight model airplanes	2014	ACM TOG	10.1145/2601097.2601129
cibulski2017	Super-Ensembler: interactive visual analysis of data surface sets	2017	SCCG	10.1145/3154353.3154362
eichner2020	Making Parameter Dependencies of Time-Series Segmentation Visually Understandable	2020	EG CGF	10.1111/cgf.13894
biswas2017	Visualization of Time-Varying Weather Ensembles across Multiple Resolutions	2017	IEEE TVCG	10.1109/TVCG.2016.2598869
steiner2017	Integrated Structural–Architectural Design for Interactive Planning	2017	EG CGF	10.1111/cgf.12996
gunther2016a	Inertial Steady 2D Vector Field Topology	2016	EG CGF	10.1111/cgf.12846
matkovic2017	Quantitative Externalization of Visual Data Analysis Results Using Local Regression Models	2017	CD-MAKE	10.1007/978-3-319-66808-6_14
schwarzl2019	Cellpackexplorer: Interactive model building for volumetric data of complex cells	2019	Elsevier Com	10.1016/j.cagx.2019.100010
zaman2015	GEM-NI: A System for Creating and Managing Alternatives In Generative Design	2015	CHI	10.1145/2702123.2702398
ribes2019	A Visual Sensitivity Analysis for Parameter-Augmented Ensembles of Curves	2019	ASME Journa	10.1115/1.4046020
unger2012	A Visual Analysis Concept for the Validation of Geoscientific Simulation Models	2012	IEEE TVCG	10.1109/TVCG.2012.190
konev2014	Run Watchers: Automatic Simulation-Based Decision Support in Flood Management	2014	IEEE TVCG	10.1109/TVCG.2014.2346930

Data Characteristics of Surveyed Papers

paper	data-a	data-t	data-s	data-st	data-ta	data-sa	data-sta	param-a	param-t	param-s	param-st	param-ta	param-sa	param-sta	output-a	output-t	output-s	output-st	output-ta	output-sa	output-sta
101	*3*	*2*	*31*	*4*	*11*	*15*	*5*	*63*	*5*	*27*	*6*	*2*	*6*	*5*	*10*	*0*	*19*	*20*	*21*	*26*	*10*
sorger2016			x										x							x	
gunther2016a		x				x				x	x						x	x			
liu2021b				x				x										x			
amirkhanov2010			x							x										x	
bogl2013					x			x	x										x		
waser2010			x	x										x				x			
bruckner2010								x										x			
bernard2016					x			x											x		
schulz2017								x									x			x	
wang2017			x					x		x											x
dayama2020			x							x							x				
dang2015			x					x									x				
prevost2013										x					x						
beham2014								x									x				
bao2013										x					x						
abuzuraiq2020										x					x						
afzal2011	x		x									x									x
bernard2012a					x			x											x		
bernard2018					x			x											x		
bernard2019					x			x											x		
berseth2021			x							x										x	
biswas2017			x					x		x											x
bock2015										x	x							x			
bogl2014					x			x	x										x		
bors2017					x				x										x		
brunhart-lupo2016								x											x		
bryan2015	x						x	x													x
chaudhuri2013			x										x				x				
cibulski2017								x													x
coffey2013								x									x			x	
desai2019			x					x										x			
doraiswamy2015			x							x										x	
eichner2020					x			x											x		
frohler2016						x		x												x	
gavrilescu2010			x					x									x				
geilinger2018										x	x							x			
hao2011					x			x											x		
hazarika2020								x												x	
he2020								x									x				
holbein2018		x	x	x		x		x										x			
johnson2019			x					x		x								x			
kazi2017			x							x							x				
kerr2010			x					x												x	
khan2019								x									x				
konev2014			x	x										x				x			
koyama2014						x		x												x	

paper	data-a	data-t	data-s	data-st	data-ta	data-sa	data-sta	param-a	param-t	param-s	param-st	param-ta	param-sa	param-sta	output-a	output-t	output-s	output-st	output-ta	output-sa	output-sta
koyama2020						x		x												x	
lin2013			x										x							x	
liu2018b			x										x							x	
luboschik2015								x										x			
luboschik2012a								x											x		
luboschik2014								x											x		
malik2010			x					x												x	
marsault2013										x					x						
matejka2018								x									x				
matkovic2010								x										x			
matkovic2010a	x							x	x												x
matkovic2014								x							x				x		
matkovic2017								x											x		
matkovic2013								x											x		
merrell2011			x							x					x						
millward2013											x							x			
mohiuddin2020								x									x				
odonovan2015										x					x						
obermaier2016								x													x
orban2019								x											x		
poco2014								x											x		
pretorius2011						x		x												x	
pretorius2015						x		x												x	
raidou2016						x		x												x	
raidou2014						x		x												x	
raith2021							x	x												x	
ribes2019								x											x		
ribicic2013			x											x				x			
ribicic2012			x											x				x			
rohlig2015					x			x											x		
rojo2018							x	x		x	x							x			
ruppert2014								x											x		
sagrista2017						x				x	x							x			
sagrista2020							x		x	x								x			
schreck2012					x			x											x		
schultz2013						x		x												x	
schulz2018										x					x						
schwarzl2019								x									x				
steed2013												x									x
steiner2017			x							x										x	
swearngin2020			x							x							x				
torsney-weir2011						x		x												x	
umetani2012										x					x						
umetani2011			x							x							x				
umetani2014										x								x			
unger2012							x	x													x
vonlandesberger2						x		x												x	

paper	data-a	data-t	data-s	data-st	data-ta	data-sa	data-sta	param-a	param-t	param-s	param-st	param-ta	param-sa	param-sta	output-a	output-t	output-s	output-st	output-ta	output-sa	output-sta
walch2020			x										x							x	
waser2014			x			x								x	x			x			
weissenbock2016						x		x												x	
woodbury2017								x									x				
wu2011										x										x	
yanez2017			x					x													x
yumer2015			x										x				x				
zaman2015								x									x				

Online Material

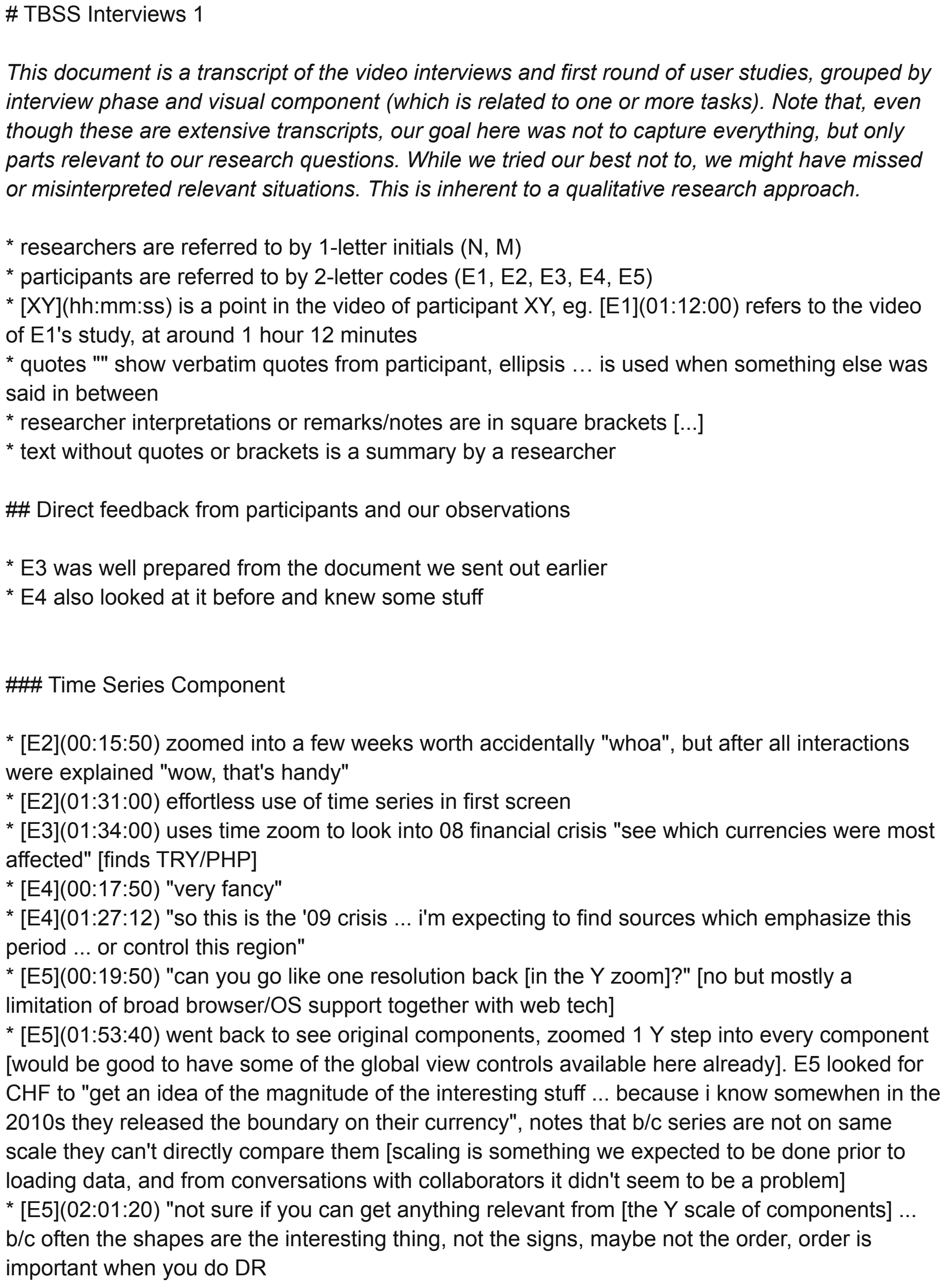
All supplemental material to P5 may also be found on the open access web page of the article: https://doi.org/10.1111/cgf.14785 (accessed 15th May, 2024). Supplemental material not reproduced in this thesis:

- Archive: https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2Fcgf.14785&file=cgf14785-sup-0001-data.zip (accessed 15th May, 2024). Contains list of excluded papers, final codebook, and a bibtex file of all papers in the survey.

Supplemental Material to Chapter 3

Interview Transcripts

First Interview



# TBSS Interviews 1

This document is a transcript of the video interviews and first round of user studies, grouped by interview phase and visual component (which is related to one or more tasks). Note that, even though these are extensive transcripts, our goal here was not to capture everything, but only parts relevant to our research questions. While we tried our best not to, we might have missed or misinterpreted relevant situations. This is inherent to a qualitative research approach.

* researchers are referred to by 1-letter initials (N, M)
* participants are referred to by 2-letter codes (E1, E2, E3, E4, E5)
* [XY](hh:mm:ss) is a point in the video of participant XY, eg. [E1](01:12:00) refers to the video of E1's study, at around 1 hour 12 minutes
* quotes "" show verbatim quotes from participant, ellipsis … is used when something else was said in between
* researcher interpretations or remarks/notes are in square brackets [...]
* text without quotes or brackets is a summary by a researcher

## Direct feedback from participants and our observations

* E3 was well prepared from the document we sent out earlier
* E4 also looked at it before and knew some stuff

### Time Series Component

* [E2](00:15:50) zoomed into a few weeks worth accidentally "whoa", but after all interactions were explained "wow, that's handy"
* [E2](01:31:00) effortless use of time series in first screen
* [E3](01:34:00) uses time zoom to look into 08 financial crisis "see which currencies were most affected" [finds TRY/PHP]
* [E4](00:17:50) "very fancy"
* [E4](01:27:12) "so this is the '09 crisis ... i'm expecting to find sources which emphasize this period ... or control this region"
* [E5](00:19:50) "can you go like one resolution back [in the Y zoom]?" [no but mostly a limitation of broad browser/OS support together with web tech]
* [E5](01:53:40) went back to see original components, zoomed 1 Y step into every component [would be good to have some of the global view controls available here already]. E5 looked for CHF to "get an idea of the magnitude of the interesting stuff ... because i know somewhen in the 2010s they released the boundary on their currency", notes that b/c series are not on same scale they can't directly compare them [scaling is something we expected to be done prior to loading data, and from conversations with collaborators it didn't seem to be a problem]
* [E5](02:01:20) "not sure if you can get anything relevant from [the Y scale of components] ... b/c often the shapes are the interesting thing, not the signs, maybe not the order, order is important when you do DR

### 2D Result Projections / EXPLORE results

* [E1](00:26:00) needed 3 explanations of those, but possibly just didn't get the connection between visual encodings and the task until N pointed it out explicitly. after N explained everything, E1 was frowning [seemed still confused]
* [E1](01:20:20) "what is significance of these crosses [points to k1/k2 projection] like where they are located ..." [forgot / didn't get that location is irrelevant in these, but explanation seemed to help]
* [E1](01:21:30) used projection to select 2 similar results, also used them to determine weight parameter
* E1 generally in the following used projections quite extensively to reason about different results on a higher level
* [E1](01:32:00) used selectboxes interestingness->shape
* [E2](00:22:00) didn't get that 1 point is for 1 result at first, seemed to understand the explanation that followed, esp. the near=similar aspect, at [E2](00:26:30) however they didn't seem to get the size encoding
* [E2](00:24:50) surprised "interesting" after N explained how component projection works, confirms that they don't usually look at their data like this
* [E2](00:28:30) during tasks: scrolls and hovers quickly in many places, "not sure if i understood it correctly", guesses a correct pair but frowns [not satisfied], awkward moments where neither researchers nor participant are sure if they should ask something, M chimes in [suspecting that source of participant's dissatisfaction was related to component projection, even though E2 referenced k1 projection] starting an explanation by telling E2 "it" [what?] is because component projection shows shape similarity instead of interestingness, but after E2 changed it realizes that it wasn't, N explains again near=similar, then everyone moves on
* [E2](01:33:00) asks if there are results already [forgot about precomputation] and what the opacity means in the projections
* [E2](01:41:00) checks the projections [not sure what they looks for]
* [E2](01:53:00) confirms that a new and an old, which the new one was based on, result are similar in k1 space by looking at that projection. or E2 accidentally looked at k1 space and intended to see component space. E2 mentions the two are similar which is expected because only a few lags were added.
* [E3](00:25:00) understood everything it seems, and solved the tasks
* [E3](01:50:00) used projections to judge if a new parametrization is the one they just entered by comparing it to the blue one which didn't have a similar result in k1 space before
* [E4](00:26:20) compared a pair of results *between* projections, also seemed to understand the projections quite well from the start. "i overlooked [results with X symbol]"
* [E4](01:19:45) asks if distances can be compared between the projections [generally no, since MDS projects every space separately. neither can be the size of points, probably.]
* [E4](01:27:40) wanted to select more/all simultaneously as they checked out all precomputed results [he didn't unselect any after the parameters, so an overview would not have helped much?]

* [E4](01:29:00) used selectbox to only compare component 1 of all results [short confusion b/c projection compared skewness but components looked differently]
* [E4](01:35:00) "how many-dimensional is this [shape / component 1] plot in reality?" [N didn't know in the situation, it's (amount of successful results) many], E4 then concluded that there is not so much difference between green and blue [while comparing still only 1st component, but later they said there was no reason b/c they were not actually similar] and dropped one for easier comparison
* [E5](00:25:50) asks detailed questions about how projections were obtained. what vectors go in, what algorithm is used to project etc. "so [k1/k2 projections] are not related to any blind source separation? [not sure if N understood question, but answer was no since it's computed without unmixing matrices or components]", E5 later has no questions around projections [and understand it], just asks about k1/k2 difference
* [E5](00:29:30) tasks: finds 2 near points "based on what you said earlier", comparable 1st component E5 solved after hint to selectboxes
* [E5](00:32:00) "what's the X and Y in this component plot? [lengthy explanation by N, N: does that help?] yeah, so you use MDS at some point ... [inaudible]"
* [E5](01:41:00) selects all results to see which parameters were precomputed. makes new parametrization with only linear part "something i'm more familiar with", applies empty PCP filter, removes all default lags, interacts a lot with 'daily' view which in the prototype is slowish. initially prefers EV diff over autocorrelation but notes that larger lages have higher diff "which is maybe not good ... so much of the data gets discarded with these ... maybe i'll use autocorrelation instead"

### Inter-result Clustering / EXPLORE components

* [E1](00:34:50) found stable/unstable components immediately
* E1 referred to a cluster as 'component' and described it as 'unstable' if the contained components didn't look similar [in their semantics some group of components is also a component]
* [E1](01:21:34) set k to outside of bar charts
* [E1](01:30:00) inspected first components of selected results (were clustered together), found a "disagreeing" component and selected that result too
* [E1](01:37:30) used k to investigate clustering more
* [E2](00:31:00) frowns during explanation [not understanding it?], but seemed to have a little a-ha moment when N explained how to choose K and how a small K is useful, and also when they saw the contained components "ah nice"
* [E2](00:36:50) had no trouble interpreting stable/unstable vis, just asked about meaning of color
* [E2](00:38:40) asked why no blue color [from not converging run] is visible in this view, but it made sense to E2 after N reminded that not converging means no components
* [E2](00:39:40) "what is your criteria for convergence, how do you check?" [not sure if E2 was satisfied with answer]
* [E3](00:33:00) asked about ordering of time series in a cluster and how to compare between clusters

* [E3](00:36:50) during tasks: found stable component quickly, for unstable E3 increased k a bit and found a single-element cluster
* [E4](00:32:20) would appreciate to see which of the components in a cluster is the most central one [possibly use bold font?]
* [E4](00:33:20) also, with similar semantics as E1, referred to a cluster as a single 'component'
* [E4](01:20:20) provides interpretation of the diagonal "if most bars are on the diagonal, then most components agree" [true for lowest k]. shortly after E4 asks for other DOI functions, "something based on time ... just thinking out loud ... maybe something autocorrelation-based ... but then you need to choose the lag" [reason was that E4 saw an "interesting" component (which? the custom sine?) but neither of the DOI functions sorted it on the extreme end, so to find it one would have to look in the middle]
* [E4](01:31:00) used the view to decide on a DOI function and get overview of components [as intended]
* [E5](00:37:00) wants to confirm that clusters don't contain components from same parametrization, wants to know "what we apply the k-means clustering to", wants to know if "do you need to do anything to fix the sign ... what's going on before the clustering" [first one ever to ask about this, seemed ok with answer], wants to confirm that all components were produced by the same method, then "nice... perfect ... i think i understand now", a bit later E5 notes that the DOI function changes the clustering view [N should have explained again why and how, but didn't], E5 solves both tasks quickly, E5 also seemed to apply semantics as E1 and E4 as they apparently looked for a cluster that all very different components [unstable group] instead of a cluster with all but one black bar [mostly stable group but 1 outlier]

### Parameter Comparison / COMPARE parameters

* E1 [seemed confused about / not fully understand the granularity setting]
* [E1](01:30:30) immediately saw that green result used much higher lags than the others, inferred that this is what made the result so different
* [E2](00:42:00) [not sure if E2 agrees with the task of "find large holes in lag set"]
* [E2](00:43:00) [seems to agree that coarser analysis of lags is useful, and getting the granularity part]
* [E2](01:34:20) wonders why a k1 lag set is missing [weight was set to 0, maybe a message would be good]
* [E2](02:03:30) wasn't sure if lag granularity is only a view parameter or if it changes/computes results, and why it changes the projections
* [E3](00:43:40) "but you cannot anymore see how many are in each interval ... maybe it would be a good thing [to see] ... puts more weight [if more lags are used around a certain position]" [we could height-encode the bars then]
* [E3](01:36:30) compares parameters of 3 initially selected results, finds 1 has no linear part, 1 does not use short lags. a few minutes later checks the view again to compare only k2 lags, then goes "ah that explains [not sure what] [because k2 param often dominates?]"
* [E4](00:37:30) figured out alone how/why lag granularity setting changes projections
* [E4](01:31:40) increased lag granularity to 3 [presumably to see if any bars align then, but they didn't]

* [E5](00:45:20) had no questions but relation between lag granularity, projection and lagset vis was a bit unclear ("why does the projection change?")
* [E5](01:47:30) mentions that E5 usually would try [a default set of 1..12] and not the one E5 just chose, so it'd be interesting to compare the two and it's good that it was already computed

### Model Checking / COMPARE model assumptions

* [E1](00:43:00) found visual differences between the charts, but did not know how to interpret
* [E2](00:46:00) no reaction after explanation of eigenvalue diff chart [did E2 know how what it shows], E2 didn't get that charts below plot values per lag, seemed to understand the intended purpose of the charts but not sure if E2 agrees that's useful
* [E2](01:35:00) "so here it [what?] goes down, so maybe it's not using the optimal set, do i understand it correctly?" [N thinks what happened is that because the diagonality was not perfect, ie. 0 at all lags, E2 meant that theoretically there might exist a result with more diagonal scatters. N's subsequent questions confused/intimidated E2 in some way because E2 became defensive and moved on.] in the end E2 mentions that "difference is very small so not sure if it's relevant [regardless of what it means]"
* [E3](00:49:00) knew [by heart or by script, not sure] how to interpret diagonality charts
* [E3](01:28:50) noticed difference in diagonality for new parametrization "what happened ... why did it get worse" [N: maybe b/c of a few larger lags?] "yeah ... maybe they were not helping"
* [E3](01:37:50) finds no difference in scatter diagonality between 3 initially selected results
* [E4](00:40:10) "not sure about the interpretation ... so this tells how well the method works ... [continues thinking hard, was a bit confused why line charts are different, later remembers that all parametrizations find a different unmixing matrix] ... jaja my mistake i get this now [N points out 2 connected analysis tasks] should i look at the line overall or only at the points with the triangle ... so i guess [this parametrization] tries to make all these [scatters] diagonal, but it has no guarantees outside ...[ends dicussion] but ok, it's interesting, it shows somehow the full picture, but this is something i have never seen before", E4 asks for superimposition of the curves for better comparison
* [E4](01:32:00) "all curves look pretty similar ... some roughness at the end, not sure what that is ... seems more unstable for higher lags, which makes sense when you have less data" [synthetic dataset did not exhibit this though]
* [E5](00:47:40) asks what `W` is, states 2x that they understands but didn't explain what they sees, there's questions around the meaning of X ("is it time or is it lag" [i assume stats people rarely have a calendar on the X axis that's why it's not obvious]), and around the X resolution ("is this one or three triangles" ... "can i zoom"), "i don't really understand what's happening here or is it just that [the triangles in front] don't affect the values [in the back]? ... [N clarifies a bit] it seems that the [tang et al triangles] were chosen quite well here ... if the goal is diagonal scatters ... because it's quite low [in the front] ... could you use this graph to choose new lag sets? does this reveal any information regarding that? ... [some confusion around what `W` is] ... i think it's quite clear now ... i think they are very nice ... very interesting pictures ... to see what other lags a W also diagonalizes ... very useful"

### Matrix Comparison / COMPARE unmixing matrices

* [E1](00:46:00) solved all tasks immediately, did not seem to have troubles
* [E1](01:31:00) used MD index, then noticed DKK column "there is something about that series that is interesting"
* E2 seemed to understand the encodings, but again to not really the purpose behind
* [E2](01:37:20) used this view to verify if results are identical, after looking at components
* [E3](00:50:00) had short confusion what the selectboxes above component projection pertain to
* [E3](00:52:00) mentions that judging MD index value between the extrema depends on dimensionality of the data, ie. 0.5 is a different flavor of similar for 3x3 than for 20x20 matrices. then E3 visually compared components between matrices, unsure if that's allowed: you'd be comparing value distributions in a normalized 0..1 space. if a row had values (1, 10, 100) it'd look the same when multiplied by any constant, but you wouldn't necessarily consider the components similar/same?
* [E3](01:39:30) uses both views to judge similarity between results [N points to DKK] [E3 thinks] "it's main contributor for many components for some reason" [N: odd?] [E3 wants to see DKK input series, N points out a way to do it] "probably because of the scale ... makes me wonder if it's the most informative thing to look at ... b/c mostly depending on the scale of the currency ... maybe [the data] should be rescaled prior to this analysis"
* [E3](02:01:00) suggests something that gives better intuition about high-dimensional MD index [possibly just showing the MD index of two random matrices next to the heatmap?], but also says "ofc you can look at the components themselves"
* [E4](00:46:50) notices that X11 column has a lot of black squares in all matrices, wants to see X11 to investigate why
* [E4](00:47:50) mentions like E3 that judging MD index is difficult because "there is no clear interpretation of what the number means when it's not near the extremes, is 0.5 ... what does it mean? ... if there was a scale/legend then i'd like to have it here"
* [E4](01:32:20) "ok now here's the clear thing" [after clicking through the previous tabs, maybe that one should be earlier], E4 notices purple and orange are very similar, goes back to parameters, notices wildly different k1 and concludes that "they did not play a role and information is in the quadratic part ... so i would look at them in more detail [why?]", E4 notices DKK column "for some reason very influential ... also [HKD]", in the end E4 drops one of the two b/c they're the same based on MD index
* [E4](01:51:20) "let's see how [the new result] changed compared [to the old] ... hard to say from looking [at the unmixing matrices]"
* [E5](00:55:00) "we did something similar [to the unmixing matrix heatmap] in a paper", the paper in question doesn't appear to have such a figure though. tasks: E5 mentions that MD index seems too high compared to the visual similarity assessment [of the first 3 rows and first 3 columns] "but maybe not if you look closer ... [based on MD index] unmixing matrices are kind of different"
* [E5](01:38:30) asks if there's a way to export the graphics [like 'save as']
* [E5](01:48:50) mentions again that MD index is "surprisingly large ... maybe as large as they can even be ... but i'm not sure" [is it the same dimensionality thing others mentioned?]

* [E5](01:57:00) notes the dominant DKK series and doesn't like that it effectively hides the other factors [N asks if whitened input data would be an effective solution, but no answer was given]
* [E5](02:09:40) asks for a bivariate color scheme instead of grayscale but realizes that's for another purpose (correlation matrix) and concludes the current version works fine

### Component Comparison / COMPARE components

* [E1](00:49:20) tried to hover over a line to see actual correlation value
* [E1](00:51:00) completed 'most different component' task immediately [knew how clustering supports them there]
* [E1](00:52:40) after a hint to superimpose, E1 found time ranges where components are different
* E1: correlation slope tasks were not a problem
* [E1](01:34:10) used correlation slope to verify assumption that components are all the same
* [E1](01:37:50) exploring clustering and correlation slope
* [E2](00:55:00) [when superimposing] "how do you match these"
* [E2](00:56:00) correlation slope made sense immediately, "nice"
* [E2](00:57:00) tasks: [clustering] mistook/guessed the lowest bar would be the most different component initially [which in retrospect is in a way correct, all other components had much bigger bars], but then they used the clustering as intended [superimpose] completed it after rephrasing question [correlation] completed it on first try
* [E2](01:36:40) "these look very similar", then used superimpose and slope to verify this, "almost identical ... that's quite surprising ... when you look at the lag sets how do you get such similar results ... i am surprised"
* [E2](01:39:50) used superimpose with 3 results, which works but is not terribly useful when all 3 results are different. E2 did not announce an insight either. correlation slope, used after, showed better results.
* [E2](01:53:30) judged [too?] quickly two sets of components as "similar" in the side-by-side view, used "superimpose" which showed quite some differences and repeats "very similar, not surprising", N asks about it and E2 points to k1 projection "they're similar here" and then to component projection "but there are some differences here" [not sure if they got the mental connection between component projection and thick correlation slope lines] then E2 goes back to superimposed view "there are not much differences here" [is it a color scheme issue, as blue and purple were used? but the bars are there to unambiguously tell that stuff is not identical]
* [E3](01:01:00) tried to get a tooltip with actual value of pearson correlatoin in slope graph
* [E3](01:04:00) about the clustering: "not sure what would be the practical use ... maybe it could sometimes help ... haven't thought about this" [it is true that b/c components are standardized to 0 mean and unit (1) variance, any differences tend to equal themselves out when considering the whole time series. possible countermeasures: project back to original space and cluster there, or cluster based on interestingness, or cluster only on a part of the time series]
* [E3](01:44:40) "first component in each is such that there's something happening in 01/2001 ... otherwise pretty constant ... other components have this 2008 [volatility]" [N points out controls,

E3 uses clustering] "it's not so easy to see why these clusters are different ... [there are] similar looking components in different clusters", E3 suggests clustering based on different metrics such as interestingness might be useful too but "not sure what i would use", E3 superimposes 3 results and wants to see only 2
* [E3](01:51:00) uses correlation slope to see how components changed with same lags but no quadratic part
* [E4](00:53:00) "[sees slope graph] very nice"
* [E4](00:55:00) tasks: [clustering] figures out intended use [correlation] no problem [superimpose] no problem, even uses X and Y zoom, which others so far didn't [too hesitant to try?]
* [E4](01:23:00) wonders how to best get correlation slope between red and green, N suggests to unselect blue [a more direct way to accomplish this would be to switch the color/result ordering with drag&drop in the toolbar]
* [E4](01:37:10) superimposes 3 sets but nothing specific emerges, then separates and clusters [k=4] "if i knew something about the data i would try to interpret the clusters" [nothing specific emerges here], then checks correlation "no thick bars". E4 sums up thought process that working theory is information is in k2, so they would compare 1st components of [another k2-"heavy" (more lags used in k2 than k1)] and the current truth-pick, compare the spikes, see what happened and try to interpret [M suggests some annotation in time that shows up in all time series, combined with going back / viewing input series again.] "i think that would be useful here ... [but not only 1, also the others to compare]"
* [E4](01:53:00) used correlation slope to compare new and old result
* [E5](01:05:20) also tried the tooltip thing on the lines
* [E5](01:10:00) tasks: in clustering E5 didn't understand that task is to look for most different in a single set, later it was cleared up and they now "sees how the clustering would be a good idea". other tasks were solved pretty well.
* [E5](01:50:40) is first person to Y zoom into all first components and compare side by side [after trying superimposition which maybe didn't highlight differences well enough?]. a bit later E5 had problems reading the DOI value from the last component. E5 mentions then that having the ability to sort on absolute value would be nice as last series with skewness had biggest absolute value
* [E5](01:53:00) notes spike in first components is on same date, asks what happened

### Parameter Input Component / EXPLORE parameter space

* [E1](00:56:35) after N explained that bar chart shows usage of weight and it helps to find parameters one didn't try yet: "that's good"
* [E1](00:56:45) after lag selection components were visible: "whoa" [overwhelmed]
* [E1](01:00:00) did not know how to interact with PCP [used it for first time] and it wasn't introduced before. then "now i understand"
* [E1](01:01:50) multivariate acf analogy made sense to him, scatterplot i'm not sure
* [E1](01:05:50) struggles with PCP interactoin
* [E1](01:07:10) tried to click label to select [maybe that would have been more expected and simple?]

* [E1](01:08:10) N asked to select 3 more lags, E1 used the first 3 available [because convenient or because something else?]
* [E1](01:38:30) tries new param. with lag that was not used before. "not sure where to start" with lag selection. N gives suggestion. E1 has trouble interacting with PCP again b/c E1 forget that multiple brushes are possible. unselected an accidentally selected lag without problems. tried right click to reset brush [probs because that works for time series where brushing is supported too]. after that E1 seemed confident using PCP and other views.
* [E1](01:42:30) looked at scatterplot and filtered for autocorrelation before [not sure if E1 expected the selected time series to update based on the one with most autocorrelation, b/c there seemed to be a disconnect in their mind between much autocorrelation and the pattern of the scatterplot]
* [E1](01:44:40) wanted to see acf for 1y lag, but had trouble finding it just from looking at acf plot with multiple lags, then filtered years axis [maybe good if one could switch between lag and calendar labeling in acf?]
* [E1](01:46:00) had questions around the calendar fit metric
* E1 seemed to enjoy all the exploration though, was really deep in it
* E1 didn't use the eigenvalue metric [probs b/c E1 didn't know how to interpret it, N didn't explain]
* [E2](01:03:00) was not familiar with PCP vis/interaction
* [E2](01:09:25) "hmm" and smile after explanation of PCP usage [one could read it as E2 prefers a table hehe]
* [E2](01:14:10) tasks: E2 was still struggling with the interactions and interpretation, needed very specific instructions to complete the task ("go here, do this, look for that, go there..."), N then skipped the other tasks to reduce frustration
* [E2](01:21:20) clicked a lag in ACF but wasn't selected [i think E2 managed to click one of the 1px lines when E2 was searching for a tooltip]
* [E2](01:23:00) "interesting" [as in 'I don't get it'], M explains PCP more, "nice"
* [E2](01:38:00) "which one is used with 0/1 weight? i always forget". later, pointing out differences in parameters, E2 refers to a lag set of 1-300 as "quite long lags" [which shows that they don't usually consider lags in the higher 100s/1000s]. E2 mentions the amount of lag as a difference [and E1 at some point said that parameter exploration involves finding 'how many lags you need' or something like that, maybe that's a theory topic they research?]
* [E2](01:41:10) mentions that more weight choices in the range 0.9-1.0 would be nice
* [E2](01:42:50) "what is the meaning of the color"
* [E2](01:43:20) is reminded by M about the filter possibility in the PCP
* [E2](01:45:30) "if the eigenvalue difference is high, does that mean there is some structure ... should i select high ones or not .. what is the optimal choice"
* [E2](01:47:10) browses some lags that were filtered before, mumbles "yeah, not really [anything remarkable to see?]"
* [E2](01:49:20) [would you select lags in k2 differently than in k1 [collaborators said no]] "i don't know, maybe i should know the data/application better, now i'm just looking for high autocorrelations"
* [E2](01:51:00) asks around the computation time and parameter influence there

* [E2](01:51:40) [M asked why they used PCP filters the way they did] "just thought there might be structure when ... OTOH you have low value for diagonality and eigenvalues are relly different ... i'm not sure maybe that wasn't wise"
* [E3](01:07:50) mentions that more values in 0.9-1.0 would be cool b/c otherwise "the quadratic part tends to dominate"
* [E3](01:11:00) hasn't used PCP either, but managed quite well
* [E3](01:25:00)'s reasoning behind parameter selection: "some higher lags with high diff in eigenvalues", k2 lag selection should work the same
* [E3](01:48:00) makes new parametrization based on existing one "i'm interested to see what happens when i drop the quadratic part altogether"
* [E3](01:53:40) makes new parametrization based on existing one but with additional lags, which E3 directly puts in and asks for more R-like commands in there to resemble `seq()`, bit later uses that instead of clicking. short confusion around additional dimension when basing param. on existing. uses eigenvalue difference to filter lags.
* [E4](01:02:10) "everything makes sense but [the PCP] is a bit unfamiliar to me", but after explanations E4 interacts naturally with it
* [E4](01:07:00) wonders why lag 3000 is such an outlier and bad fit to 143 months [no idea]
* [E4](01:12:00) chooses 3 lags based on autocorrelation
* [E4](01:14:00) about k2 selection: "i'm not an expert ... maybe one shouldn't pick them acc. to linear autocorrelation ... is it possible to get a quadratic metric?"
* [E4](01:45:20) asks about new dimension, and to have a union filter in the PCP, and to select all filtered lags at once. some problems with selecting lags because not all pixels in an acf box have event handlers [or another reason, but hey prototype]. E4 selected default 1:12 plus the 11 monthly lags with highest autocorrelation. in k2 selection, E4 asked if direct lag input overwrites, and how to reset the default lags. E4 then resets the defaults, types 5 monthly lags from memory `21,42,63,84,105` and then adds the first 10 `1-10`
* [E4](01:49:30) asks what happens when a new result comes and all colors are used [nothing, but one could e.g. use a temporary color, not allow a new parametrization when selection is full or have a different process altogether where the new result is not suddenly pushed into your visualizations but rather you're informed first and have to actively include it - then at this point you could choose one color to replace, and it could help regarding change blindness]
* [E5](01:19:00) was "not very familiar" with PCP, but after explanation seemed to interact very naturally. selected 3 lags based on autocorrelation and eigenvalues. "this whole interface is very useful, very convenient". about different dimensions in k2: "depends on the data ... if it's real data, whatever the expert says ... at least partly depends on the application ... [finds out that autocorrelation dimension is again linear autocorrelation] maybe it would make sense to show a different measure ... i'm not that familiar with gsobi ... how the implementation is performed ... is there some nonlinearity unmixing going on in the functional? ... but if [the PCP] is from the viewpoint of classic BSS that only has linear measures of autocorrelation that [the PCP] is pretty relevant ... would be relevant to allow to select two lags sets separately ... not sure what would be a good measure for gsobi that would help k2 selection"

### Impossible Tasks

* [E3](01:49:20) get overview of existing parametrizations to LOCATE one with desired weight parameter

### Other

* [E1](00:33:36) used the colored boxes to unselect results
* [E1](01:09:24) spinner not visible because zoom windows / record button from local software hide it
* [E1](01:19:55) "i think it [selection limit of 5] is good because more it could be too much to ... [effectively analyze?]"
* [E5](01:50:00) thinks kurtosis is generally more interesting than skewness [which is the default]
* [E5](01:55:00) asks for a way to remove a single of the time series [b/c DKK is so dominating in the matrix], also mentions it's very nice to be able to go back and forth while keeping analysis state [a bit sad we forgot to tell earlier participants about this]
* [E5](02:03:00) mentions that it'd be possible to compare the BSS method to PCA "b/c the first, whitening step in many BSS methods is basically PCA, so seeing PCA output gives then information on what the (joint) diagonalization step does ... to have an idea of the importance of the two [whitening & rotation] steps [N: to get a look into the black boxy thing] yeah ... it reveals something about what the invariant coordinates are"
* [E5](02:14:50) suggests to plot factors next to cluster overview, points to special figures in their paper that he'd also use [that presumably plot the individual factors of a component]

### Dataset familiarity

* E1 used exrates before, but mentions around 1:55 that knowing the dataset more would be beneficial, and having more clear analysis goals
* E2 did not use exrates before, at least didn't mention it, works more on theoretical issues, mentioned a couple of times knowing data/purpose/domain better would be good
* E3 didn't answer
* E4 noted "at least some version of it" they has seen in the past

## Pre-study questions

### Experience in statistics/BSS

* [E1](00:00:00): studied math/statistics, did PhD about BSS but not teach, 7y in stats research with RStudio, but doesn't use BSS methods in current job
* [E2](00:00:30): teaching stats at uni for 20y, researching BSS at least 10y, supervised a PhD in BSS
* [E3](00:00:30): math major, stats minor during study, then did phd in bss. ~10y in stats/bss. taught statistics courses and bss and some signal processing (where E3 is postdoc).

* E4: studied statistics from 2009 for 9 years, bss phd finished a few years ago, afterwards bss-related postdoc in statistics: 10y as academic. postdoc: bss for non-temporal data.
* E5: math masters, studied bss afterwards like asymptotics and adapting existing methods, some applied work (where domain experts did interpretation), basically finished with phd

### Experience in visualization

* [E1](00:03:00): "cannot say i'm very experienced, maybe a little bit", "only basic things, no web apps", uses ggplot2 to show descriptive data stats and analysis results
* [E2](00:01:20): using a lot in teaching and research, but "just basics ... haven't done any ... rshiny apps", mostly static vis like scatterplots, histograms, time series, "basics". generates with R basic functions in RStudio.
* [E3](00:02:40): "usually looking at source time series ... sometimes real data like EEG/fMRI ... EEG: 128 components, ~100k time points"
* [E4](00:03:30): "use visualization ... a great way to show information ... don't know that much tools ... only what's in R like ggplot and base plots ... interesting topic but don't use it that much", "i usally have scatterplots ... line charts ... boxplots ... i tend to stick with these basic plots ... [M suggests heatmaps] that's a good point ... i like it ... neat and useful", uses Rstudio
* [E5](00:04:00): "some experience with R, nothing too fancy ... maybe something where i imported photographs to R, that'd be the most fancy vis thing i'd have done ... [how do you make figures] always in R ... more convenient than latex ... [in RStudio?] sometimes RStudio or CLI based R on a server ... [ggplot?] yeah i like to use ggplot whenever we makes the graphics for publications ... [frequent visualizations E5 uses?] histograms ... line charts ... some extreme value indices"

### How did they use BSS

* [E1](00:05:00): temporal data, "take lag data and count and see what structure emerges", also supervised DR for prediction
* [E2](00:02:45): "we're mainly doing ... theoretical research, so it's not really applied", often not real data, sometimes EMG data (brain imaging), "but mainly theoretical"
* E3 mainly EEG in the past, and phd in asymptotic properties. in current job BSS for graphs/networks.
* [E4](00:05:50): "[data?] temporal and non-temporal data [goals?] find something hidden ... latent groups (non-temporal) ... something that strikes the eye/interesting (temporal) ... no specific goal"
* [E5](00:06:50): [on dataset in their paper] "when using BSS as DR then 1st component always seems to be like a mean ... objective was that how much of the variance in the original curve can be explained with how many components ... for parameters we tried different lags and the one that fit best was the one we chose for the study [strategy to parameter selection?] starting from small values b/c in the domain’s data that if there are temporal dependences they should be stronger with small lags, we didn't go very far ... looked at the eigenvalues and tried to maximize the difference between the eigenvalues [of scatters of Z, that might be autocov matrices for AMUSE or the 2 scatters from gSOBI]... and looking at the curves"

### Did they need to explore BSS parameters for a dataset

* [E1](00:06:00): did some "testing which kind of combinations of lag parameters ... work best ... on simulated data ... see how many lags we need", (for parameters) used MD index to "see ... if you have too few lags"
* [E2](00:03:00): "not really", [how would you do it?] "visualization would be really important ... try different parameters ... see how sources [components] look like ... that would be the first thing i'd do just try different things" [how would you find params to try] "that's a very difficult question ... i don't really know ... maybe try something at random, see how it goes"
* [E3](00:06:00): "i've done it ... autocorrelation plots at different lags ... with EEG tried only very short term ... but not anymore a frequent part of work"
* [E4](00:07:00): "yes and also different [algorithms] ... some didn't have parameters, some did ... trying different lags and weights ...[how you went about it] find R package or code algorithm"
* E5: see previous question

### How did they analyse components (visually/numerically, sign & order problems)

* [E1](00:07:40): "don't remember ... use eigenvalues to sort"
* [E2](00:05:20): "you have to know the application [dataset, provenance, domain N assumes] quite well to know what you are really looking for", some methods have inherent ordering like FOBI, with BSS there's ways [E2 couldn't recall], mainly the approach is to "plot all sources at once, see if there's structure", with brain data it's "easy" because you can link a component to a spatial area [rstudio shortcomings] brain datasets are too large for R, with small datasets it's "handy, no problems with that"
* [E3](00:03:50): "plot original source i'm mixing and then I plot on same figure found components and see if they match"
* [E3](00:07:10): "[with eeg] look for something else than white noise ... periodic signals ... or other time dependence ... then they are more interesting ... mostly i just look at the figures"
* [E4](00:08:30): "very little logic in [parameter selection] ... comparison done mostly visually ... sometimes there were objectives like [finding ground truth] ... mostly we picked the ones that looked like they contain most information ... parameter values i'm not sure if there was any strategy ... trial and error [what about sign] you kind a learn to see inverses, usually it was no concern ... something you get accustomed to [and ordering?] some [algorithms] have a natural way to sort components like highest autocovariance at a lag ... [choice of ordering strategy] influenced the rest of the analysis quite a lot because we only looked at the first ones"
* [E5](00:11:00): "order was achieved nicely by amount of variation explained ... and eigenvalue difference [and sign?] not an issue anywhere we had this clustering approach ... were able to cluster the age groups ... sign identification did not play any role ... [and in visual analysis you just know to look out for it?] yeah"

### Is this parameter exploration important for a practicioner and/or a frequent part of your work?

* [E1](00:08:40): "ya, i think ... it's a good idea to see how they're ordered" [N was meaning to refer to param. exploration in general, but the way the question was phrased E1 understood it to refer to only the ordering question prior], E1 didn't have the task much in the past or present
* [E2](00:09:30): "[trying random and explore] is maybe not the best ... would be nice to have ... clear instructions ... there are so many algorithms ... [N: and if there is just one?] ... then instructions for parameter selection would be nice"
* [E3](00:10:30): "i think so ... also trying parameters is some kind of robustness check to see that results do not depend too much on particular parameters ... so that they are more reliable"
* [E4](00:11:10): "more frequent in the past when i was a phd student ... maybe now ... someone else typically does it, but i still do it, but more a couple of years back [good task for a practicioner] very important ... when you have lots of parameters you should compare them, and i know from experience that the result changes quite a lot ... would be nice to ... explore more than with just trial and error"
* [E5](00:13:10): "yeah very frequent in my work"

## Post-study questions / more general feedback

* [E1](01:56:00) thinks that prototype is useful for comparing different parametrizations
* [E1](01:56:20): DKK was surprising and the two equal results with different parameters [that N mentioned explicitly before]
* [E1](01:58:20): "good in way to see how ... lag choices can .... affect the results and how ... much they don't", "changing this k2 lags ... really gives much different results compared to k1 so it's something worth to look at more"
* E1 wants to explore more on their own time

### what do you think of the prototype

* E1: "something i enjoy playing with and ... i like that ... all these things are connected ... and you see which component is where ... and more DOI functions would be nice ... but can't think of any right now ... in unmixing matrices it's very easy to see where are the important values ... and [the correlation slope] i found most interesting to see how components correlate with each other ... and that they are not always aligned ... and also that there's options [to compare in detail] ... like [superimpose columns] ... and the clustering is also ... a nice addition, but i need to familiarize myself with it more ... it looks really nice ... after a little bit of using it's easy to use after the introduction"
* E2: "really nice to play with this ... a few plots were difficult to understand at first but you explained them well ... still not sure about [projections], i understand the location [which is without semantics, but hey] but not the size ... [N asks if difficulty is in interpretation or knowing which space is shown] yeah", "easy to use ... i especially like [correlation graph]", "one thing we always wonder ... how can we say which result is the best? is there anything? [N explains] yeah hmmm [not sure if satisfied with answer]", "one has to know gsobi quite well ... some more explanations would be needed [around whats k1/k2 etc.]", "possibly it would help to know the application to select lags ... i don't know", "how do you upload a dataset in the prototype ... [exporting a parametrization to R] would be really nice"

* [E3](01:58:00) "looks nice ... nice to have this kind of tool to look at results ... much faster than typing in R to get these comparisons"
* E4 "very cool ... enable things i would not have possible in base R ...[the tasks of comparing a complete picture of the parametrizations] would be very inconvenient in R ... i would never have something like this [the prototype or correlation slope?] there so this is very nice ... very visual in a good way ... after explanations it was relatively easy to use ... the [projections] were something where i spent the most time [understanding] but when i got it i think it's very useful ... not sure if i would change it, i get it now but it took some time ... otherwise very intuitive ... along the way we had some [things that could be improved] like selecting multiple lags at once etc. but otherwise it's very nice ... [in between next question] the constrained clustering was also something that took some time in the beginning ... but when you think about it it makes sense and it's very useful"
* E5: "i like it very much ... this is an absolute time saver ... this would be very useful if you have an applied example ... i sometimes do these massive computations where i compute all possible parametrizations [below a selected lag] and pick the best one according to some criteria [e.g. explained variance] and that's not very time efficient ... i like [the PCP / lag selection] a lot ... i'm also suprised it's a very functioning prototype, not sure you should call it a prototype anymore, everything is working quite perfectly"

### something very easy / useful

* E2: "nothing really difficult ... maybe the lag selection in the beginning ... otherwise very easy to use"
* E3: "matrix diagonality is nice feature ... [bad connection] ... that you can do this model checking in a time series context [N asks for additional vis/data to look at for this task?] trying more parametrizations and different lags", "showing the correlations ... easier than looking at correlation matrix"
* E4: "i think it's very intuitive [that all time series behave the same way] ... it's very consistent ... ofc you need to know what you're doing, if i haven't ever used [gsobi/bss] then it would be difficult"
* E5: "the component comparison ... convenient and fast ... but i'd like to see more explaining stuff, i didn't originally understand [the projections] but now they seem more clear ... also [the scatter diagonality plots] is something i don't usually have the time and energy to compute, this is very interesting, might be useful"

### something not useful / too confusing

* E1: "didn't know how to read [the model comparison] ... haven't thought of this much before", "is there a way to export the results?"
* E1: lag granularity was "unclear ... kind of understood ... but maybe not that much"
* E2: small bugs in lag selection
* E3: better direct lag input, "haven't done this kind of applied analysis lately", the clustering E3 is not sure how to use it but might still be useful

* E4: "clustering and projections were conceptually challenging but from a UI point of view they're very natural, in BSS model tab "i don't know how to interpret all of these things"
* E5: "projections are fine if you explain it somewhere, just the dots are not clear immediately [N points to clustering, E5 is a bit hesitant, clicks around a lot, is then reminded of purpose (find stable component)] oh yeah that's definitely something you should do at some point
* E5: [N pointed at lag granularity] "yeah maybe it's not clear enough what this does [N: to control how fine-grained lags are compared] yeah but we still have so few [in the current view between 5 and 19] [...so you can just look at them]" - a response would be that in the particular view E5 saw it wasn't obvious that in the gray part of blue result, 2244 and 2257 are just 13 lags apart, ie very close considering the whole space. upping the granularity to some number guarantees that lags are shown equally-distant according to the buckets. (it does not guarantee that shown lags are at least this much apart, which might be easier to understand, but then harder to compare lags between results. anyways we've learned also that BSS people tend to choose only few lags because of computation time and convergence.)

### did you miss anything, something that was not there?

* E1: "not really ... [labeling of k1/k2] might be confusing for some, to know which part [quadratic or linear] is which ... otherwise not really"
* E2: "no, nothing really"
* E3: "can't think of anything at the moment ... maybe if it's easier to go back to input data [N: to compare with components or just look at?] not sure, sometimes you can see from the unmixing matrix that there is only one non-zero element [M suggests easy possibility to go back to first window, which actually works but N forgot to mention] yeah that would be [sufficient]"
* E4: looking at sources again in explore view, stuff in the lag selection (select all at once, better direct input interface), have a quadratic metric for the quadratic part "whatever that is"

### opposite: something to remove?

* E3: "nothing really"

### how does prototype compare to rstudio

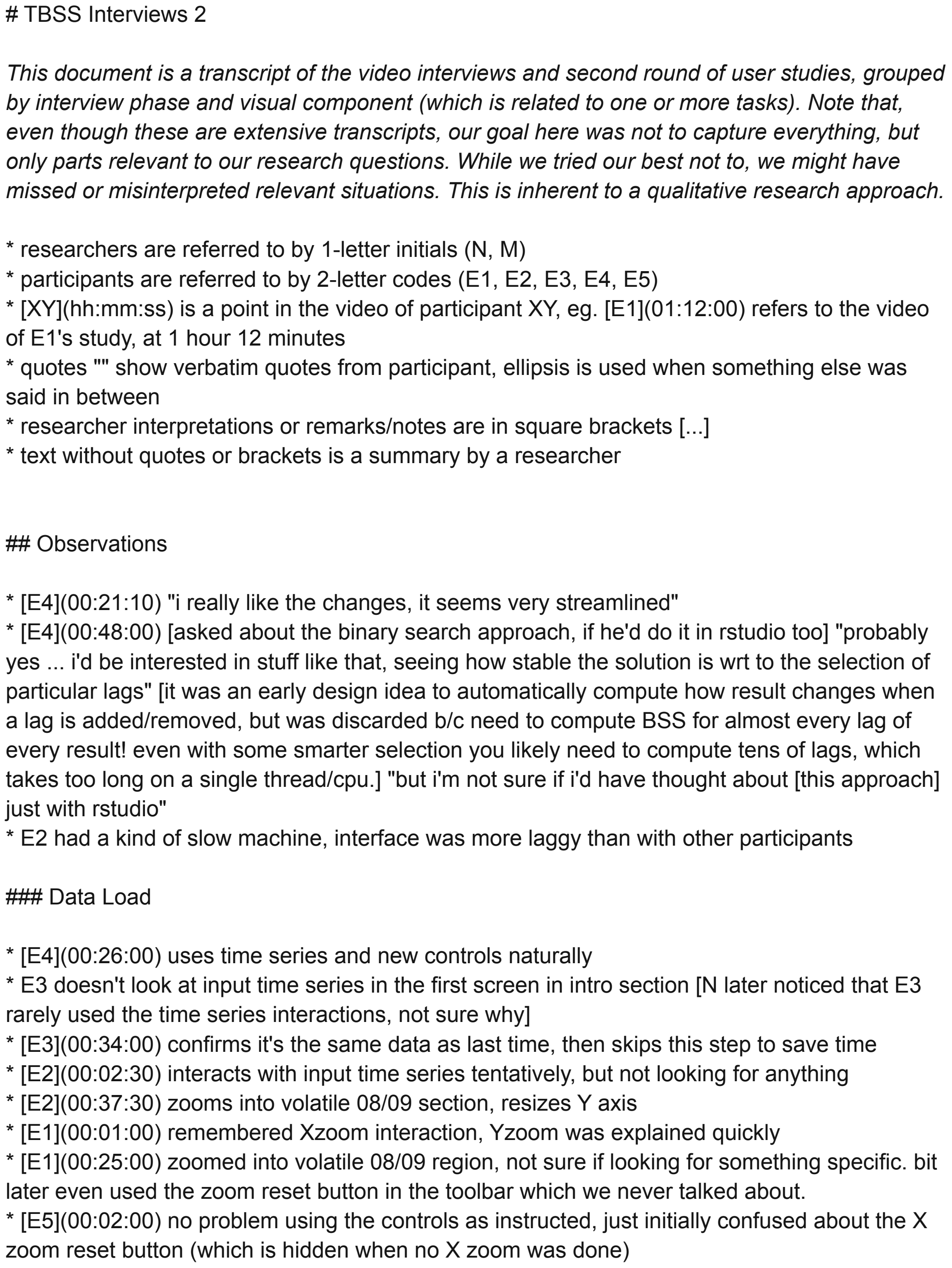
* E1: "Rstudio it's like ... copy/paste this and run again ... and it's harder there to compare results side by side, prototype is good for that ... that's a very valuable thing", "in Rstudio you couldn't do [the interactive 2d projections] and see how it's connected ... i think that's great", there's more structure in the process because "immediately you can ... see how [a result] compares to other results, to other choices"
* E2: "[prototype] does almost the same things ... [matrix vis] is very nice ... and seeing all components at once and the comparisons ... and [interactions with time series] that's really nice", E2 would plot components next to each other also in RStudio but doesn't know if it works with that many time series
* E3: "from my perspective ... not so big difference ...but for someone [inaudible]", "when i would have the data/code then it would be not so big difference ... but having something like [the

prototype] available which would make it kind of simple for any new data that i have maybe it'd save a lot of time"
* E4: [see wdyt part]
* E5: "in some things [the prototype] would be a huge time saver, esp. if you're doing applied work ... i [focus] on theoretic work, so i have simulated data and stuff like that so ... and i use methods that are not really available online ... but for someone doing applied work it would save a lot of time, esp. if they're not so comfortable in R programming ... to get the basic diagnostics done like finding stable components and testing many different lag sets and doing the comparison ... to see how sensitive your analysis is on the lag set ... if this was available i would definitely ... tell some master students to use this"

### how to improve prototype more

* E1 was too tired at this point, so we settled on email
* E2: "[e.g. with more DOI functions] i don't know, [skewness/kurtosis] are quite alright", more explanations
* E4: show all different parametrizations [in a table or something]
* E5: "i don't ... i like this a lot ... really advanced work", "this framework is really nice ... maybe it would be nice to increase the availbility to allow upload/import of data that might not come from gsobi" [like a method-agnostic data exchange format and analysts could export from R and import to the prototype]", again points at suspicious values of MD index

Second Interview



# TBSS Interviews 2

This document is a transcript of the video interviews and second round of user studies, grouped by interview phase and visual component (which is related to one or more tasks). Note that, even though these are extensive transcripts, our goal here was not to capture everything, but only parts relevant to our research questions. While we tried our best not to, we might have missed or misinterpreted relevant situations. This is inherent to a qualitative research approach.

* researchers are referred to by 1-letter initials (N, M)
* participants are referred to by 2-letter codes (E1, E2, E3, E4, E5)
* [XY](hh:mm:ss) is a point in the video of participant XY, eg. [E1](01:12:00) refers to the video of E1's study, at 1 hour 12 minutes
* quotes "" show verbatim quotes from participant, ellipsis is used when something else was said in between
* researcher interpretations or remarks/notes are in square brackets [...]
* text without quotes or brackets is a summary by a researcher

## Observations

* [E4](00:21:10) "i really like the changes, it seems very streamlined"
* [E4](00:48:00) [asked about the binary search approach, if he'd do it in rstudio too] "probably yes ... i'd be interested in stuff like that, seeing how stable the solution is wrt to the selection of particular lags" [it was an early design idea to automatically compute how result changes when a lag is added/removed, but was discarded b/c need to compute BSS for almost every lag of every result! even with some smarter selection you likely need to compute tens of lags, which takes too long on a single thread/cpu.] "but i'm not sure if i'd have thought about [this approach] just with rstudio"
* E2 had a kind of slow machine, interface was more laggy than with other participants

### Data Load

* [E4](00:26:00) uses time series and new controls naturally
* E3 doesn't look at input time series in the first screen in intro section [N later noticed that E3 rarely used the time series interactions, not sure why]
* [E3](00:34:00) confirms it's the same data as last time, then skips this step to save time
* [E2](00:02:30) interacts with input time series tentatively, but not looking for anything
* [E2](00:37:30) zooms into volatile 08/09 section, resizes Y axis
* [E1](00:01:00) remembered Xzoom interaction, Yzoom was explained quickly
* [E1](00:25:00) zoomed into volatile 08/09 region, not sure if looking for something specific. bit later even used the zoom reset button in the toolbar which we never talked about.
* [E5](00:02:00) no problem using the controls as instructed, just initially confused about the X zoom reset button (which is hidden when no X zoom was done)

* [E5](00:32:30) "i have no idea what half of these currencies are". explores a bit, notes spike in TRY, confirms the year, mentions it's "probably when they brought in the new currency" [not 100% correct: "On February 22th, 2001, the government allows the lira to float freely.", but cool that E5 made hypotheses at this point already], "we will probably see this spike in the components"

### Explore

* [E3](00:38:00) while waiting for a result, N asked if timing (how long did it take to compute result) would be interesting as well? "one could also use number of iterations" [but E3 doesn't confirm it's smt interesting] the result appears in the meantime but E3 didn't notice it happening.
* [E2](00:38:00) "so what's the idea" [seems lost as to what to look for]
* [E5](00:35:20) wants to see the sample size (ie., how many time steps do the series have)
* [E5](00:52:10) tries to distribute the orange color but that doesn't work

#### Table

* [E4](00:04:40) visual summaries made sense, encoding of lag sets is "very handy"
* [E4](00:27:00) selects [without being told how, actually] 4 exsting converging methods for inspection and looks at lags in more detail.
* [E4](00:29:10) tries add 5th selection, which is not possible but wasn't explained either
* [E3](00:07:00) [some discussions around the two 'weight' concepts]
* [E3](00:35:00) uses table to get an overview of used lags and hovers a lot over the summaries to see the actual numbers. notes that one method didn't converge, and wants to "see if a small change [to the parameters] changes that"
* [E3](00:39:00) notices that we don't use a linear scale in the weight parameter - oops. N did this change to give more space to what was often called the interesting range of `b`, 0.8-1.0, but then forgot about it. due to small labels and .8 marker almost in center, nobody noticed. [seemed not like a huge problem tho]
* [E2](00:05:00) nods and "hm"s when N explains the table [seems like all clear]. asks how one can put own data in [which E2 also asked last time so it could be they're actually eager to try it with own data they know]. task to compare lag sets, N framed badly, E2 tries to compare in detail (single lag resolution) from histograms instead of high-level differences (uniform/skewed selection, more short/long lags), and gets confused about the encoding. M explains it's a histogram and meant to provide an overview.
* [E2](00:38:40) asks, as in 1st study, what k1 and k2 is (which one is SOBI), N points to tooltips. E2 hovers a lot in the histograms [as if looking for something], N points to parameter comparison [E2 doesn't seem satisfied], N confirms if E2 is in fact searching, but "just comparing" [seems like a better strategy for E2 would've been to identify 1-4 methods, then select and view them in parameter selection]. E2 selects 4 and looks at other views.
* [E2](00:41:20) has green method selected and wants to unselect. E2 clicks, and in fact unselected the method, but since green is now the first free color, the row stays green as there's no visual difference between selection and hover state. [E2 seems confused about whether the action worked.]

* [E1](00:05:00) solved first task well. asked to describe difference between two lag sets (one with uniform distributionm one with only short lags), got that also right but tended to hover a lot like E2 and E3 :/
* [E1](00:11:10) collapsed table
* [E1](00:25:40) correctly identifies that from 5 initial results 4 converged
* [E1](00:27:30) selects 3 param.s for closer inspection (only k1, only k2, both k1/k2). accidentally changes desired color as E1 misselects one (blue), selects the correct one (purple), deselects the wrong one (blue freed up).
* [E1](00:39:50) collapsed table earlier, like 10m or so, now forgot that it's there.
* [E1](00:50:20) downloaded a method, opened it. "awesome"
* [E5](00:06:00) didn't get at first that background color encodes success, but after N explained a second time, it was fine ("ahhh i see")
* [E5](00:34:30) selects all 4 converging methods to compare parameters

#### 2D Result Projections

* [E4](00:08:23) [possibly tried to compare similarity of 1st and 2nd components by shape of plot, ie., do points go nearer or farther]
* [E4](00:31:00) clicks through n-th component projections to learn how *all* components of the methods compare [there are better ways, like the "all components" projection, or the component comparison]
* [E3](00:09:00) didn't use the projections much in introduction, but also didn't ask about them: "i think it's all clear" [N felt it wasn't]
* [E2](00:13:00) N chooses not to explain in too much detail. [N: this view wasn't popular among other participants, so it would probably take much time for something that won't be used. ofc then E2 is even more likely to not use it, but in the moment it seemed reasonable. a better alternative would have been to explain it not in technical what-is-it terms but rather look-for-this-then-do-that instructions.]
* [E2](00:40:20) notices red and purple have similar components "i'm surprised that they're so similar", then collapses this view to make space for comparison section
* [E2](00:57:00) compares component projection of old and new result, concludes they're the same b/c points are next to each other [size encoding tells different story, but we didn't explain that in the introduction for time reasons]
* [E1](00:07:00) [we introduced it but skipped tasks as E1 was nodding a lot and using it meanwhile, seemed like it was clear]
* [E1](00:26:00) used component projection to investigate two points near each other, [presumably, can't really tell from zoom recording] looked at table how they compare in terms of parameters
* [E5](00:09:00) N rehashes quickly how they work, asks if that makes sense, E5 "yes", moving on
* [E5](00:34:50) uses component similarity, sees that red and green are very similar "although the red doesn't use the linear part"

#### Component Overview

* [E4](00:33:25) tries to find if blue finds similar components than red and green, but ofc it's not working well here as no attribute is preattentive. for that, N reminds of changing cluster size and ordering later.
* [E4](00:47:00) again clicks through n-component projections, then realizing it's the same information as one row of the slope graph.
* [E3](00:14:30) after introduction asks "how did you define 'stable'" [maybe it wasn't clear that many dark bars mean all similar components, which means (with minimal cluster amount setting) that this component was found in all results, and therefore is what they called 'stable']
* [E3](00:39:30) explored components, opened first cluster, noticed one that didn't quite fit to the others, then proceeds to component comparison for more detailed inspection of the result
* [E2](00:15:30) "so [a stable component] means that i can find this component in all results?" yes
* [E1](00:09:40) found stable and unstable easily [N didn't point out how to do same task with different cluster number]
* [E1](00:26:30) used component overview to get more detailed picture than similarity projection, ie., which components do the 4 methods agree on, which not
* [E1](00:28:00) has 3 selected, checks which color disagrees on most components [cool idea, but from vis perspective would be cooler to start with empty selection, hover around table or elsewhere and assign 1-2 colors to most disagreeing methods. reason being that disagreeing components are very light in rank histo, and their color hue is not well perceptable.] unclear about hidden context-dependent action on time series label ("can i flip [the component] here?"), tries it out (deselect, select) and accidentally changed color to blue b/c that was free and preferred.
* [E1](00:45:00) looks at where green and purple bars are in each row to compare how they differ [would have been better to deselect the other colors]. N points to component comparison.
* [E5](00:11:30) task "find components that don't agree much", E5 looks around the similarity projections, but seems to stem from ambiguous wording by N. after that's cleared up, E5 finds a non-agreeing component quickly "by looking at the histogram" (mostly using position, N reminds that opacity is another way to tell)
* [E5](00:36:30) inspects first component, realizes it's the same spike as in TRY, goes to matrix view to verify
* [E5](00:49:00) asks how to interpret rank histogram

#### Parameter Comparison

* [E4](00:27:00) wants to see actual histogram, but forgets changing bin width is possible. un-hides empty bins such that k1 and k2 are aligned, increases bin size by 1 to 43, when the whole thing fits into their screen [possibly add option to auto-fit to screen?].
* [E4](00:30:00) compares distribution of lags in both spaces, while relating them to the projection view. as it's not clear when the square width can be compared, E4 wrongly compares between "all components" and "n-th component", which have different metrics.
* [E4](00:33:00) switches between bin sizes, accurately compares lags

* [E3](00:18:30) N: does the lag set histogram make sense? (pause) E3: yeah... ["i guess"] [N felt nonetheless that E3 'got' it after being introduced to the hidden bins]
* [E2](00:18:00) frowny face :/ changes bin size, but bars don't change height b/c the selected lags are few and far apart [seems to confuse E2]. also un-hiding empty bins with bin size (a few) takes long enough to trigger evaluation gulf, and E2 clicks again.
* [E1](00:11:20) used everything while N explained, so we skipped tasks
* [E1](00:29:15) points out lags are very different between red and green and b/c of bin size 1 one shouldn't show all bins as it'll get super wide. immediately recognized long lags too. not sure if E1 compared bar lengths between k1 and k2, which is not allowed.
* [E5](00:14:30) seemed to understand new lag set vis just fine
* [E5](00:43:00) removes blue param. because "i don't think it's very realistic in this currency dataset to look at linear dependencies (...) very long time into the history". E5 then wants to change green param. which E5 thinks is a more realistic lag set.

#### BSS Model

* [E4](00:16:20) [discussion about 4th CC scatters] "they are nice to have there, but yeah i agree [that they're maybe not useful to see when only SOBI was used]
* [E4](00:39:00) [checks 4thcc] "somehow the smallest lags were most important" [not sure what E4 means]. [concludes that there are no large differences that would be worth investigating.]
* [E3](00:21:30) uses scatter diagonality views correctly, doesn't comment on it
* [E3](00:48:00) superimposes scatters, concludes no big difference, but would like to see used lags at the same time [doesn't work]. N tells to juxtapose, but lags are very badly visible b/c spread far apart and thin due to lag bin size [maybe ~10 would be a better default, but that's confusing in other ways]. E3 superimposes again and zooms in, concludes they are similar "which is interesting b/c one doesn't use the k1 lags at all, maybe there should be some difference". E3 changes colors, notices differences in 4th cumulant "makes sense that blue one is lower b/c it only uses k2".
* [E2](00:19:40) asks what the "lag" label means, N explains scatter diagonality per lag is shown. E2 asks what's it mean when diagonality is very high, "the corresponding lag is not useful? can you say it like that?", N points out that one also needs to verify the parameters, then moves on [should have explained more, in retrospect, but it was probs because of time]
* [E2](00:42:00) the view is too wide for the screen and E2 doesn't notice that a portion of lags is cut off.
* [E2](00:42:50) unclear about interpretation of scatter diagonality "so these lags we shouldn't use?" N explains how this view can be used to find new lags to try, E2: "hmm" [seemed not convinced]
* [E1](00:13:30) again used everything while N explained, skipped tasks
* [E1](00:30:20) compares scatters, notes again green lags are "all over the place". zooms into range 1–180 to find differences.
* [E5](00:17:00) seemed to understand views. task "how do autocov. matrices compare", E5 mentioned "they're quite similar even though their parameters [lag sets] aren't", then notes that red param. does not use k1 lags, so not super-clear if one should even be looking at "k1 diagonality". [N did discuss this matter with collaborators via email at some point, but we feared

hiding such situations by default might be too restrictive, and they mentioned that if we do it there should be the option to see it regardless, so we just left it.]
* [E5](00:53:40) "[autocovariance matrices diagonality] look realistic, nothing too interesting going on"

#### Matrix comparison

* [E4](00:36:10) [after reminder] "i would look for black rows or columns", finds USD and HKD inputs (practically same), and one component that's "not influenced by a single input"
* [E4](00:45:00) first uses this view and MD-index to check if new result changed compared to others. notices most influential inputs are the similar, but overall it seems better separated (less gray squares).
* [E4](00:49:00) again first uses this view and MD-index to check if new result changed compared to others. notices it does, and the lag 4 in fact changes something "but i cannot say what"
* [E3](00:23:00) doesn't notice the time series are shown underneath matrix because smh it's exactly below the fold, and didn't notice the color change in labels of the matrix either (which tell where something is selected)
* [E3](00:43:20) uses scaling checkbox to get rid of DKK column, then notices USD and HKD have high values in many components. N asks how would E3 figure what's going on [leading up to JUE2 LOOK AT THEM], E3 mentions there might be a connection between the two but quickly redacts that statement with "mh, well, idk", then points to components with single high factors. N reminds that viewing inputs/components here is now possible, E3 selects PLN and RUB to check, observing that they look very similar to the components, thereby confirming a high influence. discussion returns to USD/HKD, N again suggests to look at them, E3 would like to *just* see the inputs without the components, in the end notices USD and HKD are the very similar but doesn't conclude anything from here: "somehow they are present in very many [components]"
* [E2](00:22:50) some confusion around row/col encoding [seems otherwise clear]
* [E2](00:44:10) labels seem kaputt in E2's browser, a bit too large and dense. E2 just quickly looks at matrices, unsure if at MD heatmap too, clicks in a DKK column, moves on. N brings E2 back at (00:46:10) to nudge E2 to investigate DKK by scaling factors. E2: "i wonder why [scaling inputs] affects the result so much" [reasoning probs was that BSS is scale invariant and scaling inputs shouldn't affect anything, so maybe N should have explained in more detail how this works]. N nudges E2 to compare value range of DKK and another column, but E2 has difficulties increasing Y axis. E2: "yeah i can see the scale is different but still this method should be scale invariant" N: "but wouldn't factors be different when using scaled or unscaled inputs?" [E2 seems unsure and moves on]
* [E2](00:57:50) compares old and new result, but skips MD index heatmap and goes straight to matrix encodings. reorders matrices so old and new are side by side. scales input. [different patterns emerge, one has less gray cells] N: "how would you say did the results change with the added lags?" E2: "not really, i wouldn't say so" [maybe b/c the same cells were black?] N wants to discuss but E2 moved on.
* [E1](00:15:00) skipped tasks, seemed clear

* [E1](00:32:30) wanted to see parameters also here, had to go back to that tab to see which was the one with b=0. [could have un-collapsed the table too]. notes DKK has high factors everywhere and USD too in most. [N points to scaling factors] "now it seems completely different" [N asks how to investigate HKD and USD which have high factors in many components] E1 selects them in common row, "they seem similar" [E1 clicks through components, maybe to see if any has HKD/USD features?] [N asks what to do when 2 inputs are ~same] "check how they correlate to each other ... maybe i'd check that a bit more [to see] what's going on there" [in R, I'd assume]
* [E5](00:21:30) no problems or comments
* [E5](00:36:50) scales factors, confirms that 1st component is mostly influenced by TRY, goes on to compare factors and components. e.g., E5 notes that in the linear-only param., TRY doesn't "dominate" anymore. the scaling functionality is "good to have b/c otherwise it looked like DKK dominated everything ... a very good thing". N wants to point to fact that USD and HKD are the same series, by asking "do you see other interesting pattern". E5 asks where to see the DOI values, N explains, E5 mentions he'll not consider components after the 4th "because values are so small", then goes scanning the first four rows of matrices and says E5 doesn't find any [the HKD/USD shizzle starts at the 6th in the blue result].

#### Component Comparison

* [E4](00:32:00) [after being told] investigates components of two results with slope graph, sees that they're the same: "it's weird [because red uses only some k2 lags and green has more weight (both meanings) on k1]
* [E4](00:49:20) [on its own] checks how components between new and old results compare with slope graph
* [E3](00:25:00) doesn't comment the introduction whatsoever
* [E3](00:40:20) adds/removes different methods in search for one specific component that was discovered earlier in the overview. [would be easier to go back to the overview and select the result from there, but E3 collapsed the overview and then probs forgot that it existed]
* [E3](00:41:00) after the search session before, ended up with a suboptimal color combination of green and blue (hues too similar to distinguish them in crowded images). superimposes, looks at sparklines of 1st component, "it's very similar", wants to confirm with correlation slope (0.7x correlation), then doesn't find the differences due to the mentioned color issue and small Y axis of the sparklines. N tries to solve the latter issue but no avail for a few minutes. E3 may also have thought that the sparklines encode correlation of the currently visible time range [N thought about this in the designs, but then for consistency other views would need to work the same, which then required too much computation]?
* [E2](00:25:40) groks slope view well, as in last session
* [E2](00:45:30) uses slope graph to verify that red and purple are same, tries superimposition [but then seems lost again what to do]
* [E2](00:58:10) uses slope graph again to compare old and new result. some components are same (9/23: 1-5, 8-10, 16), but others not. E2: "some changes but not much differences"
* [E1](00:17:30) skipped tasks, seemed clear

* [E1](00:36:40) uses correlation slope, "blue and green really don't have many similar components" [few thin lines]. E1 investigates red 13 and green 9 "they seem somewhat similar" [0.71 correlation] E1 flips green 9 to see if it better aligns with red features.
* [E1](00:45:40) compares old and new components with slope graph [after N's suggestion]. verdict: somewhat similar, but not equal, new still very dissimilar to the one we compared the old one to. "you have to be careful with the parametres because results really can change a lot"
* [E5](00:24:00) no questions, comments or problems as far as we could tell
* [E5](00:45:00) compares a result with same result plus one lag, mentions there's not much difference [in the first four components which E5 defined before as interesting, after those there are changes.]. changes sign, removes some other results, superimposes etc. mentions that at this point one would probably need to know stuff about currencies to tell which of the two is more realistic.

#### Toolbar

* [E4](00:20:30) [change order by d&d] "very nice ... very handy"
* [E4](00:31:55) reorders results easily
* [E1](00:19:00) also
* [E1](00:37:00) again
* E5 also... seems that really worked for everyone

### Parameter Selection

* [E4](00:24:00) asks for possibility to remove all selected lags at once [would work with the explicit seq-like syntax: -1:300, but we didn't implement that]
* [E4](00:41:00) wants to take one of the two same results and change lags, see how it affects result. doesn't use "refine" function at first b/c thought that it would overwrite the existing result. adds k2 lags 4-10, doesn't use interactive views. "if i didn't know what i wanted, i could use those." reasoning: if all information was in 1-3, result wouldn't change with 4-10 added.
* [E4](00:47:00) refines exsisting, removes a few lags without problems
* [E3](00:29:30) [N explains whole view] "mhm"
* [E3](00:36:20) changes weight to 0.5, then wonders why 2 dimensions are missing [because they're computed with W, which we don't have]
* [E3](00:51:20) sets weight to 0.98, dynamic marker is then close to 1 and it looks like 0.981, which was somewhat confusing [different color would help, otoh the selected value is printed in 2 other places.]
* [E3](00:52:00) doesn't use filter buttons and works in day resolution, tries to do an OR or UNION filter. ends up with 3 lags, remarks that they have "high autocov eigenvalues" [but the data range there is like 0.0001 to 0.0005, so not that high actually]. selects one, doesn't see it's selected b/c lag set is behind zoom gallery view. ends up selecting top 3 for eigenvalue and autocorr. metrics for k1, and top 2 [guess E3 wanted 3, but bugs] for autocorr. for k2.
* [E2](00:27:40) "this was the tricky part also last time", some bugs with lag selection (clicking ACF lines doesn't work). "N: could you select any 2 lags? E2: ... no?" [struggling with unfamiliar visualizations and interactivity here]

* [E2](00:50:30) refines parameters. when looking at lag selection "mhm..." E2 uses PCP correctly, but there's too much data and it's a bit fiddly (and the ACF plot doesn't show stuff), and also slow b/c we wait a long time for render updates. we finally switch to Monthly lags. E2 selects lower 4th CC diagonality b/c interested in "better separation" [unsure what E2 means. probs idea is that a matrix that separates inputs better, also has more diagonal scatters?]. E2 ends up adding 2 largest k2 lags with high scatter diagonality.
* [E1](00:21:00) used PCP quite well while N reiterated how it works, also hovered around the MACF and looked at things [not sure what]
* [E1](00:40:50) [wants to add short lags to a method] weight stays same. in k1 selects a whole bunch (~1000) of lags in the middle [not sure why], immediately changes to first ~350 daily lags. too much stuff, filters to weekly and looking at maybe first 50 weeks. adds first 2 weeks (5, 10) to lag set, skips k2. no change blindness when new result came.
* [E5](00:25:50) says they "recall the picture" and starts brushing right away [cool] in the daily resolution [bad] which freezes the interface. N points to filter possibilites, E5 filters to monthly, N explains rest and E5 chooses 3 lags without further problems. E5 asks if changing filter affects selected lags [no]. E5 "i think this is quite intuitive"
* [E5](00:43:20) refines existing, leaves weight the same, uses direct input to add lag 365, which E5 thinks corresponds to a year [it doesn't though]

### Discussion

#### What do you think about the prototype

* [E4](00:50:40) "could be [a learning effect], but this feels more streamlined to me, it was easier to use", "i was not missing [the clustering] [in the component comparison]", "smooth to use"
* E4, new scatter type: "thought it would be nice to have a k2 counterpart and here it is ... specialized b/c few people know it and less people understand it, not sure what i can conclude from here, but it's logical to have it", maybe some some alternatives to 4th CC exist, but we're all not sure
* E4, matrix update: works as expected, just maybe also SHOW a whole row/column?
* E4, table: "nice way of presenting, i like the histograms and weight ... makes a lot of sense", confusion if method names are color codes, and reason for showing
* [E3](00:58:20) "nice changes to last version ... like [scaling W factors], also sign change ... so you can better compare it to other components. [like i said last time] nice prototype, it's interesting to analyse the data with this"
* E3, power scale with b: "maybe not a bad idea, was just difficult to see"
* E3, similarity proj.: "i don't know the MDS thing so well... I understand the idea, but idk what to make of the [MDS coordinates]"
* E3, new scatters: "maybe [i was] not [missing it before], but ofc interesting to include it as well, not just the autocov.... i think it's a good addition"
* [E2](00:59:20): "easier to use than before, really nice improvements", "really much better than the previous one"
* E2, component overview: "wondering how to use these to select new lag sets", N explains it's not primary purpose but one could look for outliers and form new ideas about parametrizations,

E2 "but now we have a bunch of results and you still don't know which is the best one", N explains a possible sequence of checks and comparisons through the interface, E2 mentions asymptotic variances and to use the result which gives smallest such variances "but maybe that's computationally really heavy", N mentions that one can download data to do specialized analysis, E2 [points to cluster of 3 results in component projection] "that's really nice if you have these results, that you prefer this kind of lag selection, i think that would be nice ... to know here" [pointing out that more guidance how to use component projection would be good]
* E2, downloadable results: "that's really nice"
* [E1](00:51:30) "many changes i didn't think of them but when i saw them i thought this is nice, so that's good", "i like the table for the immediate overview, it's in a good order", "here [in component overview] it's a bit difficult [to remember the time series interactoins]", "you can see easily in the histogram when you have low and high lags", "[matrix visualizations] is a nice view to see how different they are", "and that you [can see the components]", "the reordering thing is one that i didn't miss last time but i really think it's a good thing, it's very simple ... very convenient"
* [E5](00:54:30) "quite easy to use, last time i used it [some time] ago (...) and still i feel i can use most of the stuff, [therefore] it's somewhat intuitive, that's a good thing"

#### Difficult stuff

* E4: ordering in component overview wasn't clear, projections are good but seeing a different scale not in the projection but in the legend was not intuitive at first
* E3: [seeing lags already in the table] [N: i guess we could add them to the tooltip, but that has limits as well] "but maybe it's not even something i would recommend to do" [N: the underlying problem was that he'd have liked to see parameters together with the scatter diagonality, but they're already there, just with a very fine default]
* E2: which direction is the good direction for different metrics (e.g. scatter diagonality)?, lag selection in general
* E1: "can't remember, always asked when there were issues"
* E1, scatter diagonality: not intuitive what to do with it, didn't recall it. N reiterates how we thought they could be used. N: "does that make sense? did we get something wrong?" "[more scatters diagonal = better] makes sense" [but E1 doesn't seem *very* sure about it]
* E5: "i feel i didn't run into any problems" [N pulls notes]
* E5, sample size: important piece of information to see if large lags make sense, at some point estimations become unreliable
* E5, hide/minimize all components after N-th: no, sometimes there's interesting stuff in the lower-ranked ones too
* E5, subset in projection: somehow we didn't discuss it b/c there were questions around the metric used
* E5, freely arrange components: might be useful, but you can also look at them side by side

#### Easy stuff

* E4: slope graph & ability to change order
* E3: "you can very fast get an idea of how similar different methods are [with the component overview] ... and if you have [similar components] but they are in different positions"
* E2: table overview "really nice, really useful", changes in matrix view "nice that you can look at [components, inputs] in more detail", tooltips "nice"
* E1: reordering
* E5: "the most relevant stuff is easy to do", [N: like what] "the mixing matrices that you can get this graphical representation of the numbers, looking at this is always important, it's also important to compare different [component ensembles] this is a very straightforward way to do it [the slope graph] (...) it's very convenient to do here (...) i think the visualization [N: not sure what E5 refers to, is gesturing with pointer to the lag set vis] is very nice"

#### Unused stuff we talked about

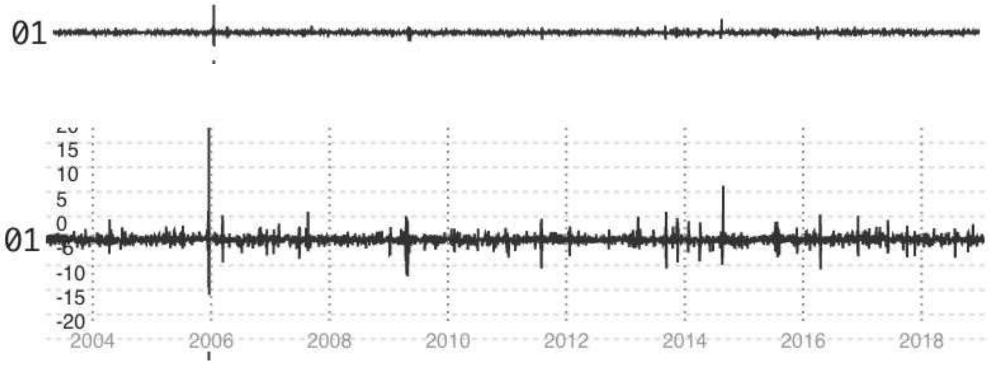
* E4: k1/k2 similarity projections too abstract, likes histograms better
* E5, download RData: didn't realize that variables are added to global environment

#### Anything missing

* E4: no, but already knew what to expect [and not many things were actually added]
* E3: "no new ideas, i think it's very good"
* E2: use own data
* E1: colorblind color palette
* E1: due to USD/HKD issue, one could add correlation slope between inputs and inputs to see how they correlate to themselves (would need to take care to not show the obvious 1s). E1 mentioned he'd go into R for this usually.
* E1: free ordering of components, but not necessary, as one can get any order underneath matrices
* E1: go back and forth between screens [works to some extent]
* E5: "i don't think so"

tsbss-vis cheat sheet

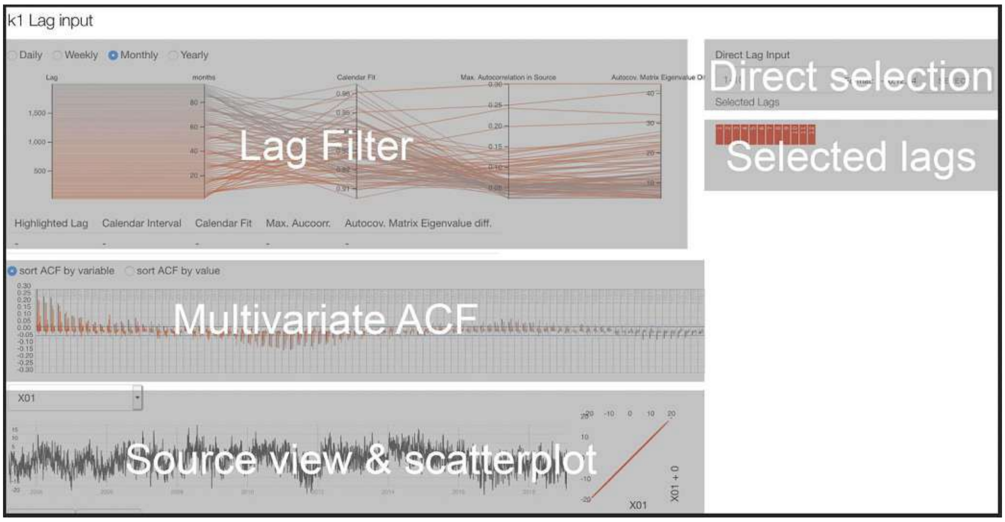
Time series



Purpose	Interaction	Encoding

● Show time-dependent variable	● Brush (left click + hold + move) zooms into brushed time range ● Double left click resets time zoom ● shift + left click increases Y resolution up to 3 times ● Right click resets Y resolution ● Hover shows time scale if Y resolution is small ● Label may be clicked to trigger an action depending on the context in which the time series appears	● If line is not black, it's a component belonging to that method ● Label on left side is either source series name or component's sort position

Lag selection

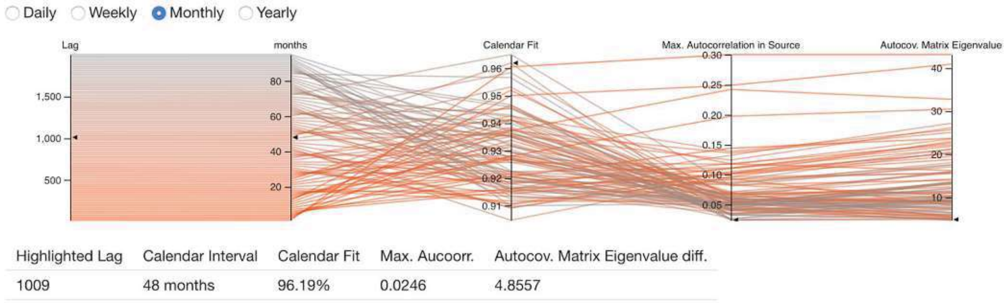


k1 Lag input
○ Daily ○ Weekly ● Monthly ○ Yearly
Lag
months
Calendar Fit
Max. Autocorrelation in Source
Autocov. Matrix Eigenvalue Diff.
Lag Filter
Highlighted Lag
Calendar Interval
Calendar Fit
Max. Aucoorr.
Autocov. Matrix Eigenvalue diff.
Direct Lag Input
Direct selection
Selected Lags
Selected lags
● sort ACF by variable ○ sort ACF by value
Multivariate ACF
X01
Source view & scatterplot
X01
X01 + 0

Purpose	Interaction	Encoding
● Select one or more lags	● All views are linked, so one influences	● Color saturation encodes the lag

	others. See following sections for details.	length, ie. more saturation = shorter lag.

Lag Filter

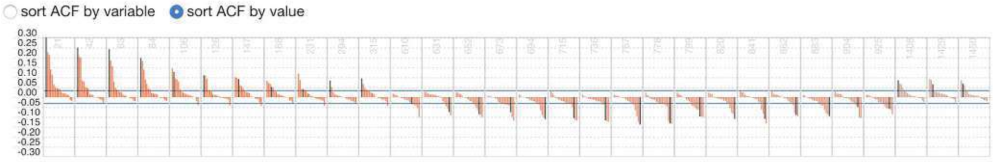


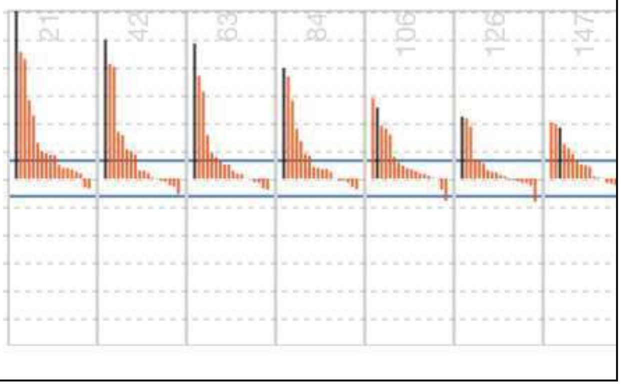
○ Daily ○ Weekly ● Monthly ○ Yearly

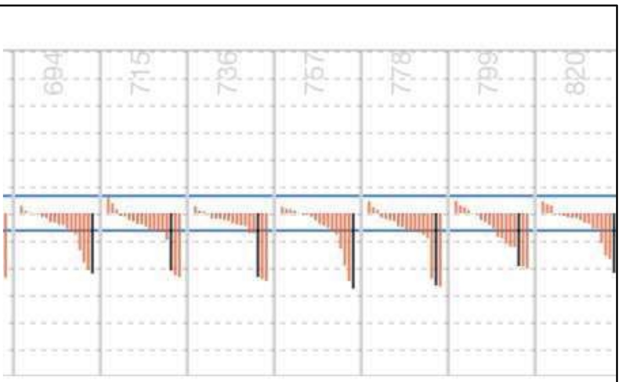
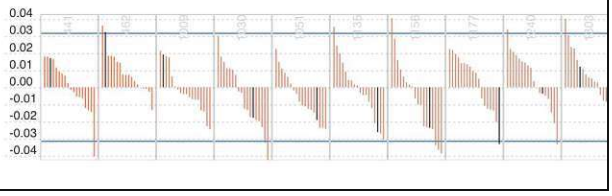
Highlighted Lag	Calendar Interval	Calendar Fit	Max. Aucoorr.	Autocov. Matrix Eigenvalue diff.
1009	48 months	96.19%	0.0246	4.8557

Purpose	**Interaction**	**Encoding (parallel coordinates)**
● Find interesting lags according to one or more derived attributes	● Brushing on dimensions filters displayed lags in Multivariate ACF ● Clicking outside of the brush on a dimension removes the brush ● Brushes can be moved ● Radio buttons on top filter lags in this view by equivalence to calendar intervals (days, weeks, months, years)	● Each vertical line is one attribute dimension ● Every other line = 1 lag ● Attributes of lag can be read from where line intersects dimensions ● Attributes of highlighted lag are marked with small triangle ● Attributes of selected lags are too, but triangle is further away from dimension (not visible)

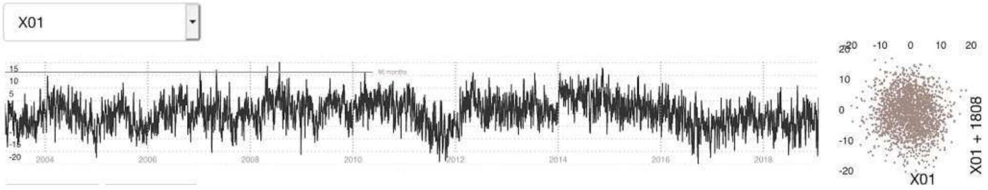
Multivariate ACF



Purpose	**Interaction**	**Encoding**
- Show autocorrelation of multiple sources	- Hover over lag to highlight it in Lag Filter and update scatterplot in Source & Scatterplot - Click lag to select/deselect it - Radio buttons on top change how lines are sorted: variable name (alphabetical) or autocorrelation value	- One “box” is one lag - Every line encodes the autocorrelation of a source at that lag - Saturation again encodes lag length - A black line encodes that source is selected and visible in Source & Scatterplot - Bold font and darker border encodes a selected or highlighted lag
Examples		
Predominantly positive autocorrelation in sources at that lag		

Predominantly negative autocorrelation in sources at that lag	
Little autocorrelation in sources	

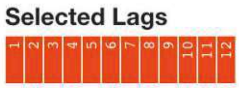
Source & Scatterplot



X01

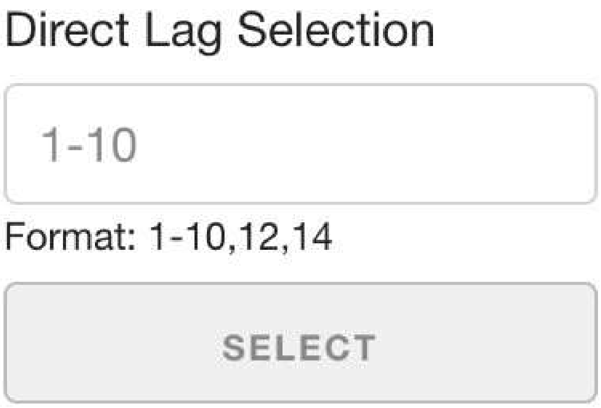
Purpose	Interaction	Encoding
• See correlation pattern in a source	• Select box on top changes which source is selected and thus visible in time series and scatterplot	• Scatterplot shows source vs source at highlighted lag • Horizontal line in time series shows time distance of highlighted lag

Selected Lags



Purpose	Interaction	Encoding
• Show which lags are currently selected	• Clicking a bar removes it from the selection • Hovering over a lag highlights it in the other views **if** it's visible acc. to current Lag Filter settings	• See Parameters

Direct Selection


Direct Lag Selection

1-10

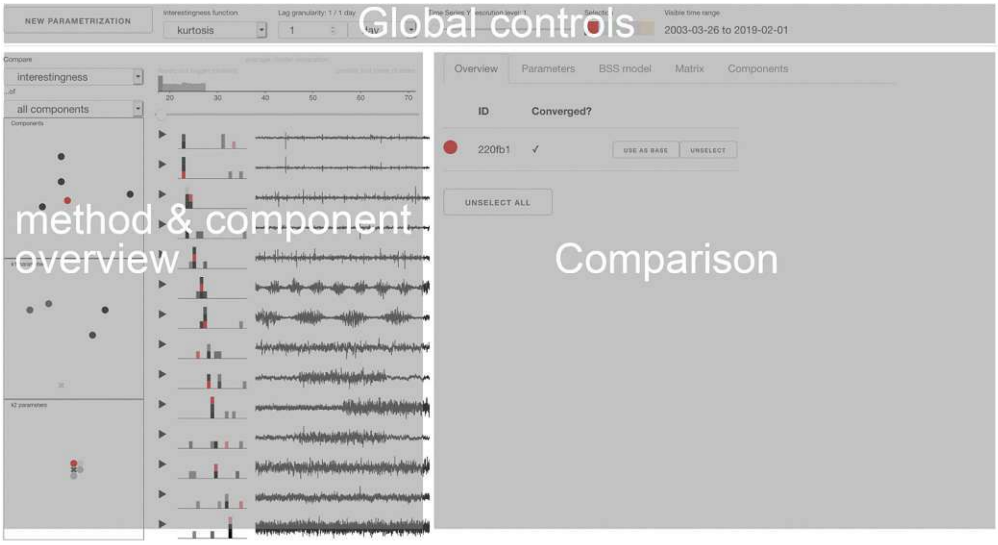
Format: 1-10,12,14

Purpose	Interaction	Encoding
• For when you exactly know which lags you want	• Enter lags and click button, lags will be toggled (ie. selected	none

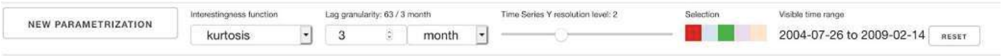
	when not selected, deselected when selected)	

Explore view

Color hue in these views always encodes that data belongs to a selected method.



Global controls



Interestingness function: kurtosis

Lag granularity: 63 / 3 month: 3 month

Time Series Y resolution level: 2

Selection: 🟥 🟩

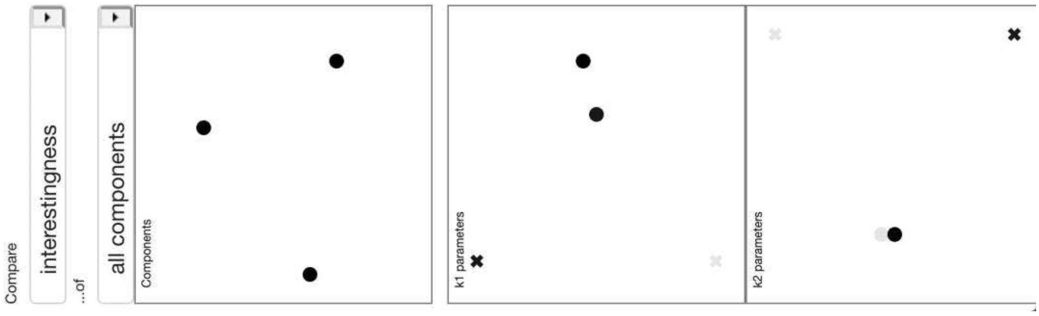
Visible time range: 2004-07-26 to 2009-02-14

Purpose	Interaction	Encoding
- Change view parameters	- “New parametrization” leaves current view and opens parameter input - “Interestingness function” changes how components are sorted - “Lag granularity”	none

	changes bin size for lag set similarity analysis (see parameters section) - “Time Series Y resolution” sets Y resolution for all time series at once - “Selection” shows how many methods can be added to comparison view - “Visible time range” shows start and end date in time series and allows to reset	

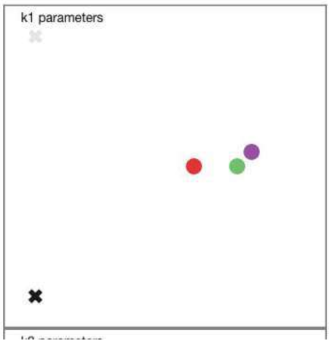
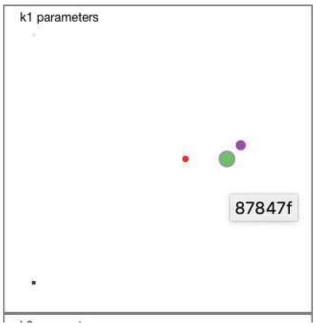
Method & component overview

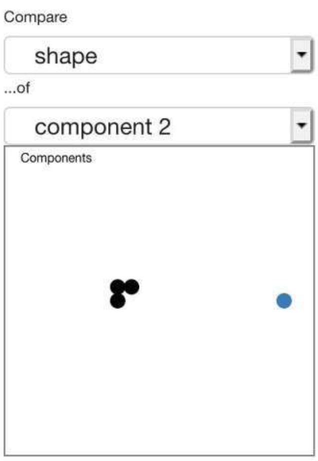
Method overview



(picture is rotated!)

Purpose	**Interaction**	**Encoding**
- Find similar/different methods according to components or parameters	- Hover encodes distance in size - Click selects method - For “Components” embedding, the above selectboxes define	- Position is a 2D embedding (metric multidimensional scaling) of higher-dimensional distances

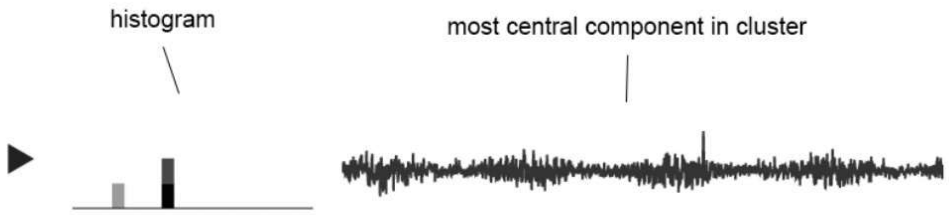
	what is compared (interestingness or shape) and for which component (all, one)	• Opacity (for k1/k2) encodes weight parameter • Shape encodes if a method converged (circle) or not (cross) • Size (on hover) encodes distance to hovered element
Examples		
Three methods have k1 parameter more similar to each other than to other two methods		
Hovering over green method reveals that red and purple are not *very* similar as they're clearly smaller than green's circle		

Component 2 (sorted by interestingness) of blue method doesn't look like the others	 Compare shape ...of component 2

Component overview

A constrained k-means clustering is applied to components of all methods, with constraints such that components of the same method cannot appear in the same cluster.

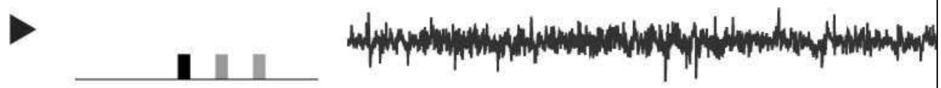
Cluster



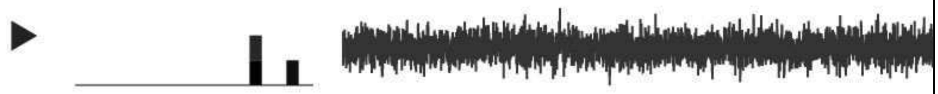
Purpose	Interaction	Encoding of histogram
- Find stable and unstable components - Get overview of found components	- Click triangle to show all contained components - Click number left of component to select this method	- 1 bar = 1 component - Opacity = similarity (eucl. distance) to central component - X position = sort position according to interestingness

Examples

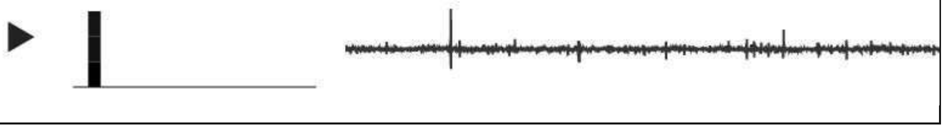
Cluster with less similar components



Cluster with similar components

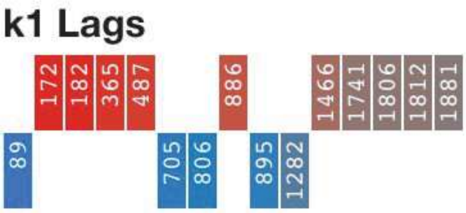


Cluster with similar components at same sort position

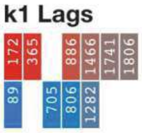


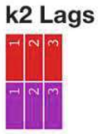
Comparison

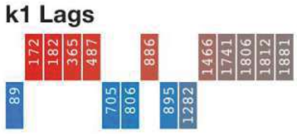
Parameters



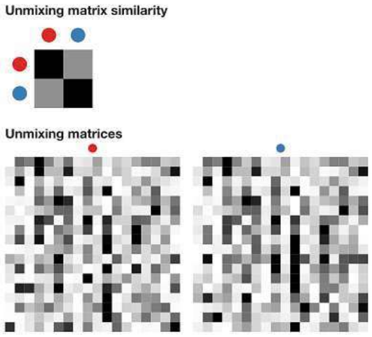
Purpose	Interaction	Encoding
● Assess similarity of lag sets ● Find out how they are same/different	● "Lag granularity" (not pictured) in global controls divides the range [1..max_lag] into equal sized bins. Each pictured lag set is filtered to include only the first lag in a bin (see examples).	● 1 bar = lag set has at least 1 lag in bin ● Y position and color hue encode method ● Saturation encodes lag length (short lag = high saturation) ● X position = bin, but empty bins are

<table>
<tr><td></td><td></td><td>omitted</td></tr>
<tr><td colspan="3"><b>Examples</b></td></tr>
<tr><td colspan="3">Similar lags at granularity 1 year (bin size 252) (some bars with same X position)


Same lag sets (all bars have same X position)


Different lags (no bars have same X position)

</td></tr>
</table>

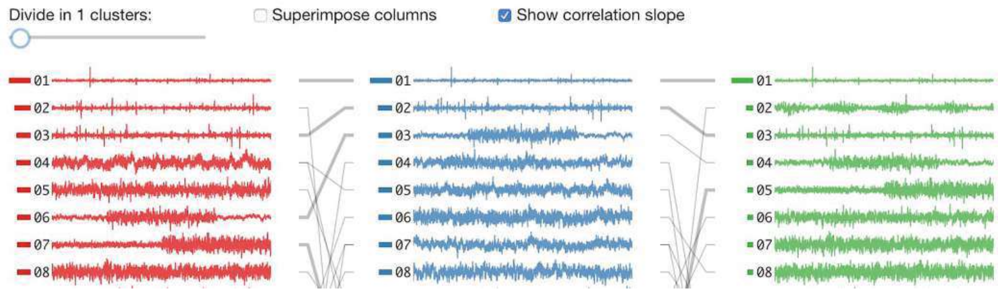
Matrix



Purpose	**Interaction**	**Encoding**
- See how similar unmixing matrices are - Find highest	- Hover over matrix cells to read values in tooltip	- Similarity encodes MD-Index between matrices, black is 0,

influencing factor		white is 1 • A matrix row encodes absolute value of factor in *row* from highest (black) to lowest (white)

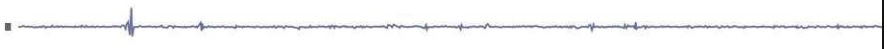
Components

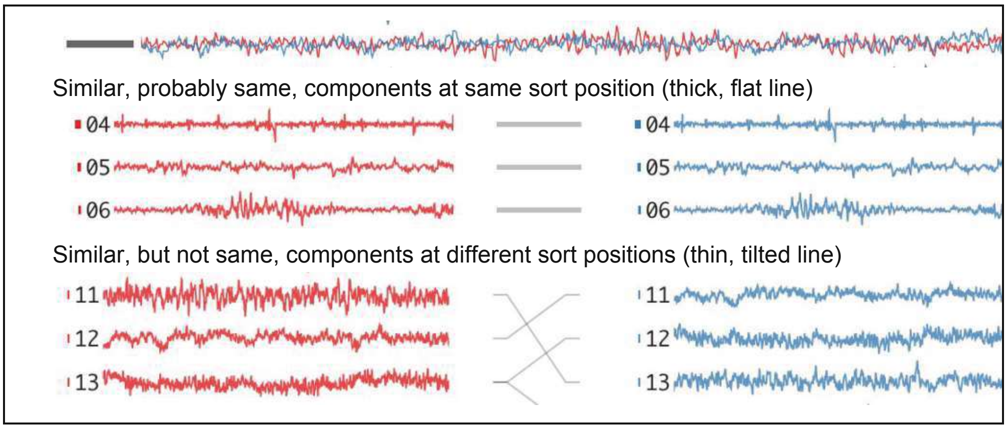


Divide in 1 clusters:

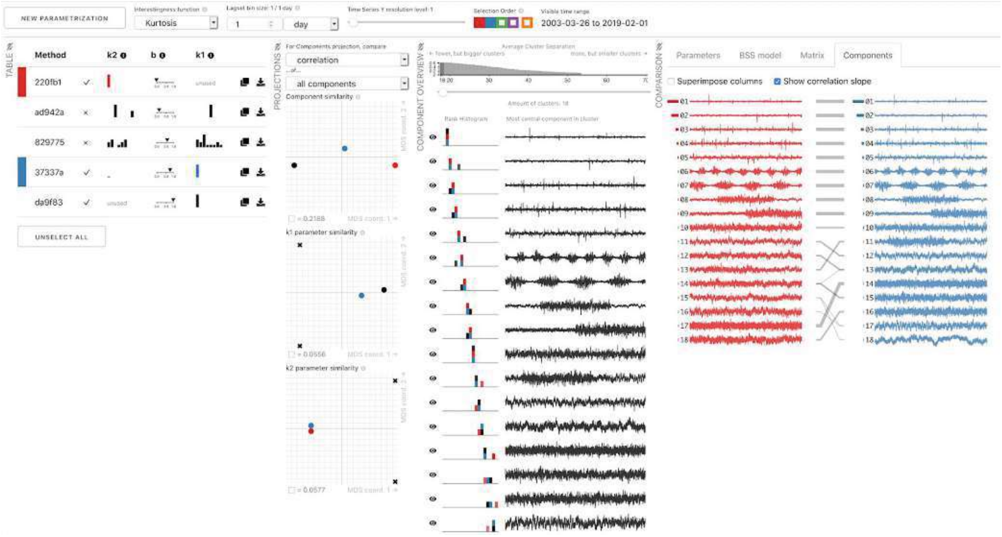
□ Superimpose columns

☑ Show correlation slope

Purpose	**Interaction**	**Encoding**
• Find common components • Find differences in components	• Change slider to divide each component's method into k clusters (PAM) • Toggle *superimpose* to overlay all components at same sort position • Toggle *correlation slope* to show lines that connect similar components (pictured) • Click number of a component to open and scroll to its cluster in the component overview	• Bar left of number encodes interestingness value relative to maximum value in method (when bar has color), or eucl. Distance between overlaid components (when it has no color) • Clusters are encoded in vertical distance between components • Line thickness encodes Pearson correlation (thin >= 0.5, thick >= 0.9)
Examples		
Similar components (low distance, one color overlays the other well)  Different components (high distance, colors do not overlay well)		


Similar, probably same, components at same sort position (thick, flat line)
04
05
06
04
05
06
Similar, but not same, components at different sort positions (thin, tilted line)
11
12
13
11
12
13

tsbss-vis updates



Interestingness function: Kurtosis

Lagset bin size: 1 / 1 day: 1 day

Time Series Y resolution level: 1

Visible time range: 2003-03-26 to 2019-02-01

For Components projection, compare: correlation of all components

Method		k2	b	k1
220fb1	✓			unused
ad942a	×			
829775	×			
37337a	✓			
da9f83	✓	unused		

[Overview - Table](#)

   [Export method](#)

[Overview - Projections](#)

[Overview - Components](#)

[Comparison - Parameters (Lag bin size = 10)](#)

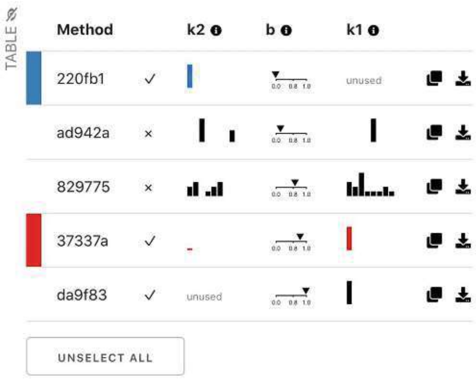
[Comparison - BSS Model](#)

[Comparison - Unmixing Matrix](#)

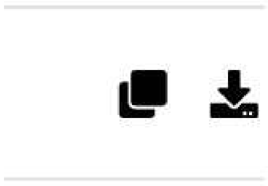
[Comparison - Components](#)

[Parameter Selection](#)

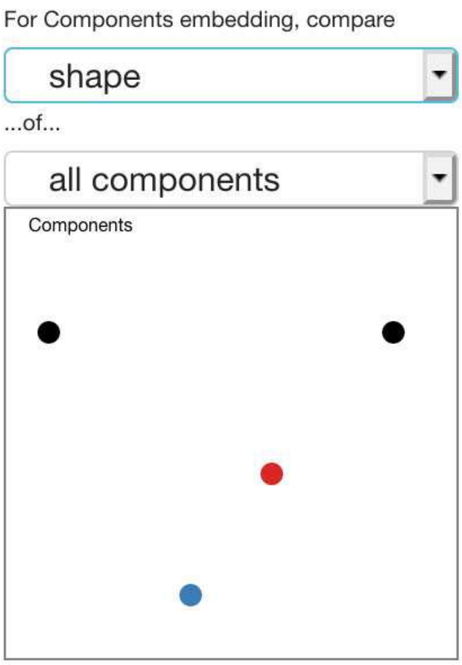
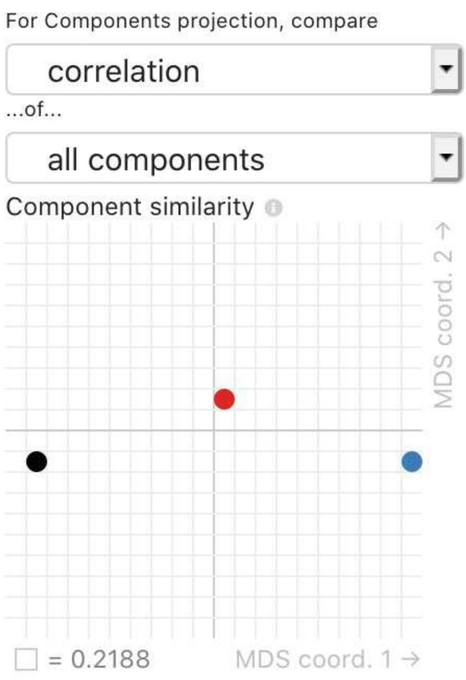
Overview - Table

Before	Now
None	 TABLE Method | k2 | b | k1 220fb1 ✓ | | | unused ad942a × | | | 829775 × | | | 37337a ✓ | | | da9f83 ✓ | unused | | UNSELECT ALL
Had to select all methods and inspect parameters	See all available methods and if they converged, their parameters, and additional actions

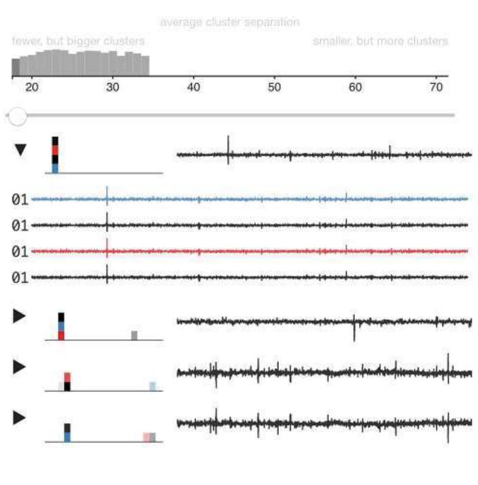
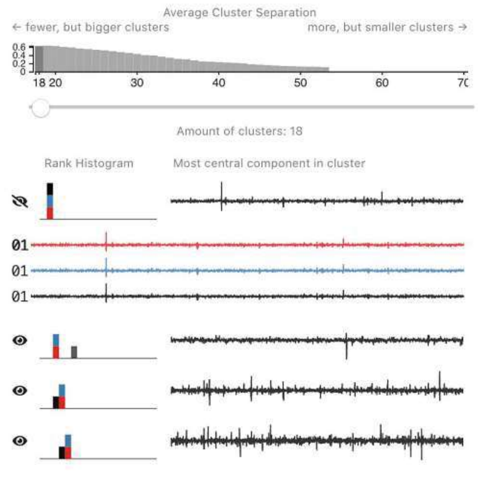
Export method

Before	Now
None	
Could not export data	Right icon downloads parameters, unmixing matrix and components as .RData file

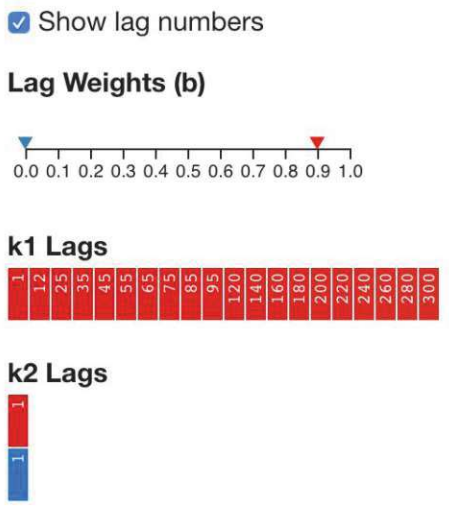
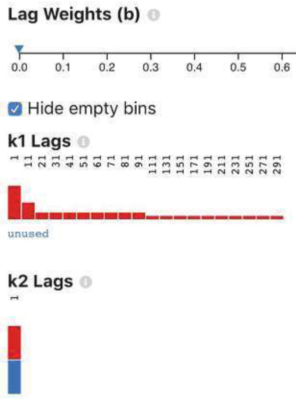
Overview - Projections

Before	Now
 For Components embedding, compare shape ...of... all components 	 For Components projection, compare correlation ...of... all components Component similarity
Shape distance measured in euclidean distance, no axis labels and ticks	Shape distance measured in correlation difference, axis labels, ticks, legend of how much difference is one tick is in the original MDS projection

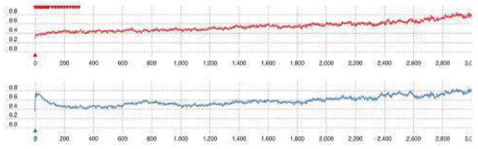
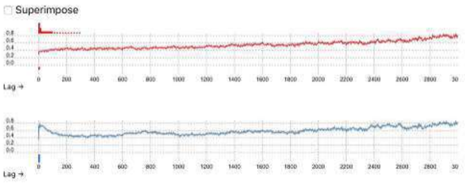
Overview - Components

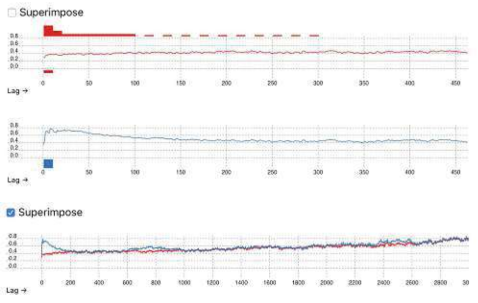
Before	Now
 average cluster separation fewer, but bigger clusters smaller, but more clusters 20 30 40 50 60 70 01 01 01 01	 Average Cluster Separation ← fewer, but bigger clusters more, but smaller clusters → 0.6 0.4 0.2 0 18 20 30 40 50 60 70 Amount of clusters: 18 Rank Histogram Most central component in cluster 01 01 01
Dissimilarity of components measured in euclidean distance, could not tell which of the components in a cluster is the medoid	Dissimilarity of components measured with correlation (not prone to sign change, note 3rd and 4th component in before image), cluster medoid has number in bold font

Comparison - Parameters (Lag bin size = 10)

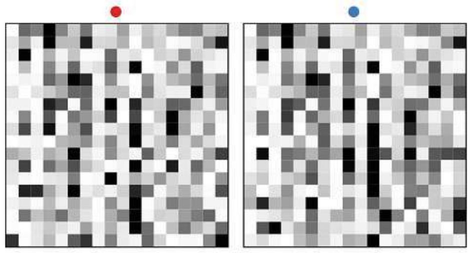
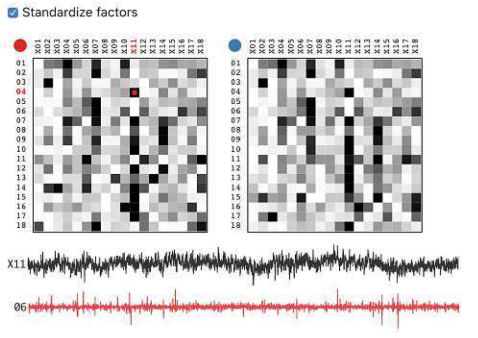
Before	Now
 ☑ Show lag numbers Lag Weights (b) k1 Lags k2 Lags 	 Lag Weights (b) 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 ☑ Hide empty bins k1 Lags 1 11 21 31 41 51 61 71 81 91 111 131 151 171 191 211 231 251 271 291 unused k2 Lags
Frequency in bin not encoded, needed to rely solely on color saturation to tell how big a lag is, unused parameter omitted (blue k1)	Frequency in bin encoded with height (ie. it's a histogram now), possibility to display empty bins to judge from position how big a lag is (however that doesn't scale well), unused parameter shows label

Comparison - BSS Model

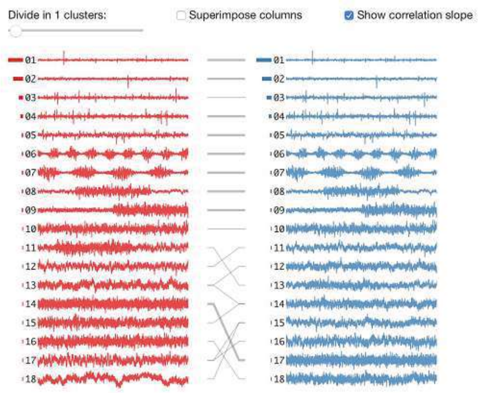
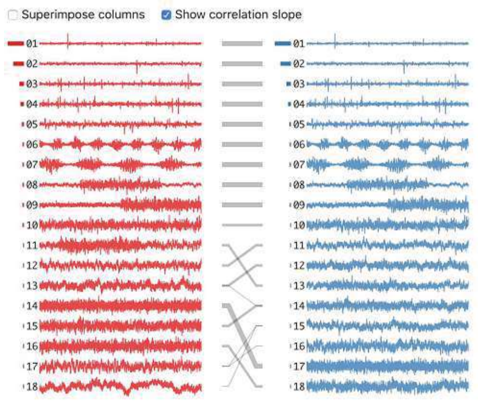
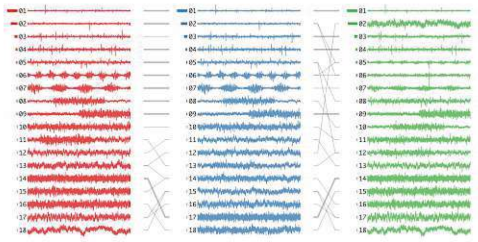
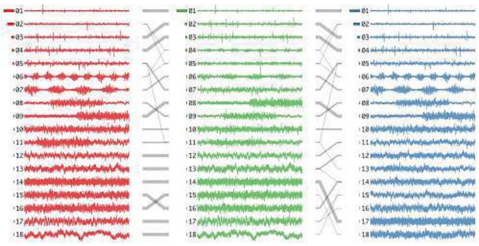
Before	Now
 0.8 0.6 0.4 0.2 0.0 0 200 400 600 800 1,000 1,200 1,400 1,600 1,800 2,000 2,200 2,400 2,600 2,800 3,0	 □ Superimpose 0.8 0.6 0.4 0.2 0.0 0 200 400 600 800 1000 1200 1400 1600 1800 2000 2200 2400 2600 2800 30 Lag →

	
Used lags encoded with triangle marks, no ability to superimpose or zoom	Used lags encoded with bars from lag set histogram, possible to superimpose and zoom

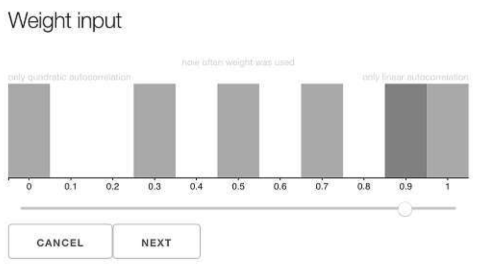
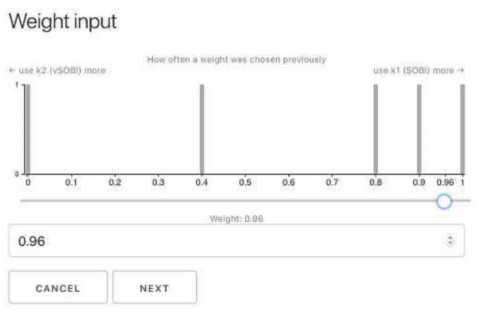
Comparison - Unmixing Matrix

Before	Now
	 ☑ Standardize factors X01 X02 X03 X04 X05 X06 X07 X08 X09 X10 X11 X12 X13 X14 X15 X16 X17 X18 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 X11 06
No labels (only with tooltip), no possibility to see standardized factors, no possibility to see input time series or components here	All of that

Comparison - Components

Before	Now
 Divide in 1 clusters: ☐ Superimpose columns ☑ Show correlation slope	 ☐ Superimpose columns ☑ Show correlation slope
K-medoids clustering based on euclidean distance, 2 classes for correlation slope, no tooltip, could not flip components if they seemed to be their respective inverse	No clustering anymore, 3 classes for correlation slope, tooltip shows correlation value, can flip components
	
To compare red and green, you had to unselect blue	Now order of colors/methods can be changed and is used in all views (ie. you can now also control the order in which components are superimposed)

Parameter Selection

Before	Now
 Weight input how often weight was used only quadratic autocorrelation only linear autocorrelation 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 CANCEL NEXT	 Weight input How often a weight was chosen previously ← use k2 (vSOBI) more use k1 (SOBI) more → 1 0 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 0.96 1 CANCEL NEXT Weight: 0.96 0.96
Only 11 available values in 0.1 resolution	0.01 resolution, direct input possible

TSBSS study

Interview structure

- Send email the day before
  - Send consent form, Z components, cheat sheet
  - Mention screencapture tool
- Before
  - Set up documents and note-taking
  - Set up stopwatch or other dedicated time measuring device
  - Restart servers to have clean slate again
- Intro
  - Small talk, nice to meet you, thanks for taking time
  - Are they comfortable & ready to start, remind that they can take a break any time
  - Introduce ourselves and explain what will happen
- Ask if they read and understood the consent form, and if they have questions
- Hit record, make sure **gallery view is on at all times** (meeting host settings are the recorded settings)
- Pre-study questions (10-15min)
  - Pre-study questions
- They share their screen and **close unrelated software**
  - Make sure they are allowed (Zoom security settings)
- Optional: The start screencapturing
  - Ask if they are ok doing it, if they tried it before, if they did how resource intensive was it (Zoom + prototype also need CPU), if in doubt leave it
- Introductory tour and tasks (target 60min)
  - http://guidance.cvast.tuwien.ac.at/synth_dist/#/data-load
  - Define vocabulary!
  - Intro script
- Optional: Additional time to explore, click around

- Break?
- Analysis on real dataset (30 min)
    - http://guidance.cvast.tuwien.ac.at/real_dist/#/
    - Open Tasks
- Post study questions (15 mins)
    - Post Study Questions

Pre-study questions (more structured than semi-structured)

- How would they describe their experience in statistics in general and blind source separation in particular?
    - Looking for: how long are they in the field already, do they teach it, did they publish in the field...
- How would they describe their experience with visualization?
    - Looking for: What vis do they use, what is visualized, do they generate them themselves, what software do they use to generate…
- How would you describe the role of visualization or visual analytics in blind source separation?
    - Looking for: How often do they use it for analysis, what kinds of visualizations do they use, had it helped them in the past in some way, where is high potential for vis…
- How do you currently use BSS algorithms?
    - Looking for: What data, what goals, what software, which parameters...
- How would they, with their current tools, given a dataset, go about selecting parameters or exploring/comparing components?
    - Maybe frame with an example: Real-world 25-dimensional time series with 3000 steps, if we asked them to compare a reasonable amount of parametrizations and provide insights how they relate to each other in terms of components obtained, properties of components, if there's a noticeable dependency between parameters and component properties… How'd they do it?
    - Looking for: Numbers they compute, plots they look at, interestingness functions / projection indices they use, how many runs do they try, anything they base parameter selection on?
- When comparing components, how do you currently deal with the fact that components are not ordered only defined up to sign?
- To sum up: What are shortcomings and limits of their current tools in this regard?
    - Looking for: Lacking interactivity, clutter, information not analyzed, scalability… that impede further analysis because it's tedious
- How prevalent is this type of analysis in your daily work? Independent of how much, would a domain expert in some field looking to use BSS benefit from it?
    - Looking for: How often, how detailed

Post Study Questions (semi-structured)

- What do you think of the prototype?
    - Looking for: Nothing, free form
- Did you encounter tasks that were easy to perform? Which?
- Did you encounter tasks that were not as easy to perform? Which, why and how could that be improved?
- Did you encounter tasks that were not possible at all? Why are they useful, what questions would they answer?
- How else could the prototype be improved?
    - Examples: Other measures to compute for scatters, other DOI functions, additional vis, other scatters, showing events...
- How does the prototype compare to their current tools?
- Do you know any BSS practitioners, i.e. domain experts and not part of your research clique, that we could get in contact with for future studies?
    - If not, where would we look?

Intro & closed tasks

Vocabulary

- method = result, or ask how they call it
- shape = how a time series looks, what features does it have where etc
- Interestingness = function (time series) -> number
- Sort position = the position of a component according to the decreasing sort order imposed by interestingness function [rank is already something else for statisticians, so avoid it]
- Parameter = k1, k2 or weight (b)

Intro

Order: Data load (left, right), explore (embeddings, clusters, comparison), new parametrization (weight, lag select)

- Data load screen
    - Left side: Input time series
        - Purpose: Sanity check of data
        - Interactions
            - Zoom = brush
            - Double click = reset zoom

      - Option/Alt + left click = increase Y resolution
        - No way to increase X resolution, need to work around with Zoom
      - Right click = reset Y resolution
      - Works everywhere but Lag Input
    - **Tasks**
      - Zoom into time range (brush)
      - Expand Y axis (shift + click)
      - Reset both (right click, double click)
  - Right side: Change Lag Boundary
    - We run computations and show visualizations only up to this lag, by default ¾*N
    - If there’s domain knowledge that says temporal correlation is lower, one can decrease it here. Example: EEG data are long (minutes in ms resolution), but Tang et al. say correlation only 300ms after.
    - It cannot be increased, we assume higher lags would contain too few points.
- Explore screen
  - Here, **color hue** always encodes the selected method. There’s a pool of 5 colors that are picked from in deterministic order (red, blue, green, purple, orange).
  - Components are always **sorted** according to selected interestingness
  - Left: Method overview maps
    - Methods and parameters are **embedded in 2D space**. One mark (dot or cross) represents one method in every* map. The closer two marks, the more similar are the methods.
    - Since the reduction from e.g. p-dimensional space to 2 comes with distance errors, and distances were altered additionally to avoid occlusion, the real distances are also encoded in mark size on hover. If a close mark changes size on hover -> it’s not as similar as distance suggests.
    - Result map
      - How distance is computed can be changed with selectbox 1: With interestingness of a component, or with distance to shape.
      - Which components are compared can be changed with selectbox 2: All components, or a single one (after sorting by interestingness).
    - Parameter maps
      - Distance encodes lag set similarity (Jaccard) -> very far apart means no lags in common, near means a few, close means many or all
      - Opacity encodes weight of the parameter -> if barely visible, the other lag set was used more
    - **Interaction**

- Hover encodes distance to hovered method in mark size in all maps, and reveals its components in histograms of component overview
- Click selects the method for closer inspection

- **Tasks**
  - Find a pair of very similar or different methods
    - How/in which way are they similar and different?
  - Find methods that have a comparable first component

- Middle: Component overview
  - A constrained k-means clustering is applied to components such that at most k clusters are formed, and components from same method cannot be assigned to same cluster.
  - K input
    - Bars represent a clustering quality metric (cluster separation: average minimum distance between any 2 elements of 2 clusters, higher is better) that should help inform selection of K
    - K at minimum forces clustering to find the closest component from all methods, can therefore be used to find stable components
    - K at another value allows to distribute components into more clusters, hence it can reveal components unique to a method, but after some value of K, components will be put into single clusters
    - One can also look at minimum K and look for clusters with many similar components but one
    - Interaction
      - Move range slider to change K
  - Clusters
    - Purpose: Get an overview of all obtained components
    - Clusters are sorted according to sort position of most central component
    - Toggler
      - Shows contained components in cluster underneath
    - Histogram (left)
      - X Position encodes sort order for each component in the cluster, the more left, the higher sorted
      - Components with same sort order are stacked. The higher, the more components are sorted at this position.
      - Opacity encodes (shape) distance to cluster medoid. The darker, the closer.
      - Good clusters have mostly dark bars in mostly the same position, distant gray bars indicate unrelated components, distant dark bars indicate similar components with different sort position

      - Bad clusters have mostly gray bars, and wide range of sort positions
    - Most central component (right)
      - Shows the cluster representative, the time series closest to all others in the cluster
    - Components (when toggled)
      - Number is sort position in own method
      - Interaction
        - Clicking number selects the method
        - Same interactions as time series
  - **Tasks**
    - Find a component that's common in all/most methods, ie. stable
      - [kurtosis, k=18] would you consider the 5th medoid stable? why?
    - Find a component that's found in only one/few methods, ie. unstable
      - [kurtosis, k=18] would you consider the 14th medoid unstable? Why?
- Right: Method comparison
  - Goal: Analyze different aspects of 1-5 methods closely
  - Overview
    - Table of selected results
    - ID is a hash of parameters, useful if you want to find a specific method again because colors are not stable
    - Interaction
      - Use as base opens parameter input with the method’s parameters preset
      - Unselect removes it from selection
      - Unselect all
  - Parameters
    - Weight
      - Triangle mark in hue of method is placed on common weight dimension, possibly stacked
    - Lag Set
      - One bar represents one lag bin, the hue is encodes the method, the saturation the length of the lag, X position encodes order but empty bins are omitted
      - Many well saturated next to low saturated bars -> There’s a “hole”. Lag sets with interleaved bars -> not similar, lag sets with bars in same x position -> similar.
      - If lag granularity is higher than lowest resolution, only its first lag in a window will be used, ie. lags 1-12 and resolution 3 days -> 1,4,7,10

- Interaction
  - Show lag numbers adds labels to bars
- **Tasks**
  - Explore
    - Find a pair of very similar or different methods with regards to their parameters, if there isn't any, how coarse do you need to make the lag analysis so they are?
      - Are the parameters of 37337a and da9f83 similar?
    - How does the lag granularity affect their similarity and why?
      - Does larger granularity equal more similarity? (no because depending on granularity bins might overlap or not)
  - Compare
    - (Pick any methods that are not very similar and)
    - Judge their parameter commonalities and differences
      - Is the weight the same and if not what's the difference?
      - Is the k1/k2 lag set the same and if no, where do they differ?
      - What’s the smallest granularity at which they have more lags in common?

- Model
  - Difference in autocov. matrix eigenvalue
    - The R command is there
    - Plots difference in scatter’s eigenvalues in absolute terms
    - This is not specific to a method, therefore no hue
  - Autocov. matrix diagonality
    - R command is there
    - Plots F-norm of off-diagonal elements in a diagonalized scatter
    - Triangle marks show where lags were placed (top=k1, bottom=k2)
  - **Tasks**
    - Does the location of lags influence the autocov.matrix diagonality?
- Matrix
  - Matrix similarity matrix
    - Shows similarity of unmixing matrices computed with MD index of MD(W_a, W_b^{-1})

    - Dark = similar, light = different
  - Matrix
    - Hue on top shows to which method it belongs
    - Cells are encoded row-wise where lowest factor is white and highest factor is black
    - Rows are sorted by component sort order acc. to interestingness
    - Factors should be normalized iff one is the inverse of another, haven't seen it happen though, don't know a good threshold
  - **Tasks**
    - How similar are the matrices overall?
    - Consider the whole matrix, what patterns do you see and how would you interpret them?
    - Find the source that explains most of the first component in a result
- Components
  - Here the actual obtained components of a method are shown
  - Component
    - Left: Length of bar, encoded per method, shows value of interestingness if colored, otherwise it's distance
    - Middle: Label shows position in method, allows interaction
    - Right: Time Series with usual interactions
    - Interaction
      - Clicking the label opens the cluster its contained in the overview, and scrolls to it. This allows to find related methods with similar components.
  - Interactions
    - Divide in clusters
      - Runs PAM with that K parameter, clusters are visible by horizontal separation
    - "Superimpose"
      - superimposes all same-positioned components. Mostly useful for 2 methods, but works for more. Is useful to investigate differences of same-ranked components. Bar now encodes sum of euclidean distances, ie. long bar -> higher difference compared to other rows.
    - "Show correlation slope"
      - Complements superimposition and allows to find similar components that are sorted elsewhere.
      - Line length encodes pearson correlation. Thin >= 0.5, Thick >= 0.9

      - **Tasks**
        - What are the most different components compared to the rest (same method)? (set clustering to 2)
        - Which components are the same shape and same sorted? (look for thick straight line)
        - Which components are different shape and same sorted? (look for no line)
        - Which components are same shape and differently sorted? (look for thick tilted line)
        - Which components are different shape and differently sorted? (look for no line)
        - Which components are similar, ie. not the same but not completely different? (look for thin line)
        - In which month/year can you spot differences? When are they the same? (superimpose or increase Y resolution and visually compare)
        - Can you find other results with such a component? (set k=p, open its cluster check if other same series is there)
    - Top: Global controls
      - New parametrization opens wizard for parameter input
      - Interestingness checkbox changes interestingness function
        - Interestingness: Takes component, gives number. Components are always sorted decreasingly according to their interestingness.
        - Built-in: SKewness, kurtosis, but can be anything that maps to TS to 1 number
      - Lag granularity changes window for lag analysis
      - Semantic zoom level changes Y resolution of all time series at once
      - Visible time range shows borders of current Zoom
      - Spinner indicates loading and shows up on longer requests
- Parameter Input
  - Weight Input
    - Bars encode how often it was used so far
  - 1-2x Lag Input depending on weight
    - One of the two is left out if weight is 0 or 1
    - Same component as on data load screen, but allows to select more than one lag
  - Unfortunately, because R is single-threaded, and many interactions incur requests to server, it’s best to just wait until new method appears, to avoid hiccups
  - When finished, the new result is added to the selection if possible
  - Lag Selection

- Purpose: Select a single or set of lags
- PCP
    - Purpose: Quickly find lags with interesting properties for closer inspection
    - Like a multidimensional scatterplot with parallel axes and lines instead of points: One line per lag, read values where line intersects the axis. Selected lags are marked by triangles at their dimension intersections, highlighted lag marked by a triangle that’s closer than the others.
    - Dimensions
        - Lag: Just the lag
        - [calendar unit]: Which calendar interval (in this unit) it fits best
        - Calendar Fit: The fit describes what percentage of data point pairs at this lag are this many calendar units apart.
        - Max Autocorrelation: The highest absolute autocorrelation in sources at this lag
        - Scatter Eigenvalue diff: Sum of differences in absolute eigenvalues of a scatter matrix
        - (Scatter Diagonality, only with base result: F-norm of off-diagonal elements of W * M * W^T)
    - Color
        - The length of a lag is always encoded in the saturation: More saturation = shorter lag
    - Interaction
        - Brush on axes to filter
        - Radio buttons to change calendar unit, this filters all lags to those who fit the calendar unit
- Multivariate ACF plot
    - Purpose: Analyze temporal correlation of sources at this lag
    - Same as univariate ACF, but with more variables
    - A box is one lag, the autocorrelation of every source at this lag is encoded in a line inside the box
    - Interaction
        - Hover highlights a lag in other views and makes detail table appear to read values
        - Radio buttons to change order of lines between alphabetical source name or autocorrelation value
        - Clicking a lag adds it to the selection
- Single source time series + scatterplot
    - Purpose: Analyze temporal correlation of a single source
    - One source can be investigated in more detail
    - This source will be shown in a line chart

  - The highlighted lag is encoded in a horizontal line in the line chart
  - A scatterplot shows correlation pattern at this lag for this source
  - Interaction
    - Selectbox to change source
    - Line chart supports other interactions, but not Y resolution change
- Direct lag input
  - If you know exactly what you want and like to skip the exploratory interaction, toggle any lags with this textbox
  - Interaction
    - Unfortunately no enter submit as of time of writing, click button
- **Tasks**
  - Select monthly lags with top 50% of autocorrelation, what pattern is visible? [all short]
  - What lag fits 29 months best? [610]
  - How well does it fit the interval of 29 months? [91%]
  - What source has biggest autocorrelation at this lag? [X11]
  - How would you describe the dependency of X11 at that lag? [normal distribution]
  - Try a new parametrization

Tasks

Open Tasks

Exrates, here it's really open analysis with a time limit of maybe 30min. We'll watch and ask them to talk aloud, sharing their insights with us. We help when they forget how to do something, and try to suggest analysis paths should they run out of ideas.

- They can pick any lag to investigate up to
- Possible suggestions to look for something
    - Explain existing methods
    - Special components (e.g. very stable/unstable)
    - Differences between otherwise similar components
    - Special methods (e.g. very easy/hard to replicate)
    - Parameter influence on method outcome (weight + mainly short/long lag sets)
        - Take a method, think what differences they'd expect when changing the parameters, then try
    - What's a good setting for lag granularity that makes sense?
- Possible questions near the end that may still incur analysis
    - If we were in a dimension reduction context, which method would you select, and which of its components would you say are signal/noise?

Design argumentation

What’s that for?

- Parameter space cannot be explored easily, which makes selecting parameters harder
    - Relation to calendar not done at all, but considered useful
        - Calculated best fit between lags and calendar granules
    - Scatters are autocovariance matrices, their eigenvalues indicate correlation, scatters with large eigenvalues are expected to reveal better components, and ideally a scatter has one large and many small eigenvalues
        - Calculated difference in eigenvalues
    - While not perfect (because of linear combination assumption), autocorrelation of sources can inform lag selection too
        - Calculated max autocorrelation per lag
    - Analysts wanted to look more closely into the data
        - Multivariate ACF, as that's what they look at with univariate data
        - Scatterplot of one selected source, requested specifically to see correlation pattern
- Analysts can't explore results of different parameterizations easily
    - They can't easily find different/similar results/parameters
        - MDS maps to plot similarity of runs according to result interestingness/shape, parameters
            - Analysts can find similar/different runs by comparing distance in map and verify projection by point size
    - Where same/similar components differ in sorting is interesting
        - Constrained clustering (only max. one component per run in a cluster) plus histogram of sort order with opacity-coding gives overview of sort position distribution and component similarity
- Comparison of parameters tricky
    - Lag set parameter space is large but exact comparison not useful
        - Lags can be binned into window
        - Visual comparison of lag sets because more detailed than Jaccard distance
- BSS model assumptions need to be validated, which is currently not done
    - Scatter diagonality: If BSS model holds well, all should be pretty diagonal, and more diagonal around where lags were picked
        - Compare line chart of diagonality as function of lag, with marks for used lags
    - Scatter eigenvalue diff not actually part of model, but can inform future parameterization and is interesting to see
- Comparison of obtained components is tricky

  - Analysts don't quickly see which components to compare to another or which are interesting
    - Components are always sorted by interestingness and grouped by result
    - Slope graph indicates difference regardless of sort order
    - Superimposition indicates difference when keeping sort order
    - PAM Clustering reveals the k most central components in the result
  - Components are defined only up to sign and order, ie. no inherent order, possibly flipped sign
    - Use DOI approach to sort components by an interestingness function, used by analysts already
    - Postprocessing of result looks for same-but-flipped component and normalizes those, so they don't appear flipped in the UI if they're similar enough
  - Detailed comparison of two/n>1 time series is cumbersome with their current tool
    - Interaction allows to zoom into interesting time range quickly and to switch between overview/detail on Y axis
- Currently no process/structure when investigating parametrizations
  - Keep track of which weight was already used, for lags not so important as there are so many, and too many of the same will result in ball structure in MDS
  - Possibility to iteratively refine a method by using it as base for new parametrization, new measures can be calculated in this case (scatter diagonality)
- Besides theoretical assumptions and insights from simulation studies (with other data) there's currently not a good way to see how change in parameters affect result for a given dataset
  - With iterative refinement this can now be better investigated

TSBSS study 2

Interview structure

- Send email the day before
    - Send consent form, Zoom invitation, cheat sheet with updates
- Before
    - Set up documents and note-taking
    - Set up stopwatch or other dedicated time measuring device
    - Restart servers to have clean slate again
- Intro
    - Small talk, thanks for taking time
    - Are they comfortable & ready to start, remind that they can take a break any time
    - Explain what will happen
- Ask if they read and understood the consent form, and if they have questions
- Hit record, make sure **gallery view is on at all times** (meeting host settings are the recorded settings)
- They share their screen and **close unrelated software**
    - Make sure they are allowed (Zoom security settings)
- Introductory tour and tasks (target 20min)
    - http://guidance.cvast.tuwien.ac.at/synth_dist/#/data-load
    - Intro (recall functionality and explain updates)
    - Tasks
- Optional: Additional time to explore, click around
- Analysis on real dataset (20 min)
    - http://guidance.cvast.tuwien.ac.at/real_dist/#/
    - They can pick any lag to investigate up to
    - Possible suggestions to look for something
        - Explain existing methods
        - Special components (e.g. very stable/unstable)
        - Differences between otherwise similar components
        - Special methods (e.g. very easy/hard to replicate)
        - Parameter influence on method outcome (weight + mainly short/long lag sets)
            - Take a method, think what differences they'd expect when changing the parameters, then try
        - What's a good setting for lag bin size that makes sense?
    - Possible questions near the end that may still incur analysis
        - If we were in a dimension reduction context, which method would you select, and which of its components would you say are signal/noise?

- Post study questions (10 mins)
    - Post Study Questions

Intro (recall functionality and explain updates)

- Data Load
    - Recall this shows input time series
    - Mention added global controls
- Explore
    - Mention every section can be collapsed
    - Note tooltips
    - Explain Table & change in highlighting/ordering
    - Projection changes: Grid, coordinate labels, correlation instead of ED, MD instead of Jaccard
    - Component overview: Cluster medoid has bold font, correlation instead of ED
    - Parameter comparison: More fine b, explain how lag set vis is now like a histogram
    - BSS Model comparison: Superimpose+zoom, histogram instead of triangles, additional scatters (4th cross cumulants, supposed to help with k2 lags)
    - Matrix comparison: Labels, scaling, see components/inputs
    - Component comparison: Removed clustering, change component sign, tooltips, more classes for correlation
- Parameter Input
    - More fine b
    - Lag selection: Removed calendar fit, changed default in MACF

Tasks

- Table
    - Download components of a converging method that placed more weight on k1 than on k2
    - How do lags differ between converged and not-converged methods?
- Projections
    - Measured in correlation difference, are 1st components more or less similar than 2nd components (any DOI)
        - Solution: check the number below the grid (tick size), or alternatively find most distant points and compare their coordinates in both projections
- Component overview
    - Find stable/unstable components
- Parameter comparison
    - Set bin size to 100, which k1 bin has most lags overall and for one method?

- BSS Model comparison
  - Zoom to range lag 1–500 and reset
  - Superimpose the graphs
  - Switch between scatters, if applicable
  - Which method has the more diagonal k1 scatters
- Matrix comparison
  - Scale to unit variance
  - Select/deselect inputs/components
  - Find most important/interesting input, if applicable
- Component comparison
  - Select 3 converging methods ABC, then compare slope graph of AC
    - Solution: reorder
  - Change sign of a component
  - Read correlation from tooltip

Post Study Questions (same)

- What do you think of the prototype?
  - Looking for: Nothing, free form
- Did you encounter tasks that were easy to perform? Which?
- Did you encounter tasks that were not as easy to perform? Which, why and how could that be improved?
- Did you encounter tasks that were not possible at all? Why are they useful, what questions would they answer?
- How else could the prototype be improved?

Post-study feedback

Thank you for participating in the user study regarding visual analytics for temporal Blind Source Separation. From past user studies we know that people sometimes get ideas only after the interview finished, when they had enough time to process their experience. Please feel free to reach out to us if you feel that we didn't discuss a particular idea about future improvements well enough, or at all. Find our interview guide for after the open analysis in the following.

- How would you describe the experience of using our prototype?
- How does the prototype compare to your current tools with regard to the tasks it was designed for, ie. selecting parameters and exploring components?
- What tasks were easy to carry out?
- What tasks were not so easy to do, and why was that?
- What tasks were not supported at all, and why would they be useful?
- What kind of data did you miss in the prototype, what did it not show but should?
- Do you have contacts to BSS "practitioners" in the sense that they're not statistics researchers, but actively do / want to use BSS? Would you be comfortable to connect us?
- If you don't have such contacts or are not comfortable sharing, do you have a suggestion how we could find such domain experts?

Online Material

All supplemental material to P3 may also be found on the open access web page of the article: https://doi.org/10.1016/j.visinf.2022.10.002 (accessed 15th May, 2024). Supplemental material not reproduced in this thesis:

- Synthetic Dataset: https://ars.els-cdn.com/content/image/1-s2.0-S2468502X22001103-mmc1.csv (accessed 15th May, 2024)
- Financial Dataset: https://ars.els-cdn.com/content/image/1-s2.0-S2468502X22001103-mmc2.csv (accessed 15th May, 2024)
- Medical Dataset: https://ars.els-cdn.com/content/image/1-s2.0-S2468502X22001103-mmc3.csv (accessed 15th May, 2024)
- Video Demo: https://ars.els-cdn.com/content/image/1-s2.0-S2468502X22001103-mmc5.mp4 (accessed 15th May, 2024)

Supplemental Material to Chapter 4

who	component	statement	score			S	mean	std dev
VE1	insight	1	6		insight	1	6.40	0.55
VE1	insight	2	7			2	7.00	0.00
VE1	insight	3	5			3	5.40	2.07
VE1	insight	4	6			4	6.40	0.55
VE1	insight	5	6			5	6.40	0.55
VE1	insight	6	7			6	6.80	0.45
VE1	insight	7	5			7	6.60	0.89
VE1	insight	8	6			8	5.80	0.84
VE1	time	9	6		time	9	6.80	0.45
VE1	time	10	6			10	6.20	0.45
VE1	time	11	7			11	6.80	0.45
VE1	time	12	5			12	6.00	1.00
VE1	time	13	7			13	6.80	0.45
VE1	essence	14	6		essence	14	6.40	0.55
VE1	essence	15	6			15	6.40	0.55
VE1	essence	16	5			16	5.80	1.10
VE1	essence	17	6			17	5.60	1.14
VE1	confidence	18	6		confidence	18	6.60	0.55
VE1	confidence	19	6			19	6.20	0.45
VE1	confidence	20	NA			20	**NA**	**NA**
VE1	confidence	21	4			21	4.33	2.61
VE2	insight	1	7					
VE2	insight	2	7					
VE2	insight	3	6		insight	*	6.35	0.98
VE2	insight	4	6		time	*	6.52	0.65
VE2	insight	5	7		essence	*	6.05	0.89
VE2	insight	6	7		confidence	*	6.00	2.98
VE2	insight	7	7					
VE2	insight	8	6					
VE2	time	9	7					
VE2	time	10	6					
VE2	time	11	7					
VE2	time	12	7					
VE2	time	13	7					
VE2	essence	14	6					
VE2	essence	15	7					
VE2	essence	16	7					
VE2	essence	17	6					
VE2	confidence	18	7					
VE2	confidence	19	7					
VE2	confidence	20	NA					
VE2	confidence	21	NA					
VE3	insight	1	6					
VE3	insight	2	7					
VE3	insight	3	7					
VE3	insight	4	6					
VE3	insight	5	7					
VE3	insight	6	7					
VE3	insight	7	7					
VE3	insight	8	7					
VE3	time	9	7					
VE3	time	10	7					
VE3	time	11	7					
VE3	time	12	7					
VE3	time	13	7					
VE3	essence	14	6					
VE3	essence	15	6					
VE3	essence	16	5					

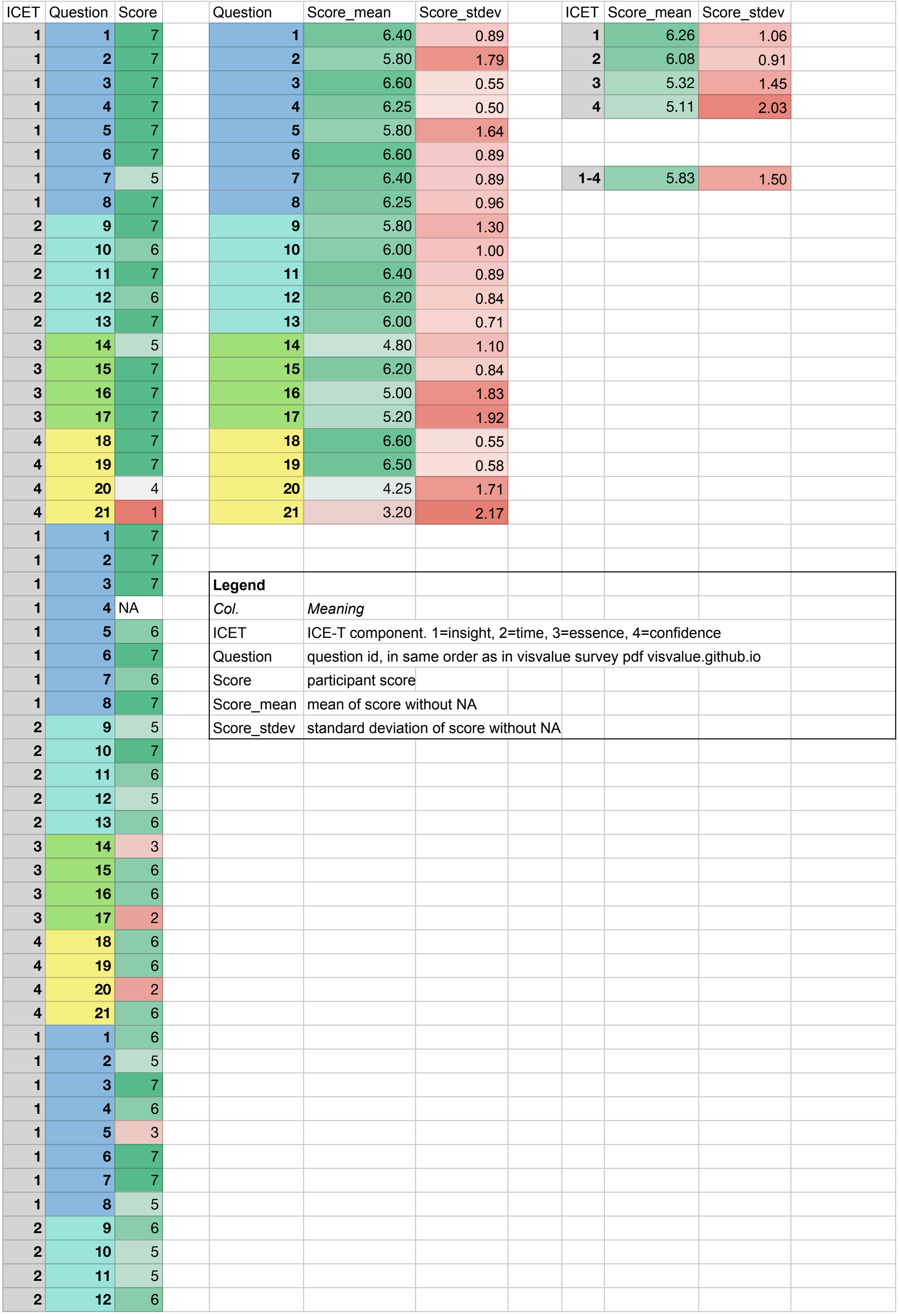
VE3	essence	17	5				
VE3	confidence	18	6				
VE3	confidence	19	6				
VE3	confidence	20	NA				
VE3	confidence	21	NA				
VE4	insight	1	6				
VE4	insight	2	7				
VE4	insight	3	2				
VE4	insight	4	7				
VE4	insight	5	6				
VE4	insight	6	6				
VE4	insight	7	7				
VE4	insight	8	5				
VE4	time	9	7				
VE4	time	10	6				
VE4	time	11	7				
VE4	time	12	6				
VE4	time	13	6				
VE4	essence	14	7				
VE4	essence	15	6				
VE4	essence	16	5				
VE4	essence	17	4				
VE4	confidence	18	7				
VE4	confidence	19	6				
VE4	confidence	20	NA				
VE4	confidence	21	3				
VE5	insight	1	7				
VE5	insight	2	7				
VE5	insight	3	7				
VE5	insight	4	7				
VE5	insight	5	6				
VE5	insight	6	7				
VE5	insight	7	7				
VE5	insight	8	5				
VE5	time	9	7				
VE5	time	10	6				
VE5	time	11	6				
VE5	time	12	5				
VE5	time	13	7				
VE5	essence	14	7				
VE5	essence	15	7				
VE5	essence	16	7				
VE5	essence	17	7				
VE5	confidence	18	7				
VE5	confidence	19	6				
VE5	confidence	20	7				
VE5	confidence	21	6				

Online Material

All supplemental material to P4 may also be found on the open access web page of the article: https://doi.org/10.1111/cgf.14530 (accessed 15th May, 2024). Supplemental material not reproduced in this thesis:

- Video Demo: https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2Fcgf.14530&file=cgf14530-sup-0001-S2.mp4 (accessed 15th May, 2024)

Supplemental Material to Chapter 5



ICET	Question	Score
1	1	7
1	2	7
1	3	7
1	4	7
1	5	7
1	6	7
1	7	5
1	8	7
2	9	7
2	10	6
2	11	7
2	12	6
2	13	7
3	14	5
3	15	7
3	16	7
3	17	7
4	18	7
4	19	7
4	20	4
4	21	1
1	1	7
1	2	7
1	3	7
1	4	NA
1	5	6
1	6	7
1	7	6
1	8	7
2	9	5
2	10	7
2	11	6
2	12	5
2	13	6
3	14	3
3	15	6
3	16	6
3	17	2
4	18	6
4	19	6
4	20	2
4	21	6
1	1	6
1	2	5
1	3	7
1	4	6
1	5	3
1	6	7
1	7	7
1	8	5
2	9	6
2	10	5
2	11	5
2	12	6

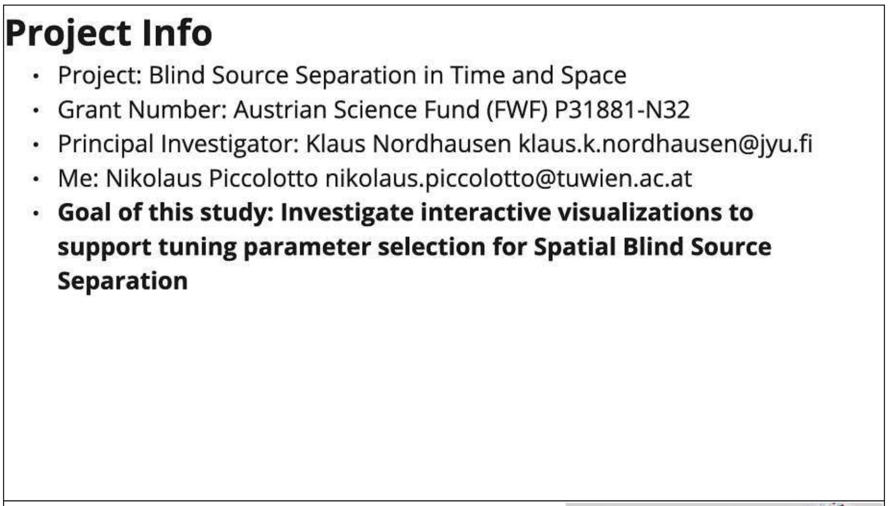
Question	Score_mean	Score_stdev
1	6.40	0.89
2	5.80	1.79
3	6.60	0.55
4	6.25	0.50
5	5.80	1.64
6	6.60	0.89
7	6.40	0.89
8	6.25	0.96
9	5.80	1.30
10	6.00	1.00
11	6.40	0.89
12	6.20	0.84
13	6.00	0.71
14	4.80	1.10
15	6.20	0.84
16	5.00	1.83
17	5.20	1.92
18	6.60	0.55
19	6.50	0.58
20	4.25	1.71
21	3.20	2.17

ICET	Score_mean	Score_stdev
1	6.26	1.06
2	6.08	0.91
3	5.32	1.45
4	5.11	2.03
1-4	5.83	1.50

Legend	
Col.	*Meaning*
ICET	ICE-T component. 1=insight, 2=time, 3=essence, 4=confidence
Question	question id, in same order as in visvalue survey pdf visvalue.github.io
Score	participant score
Score_mean	mean of score without NA
Score_stdev	standard deviation of score without NA

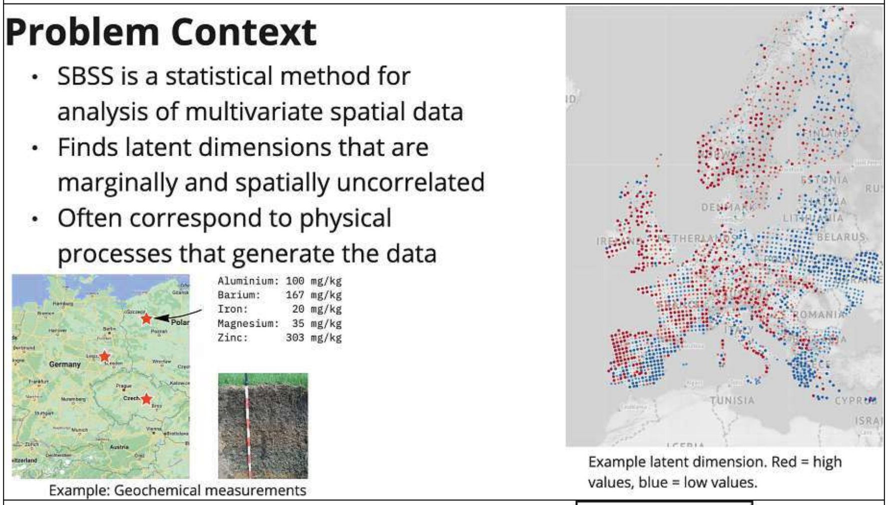
ICET	Question	Score		Question	Score_mean	Score_stdev		ICET	Score_mean	Score_stdev	
2	13	6									
3	14	6									
3	15	5									
3	16	4									
3	17	6									
4	18	7									
4	19	6									
4	20	6									
4	21	5									
1	1	5									
1	2	3									
1	3	6									
1	4	6									
1	5	6									
1	6	5									
1	7	7									
1	8	NA									
2	9	4									
2	10	5									
2	11	7									
2	12	7									
2	13	5									
3	14	5									
3	15	6									
3	16	3									
3	17	5									
4	18	6									
4	19	NA									
4	20	5									
4	21	2									
1	1	7									
1	2	7									
1	3	6									
1	4	6									
1	5	7									
1	6	7									
1	7	7									
1	8	6									
2	9	7									
2	10	7									
2	11	7									
2	12	7									
2	13	6									
3	14	5									
3	15	7									
3	16	NA									
3	17	6									
4	18	7									
4	19	7									
4	20	NA									
4	21	2									

Introduction Slides in Evaluation



Project Info

- Project: Blind Source Separation in Time and Space
- Grant Number: Austrian Science Fund (FWF) P31881-N32
- Principal Investigator: Klaus Nordhausen klaus.k.nordhausen@jyu.fi
- Me: Nikolaus Piccolotto nikolaus.piccolotto@tuwien.ac.at
- **Goal of this study: Investigate interactive visualizations to support tuning parameter selection for Spatial Blind Source Separation**



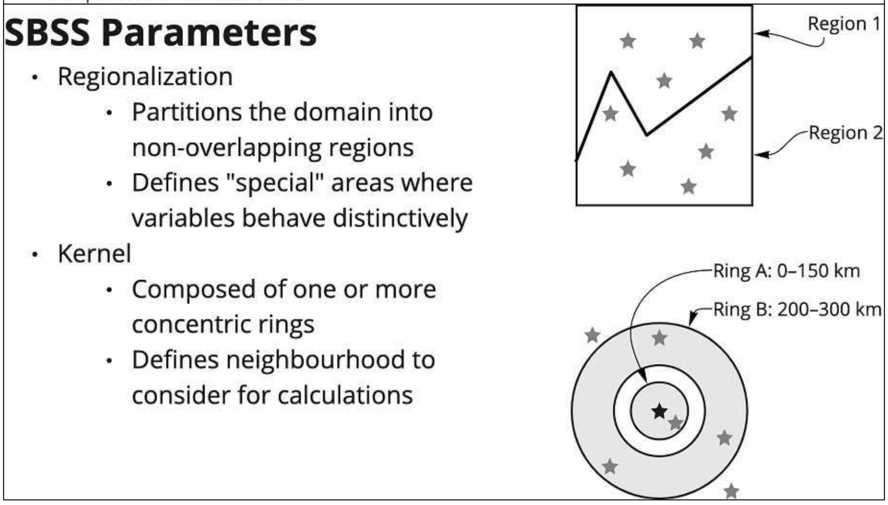
Problem Context

- SBSS is a statistical method for analysis of multivariate spatial data
- Finds latent dimensions that are marginally and spatially uncorrelated
- Often correspond to physical processes that generate the data

```
Aluminium: 100 mg/kg
Barium:    167 mg/kg
Iron:       20 mg/kg
Magnesium:  35 mg/kg
Zinc:      303 mg/kg
```

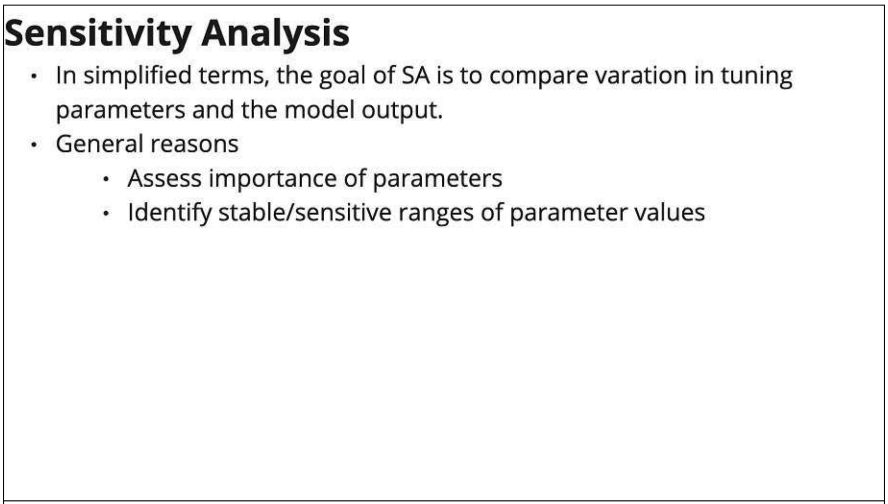
Example: Geochemical measurements

Example latent dimension. Red = high values, blue = low values.



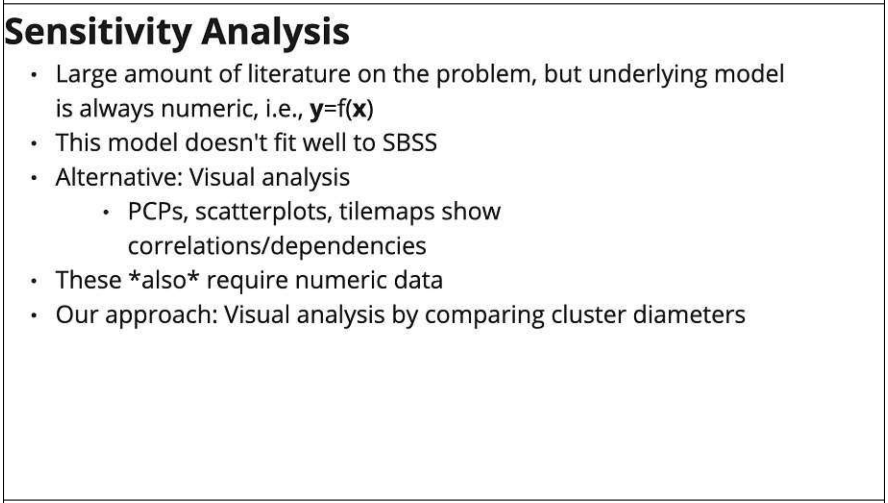
SBSS Parameters

- Regionalization
  - Partitions the domain into non-overlapping regions
  - Defines "special" areas where variables behave distinctively
- Kernel
  - Composed of one or more concentric rings
  - Defines neighbourhood to consider for calculations



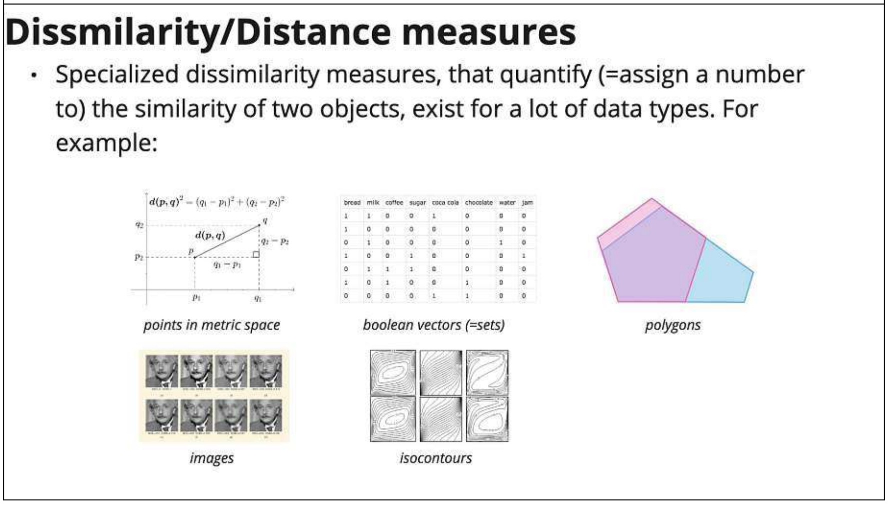
Sensitivity Analysis

- In simplified terms, the goal of SA is to compare varation in tuning parameters and the model output.
- General reasons
    - Assess importance of parameters
    - Identify stable/sensitive ranges of parameter values



Sensitivity Analysis

- Large amount of literature on the problem, but underlying model is always numeric, i.e., **y**=f(**x**)
- This model doesn't fit well to SBSS
- Alternative: Visual analysis
    - PCPs, scatterplots, tilemaps show correlations/dependencies
- These *also* require numeric data
- Our approach: Visual analysis by comparing cluster diameters



Dissimilarity/Distance measures

- Specialized dissimilarity measures, that quantify (=assign a number to) the similarity of two objects, exist for a lot of data types. For example:

$d(p,q)^2 = (q_1 - p_1)^2 + (q_2 - p_2)^2$
q_2
q
$d(p,q)$
$q_2 - p_2$
p_2
p
$q_1 - p_1$
p_1
q_1

points in metric space

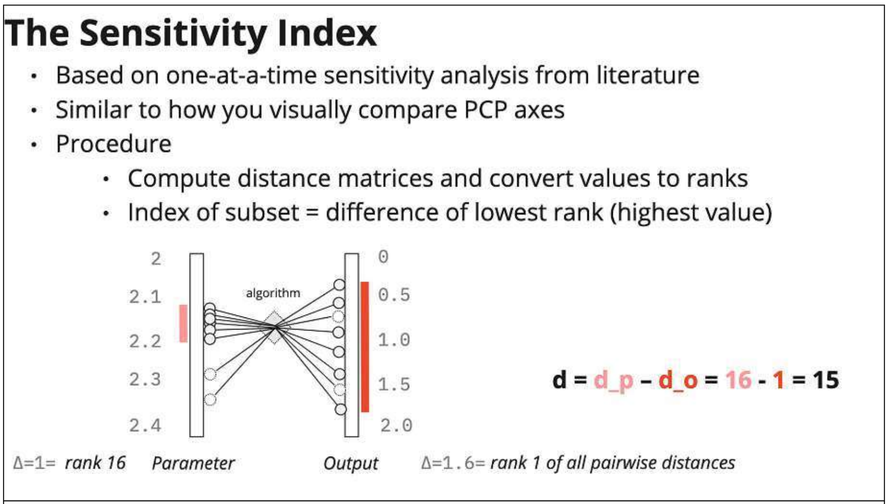
bread	milk	coffee	sugar	coca cola	chocolate	water	jam
1	1	0	0	1	0	0	0
1	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0
1	0	0	1	0	0	0	1
0	1	1	1	0	0	0	0
1	0	1	0	0	1	0	0
0	0	0	0	1	1	0	0

boolean vectors (=sets)

polygons

images

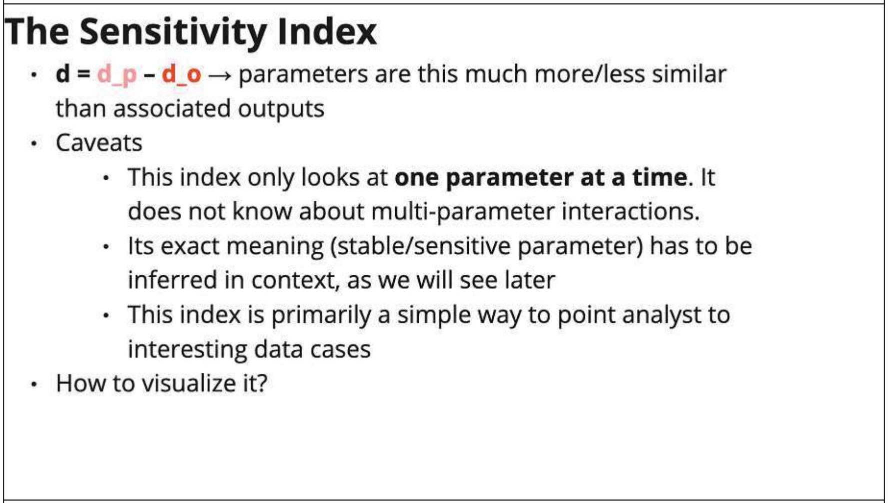
isocontours



The Sensitivity Index

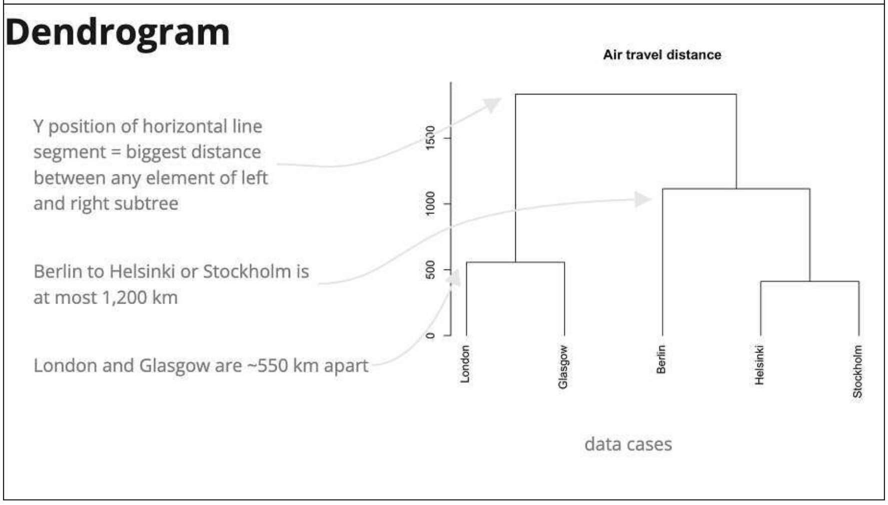
- Based on one-at-a-time sensitivity analysis from literature
- Similar to how you visually compare PCP axes
- Procedure
  - Compute distance matrices and convert values to ranks
  - Index of subset = difference of lowest rank (highest value)

d = d_p – d_o = 16 - 1 = 15



The Sensitivity Index

- **d = d_p – d_o** → parameters are this much more/less similar than associated outputs
- Caveats
  - This index only looks at **one parameter at a time**. It does not know about multi-parameter interactions.
  - Its exact meaning (stable/sensitive parameter) has to be inferred in context, as we will see later
  - This index is primarily a simple way to point analyst to interesting data cases
- How to visualize it?

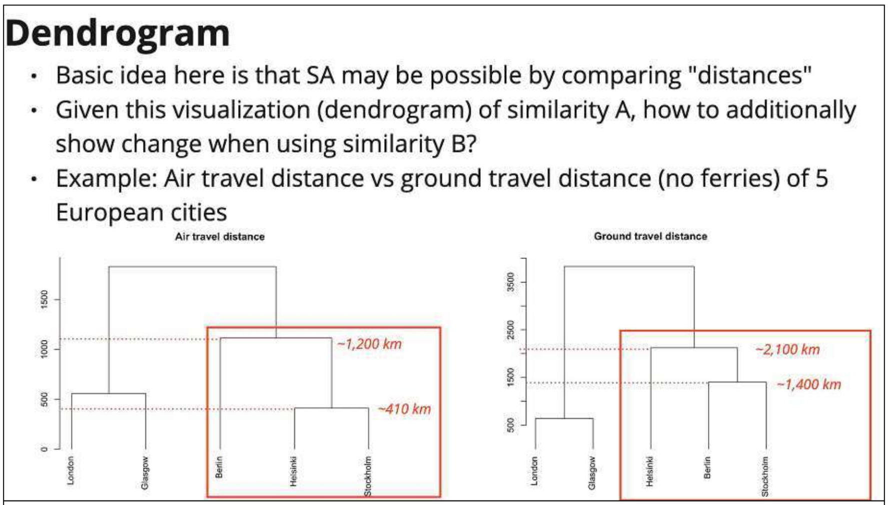


Dendrogram

Y position of horizontal line segment = biggest distance between any element of left and right subtree

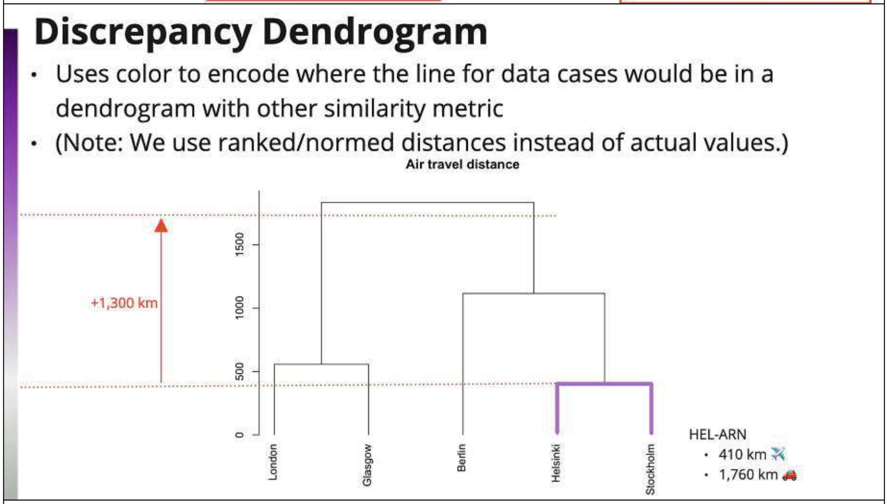
Berlin to Helsinki or Stockholm is at most 1,200 km

London and Glasgow are ~550 km apart



Dendrogram

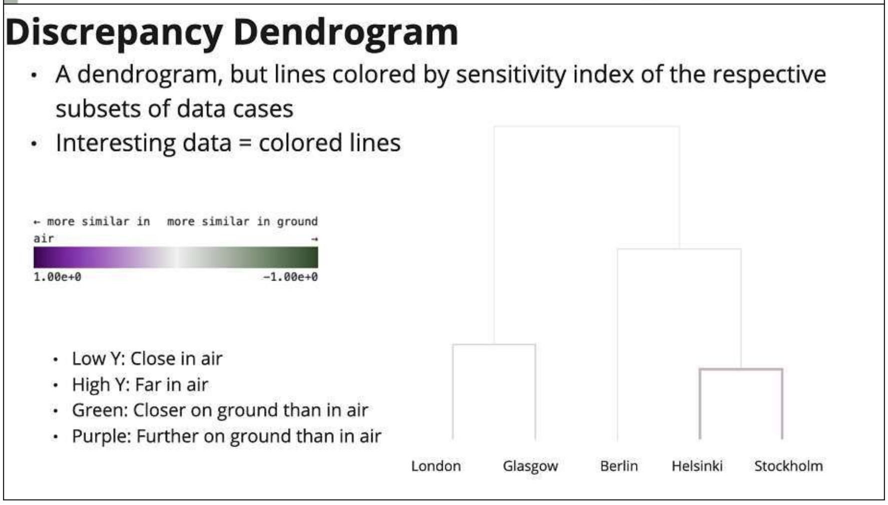
- Basic idea here is that SA may be possible by comparing "distances"
- Given this visualization (dendrogram) of similarity A, how to additionally show change when using similarity B?
- Example: Air travel distance vs ground travel distance (no ferries) of 5 European cities



Discrepancy Dendrogram

- Uses color to encode where the line for data cases would be in a dendrogram with other similarity metric
- (Note: We use ranked/normed distances instead of actual values.)

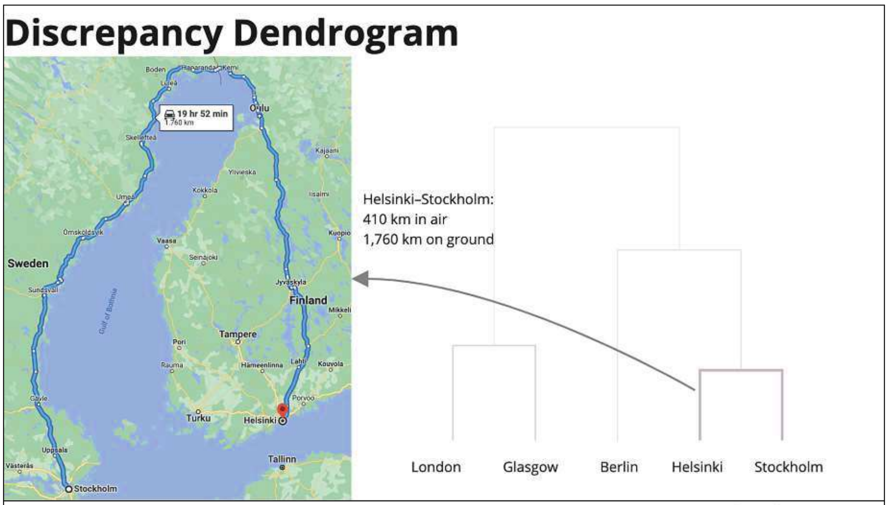
HEL-ARN
- 410 km ✈
- 1,760 km 🚗



Discrepancy Dendrogram

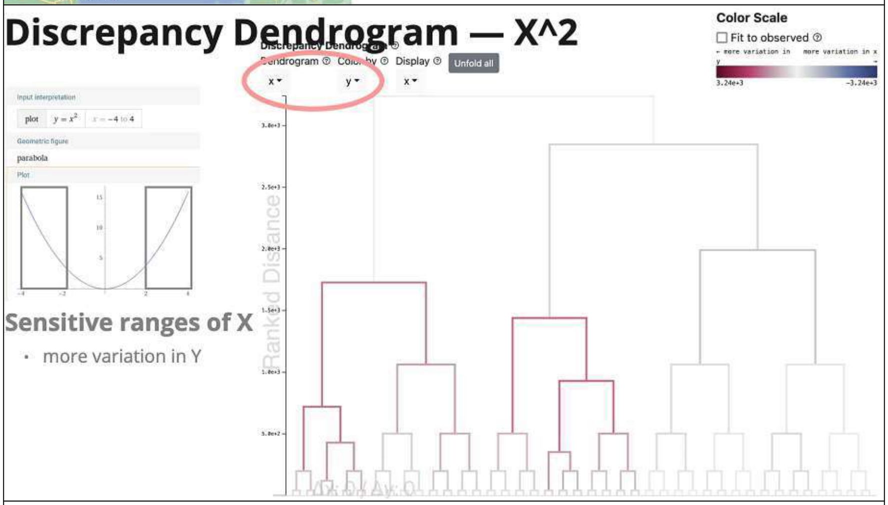
- A dendrogram, but lines colored by sensitivity index of the respective subsets of data cases
- Interesting data = colored lines

- Low Y: Close in air
- High Y: Far in air
- Green: Closer on ground than in air
- Purple: Further on ground than in air



Discrepancy Dendrogram

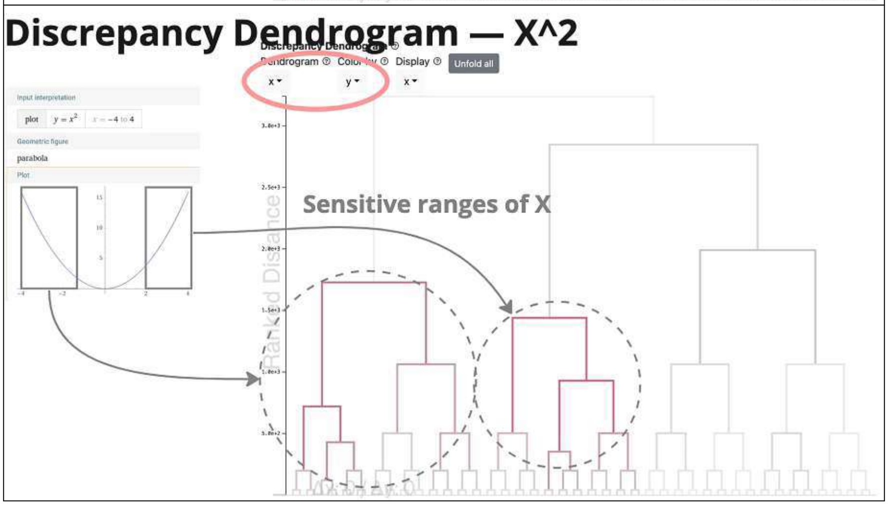
Helsinki–Stockholm:
410 km in air
1,760 km on ground



Discrepancy Dendrogram — X^2

Sensitive ranges of X

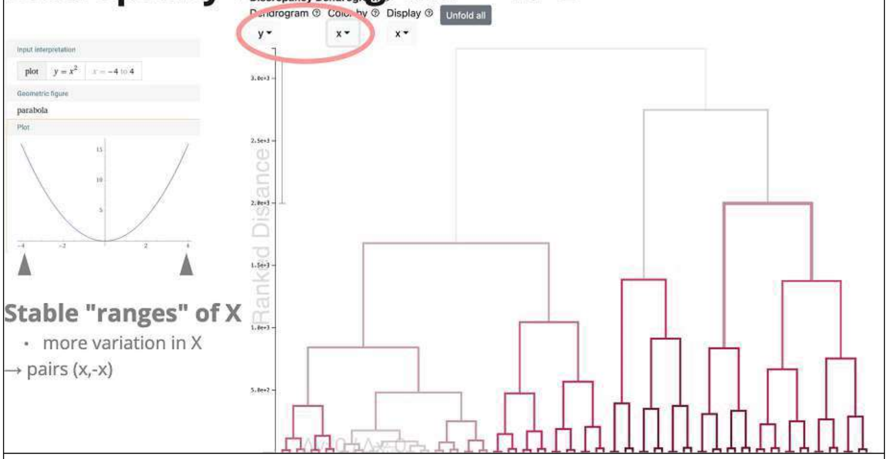
- more variation in Y



Discrepancy Dendrogram — X^2

Sensitive ranges of X

Discrepancy Dendrogram — X^2

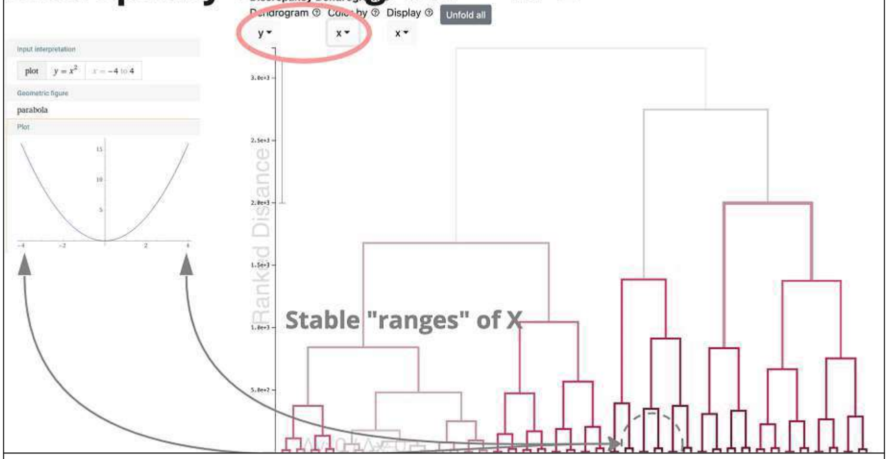


Stable "ranges" of X

- more variation in X

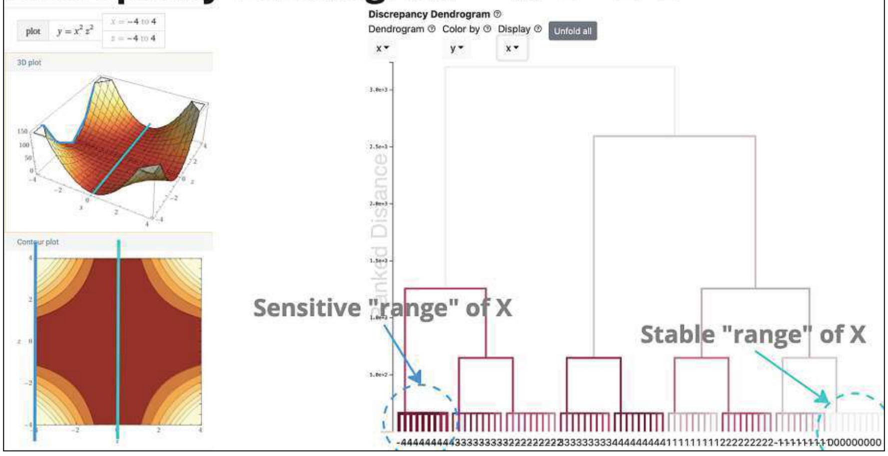
→ pairs (x,-x)

Discrepancy Dendrogram — X^2



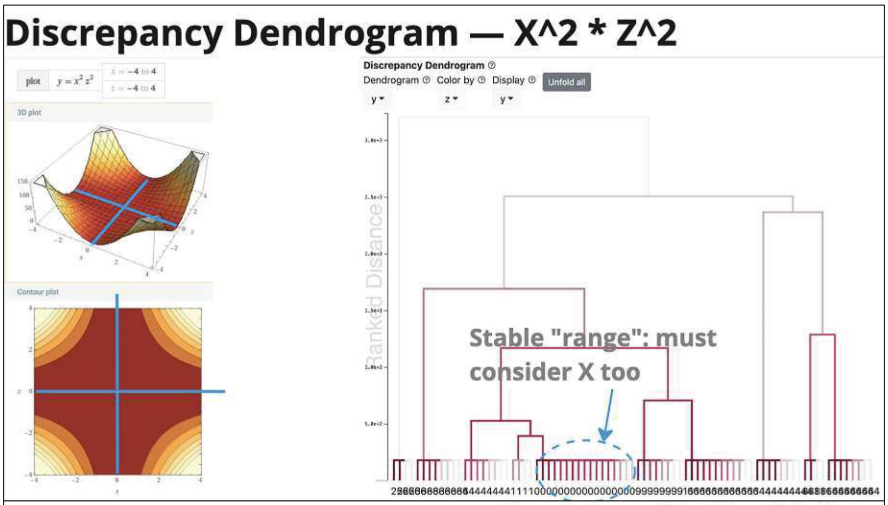
Stable "ranges" of X

Discrepancy Dendrogram — X^2 * Z^2



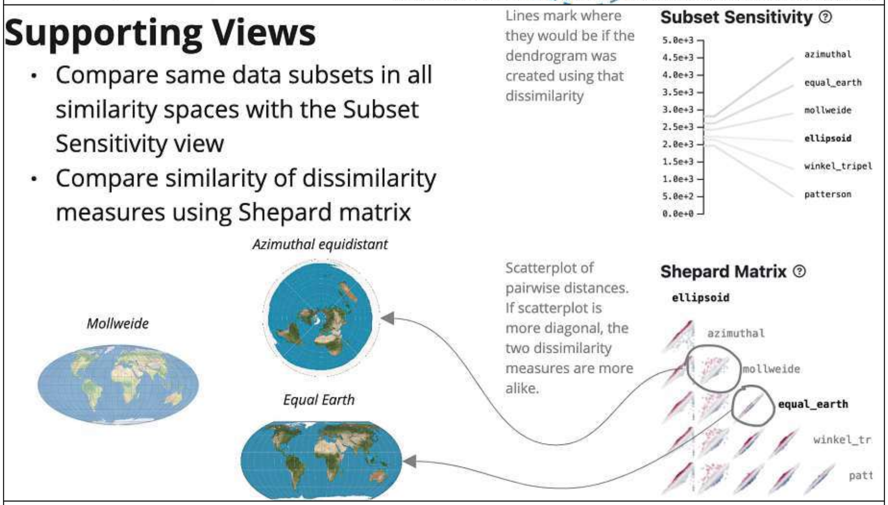
Sensitive "range" of X

Stable "range" of X



Discrepancy Dendrogram — X^2 * Z^2

Stable "range": must consider X too

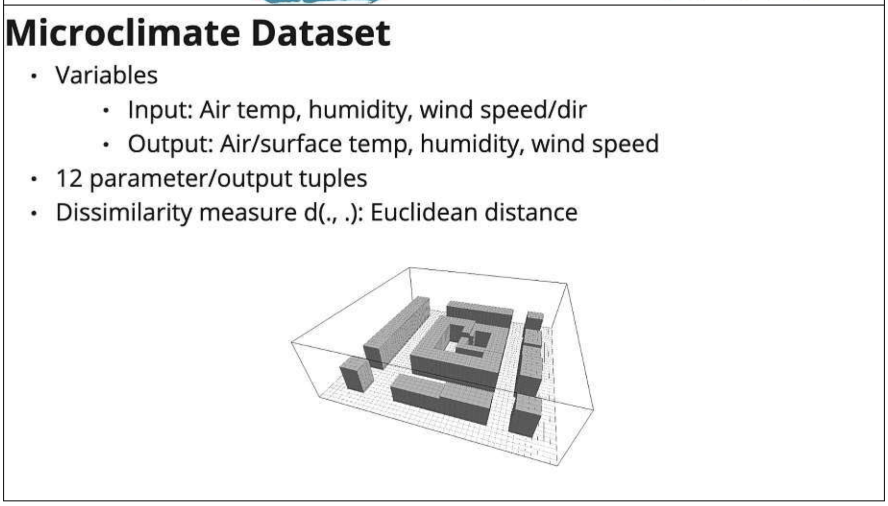


Supporting Views

- Compare same data subsets in all similarity spaces with the Subset Sensitivity view
- Compare similarity of dissimilarity measures using Shepard matrix

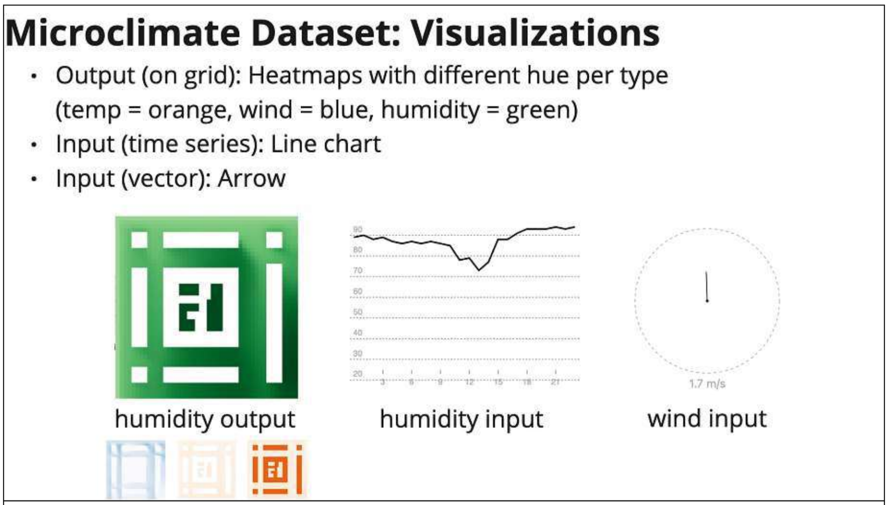
Lines mark where they would be if the dendrogram was created using that dissimilarity

Scatterplot of pairwise distances. If scatterplot is more diagonal, the two dissimilarity measures are more alike.



Microclimate Dataset

- Variables
  - Input: Air temp, humidity, wind speed/dir
  - Output: Air/surface temp, humidity, wind speed
- 12 parameter/output tuples
- Dissimilarity measure d(., .): Euclidean distance



Microclimate Dataset: Visualizations

- Output (on grid): Heatmaps with different hue per type (temp = orange, wind = blue, humidity = green)
- Input (time series): Line chart
- Input (vector): Arrow

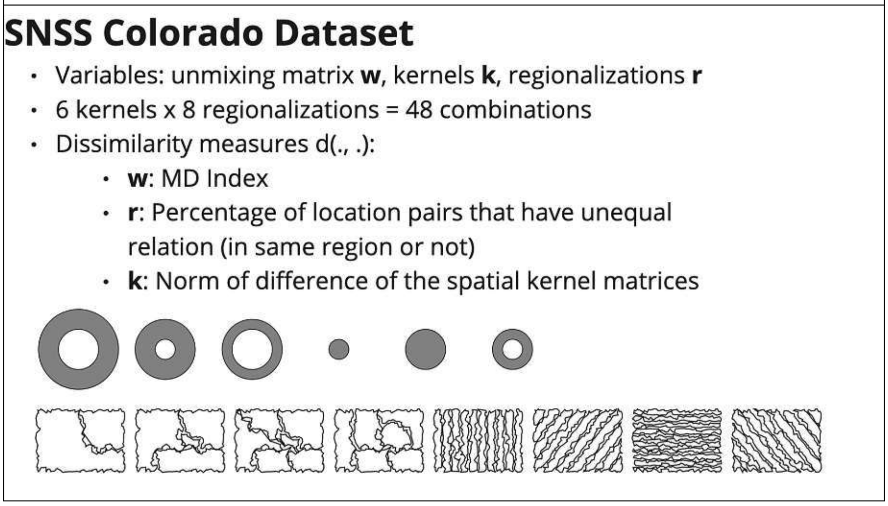
humidity output

humidity input

wind input

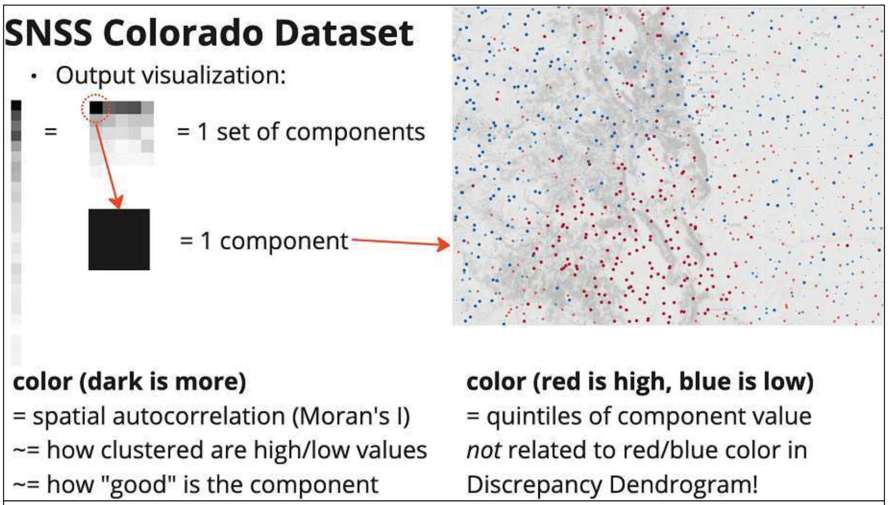
Demo time

link



SNSS Colorado Dataset

- Variables: unmixing matrix **w**, kernels **k**, regionalizations **r**
- 6 kernels x 8 regionalizations = 48 combinations
- Dissimilarity measures d(., .):
  - **w**: MD Index
  - **r**: Percentage of location pairs that have unequal relation (in same region or not)
  - **k**: Norm of difference of the spatial kernel matrices



SNSS Colorado Dataset

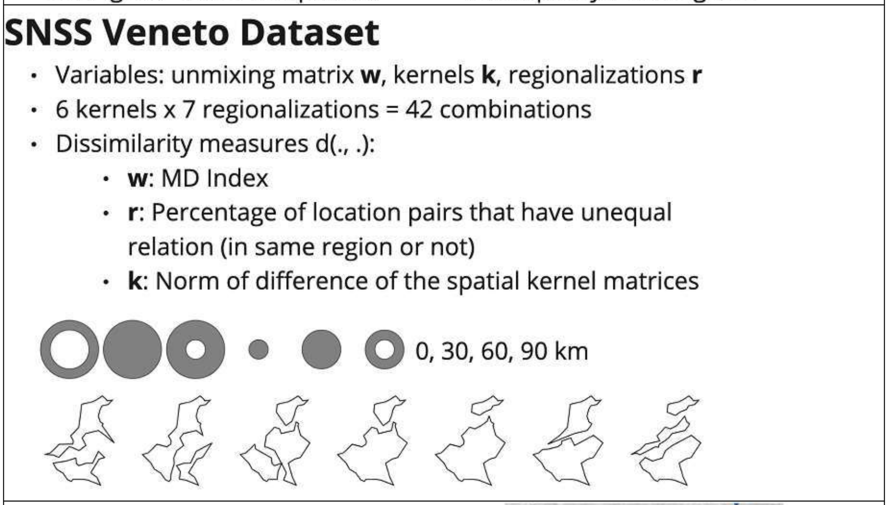
- Output visualization:

= = 1 set of components

= 1 component

color (dark is more)
= spatial autocorrelation (Moran's I)
~= how clustered are high/low values
~= how "good" is the component

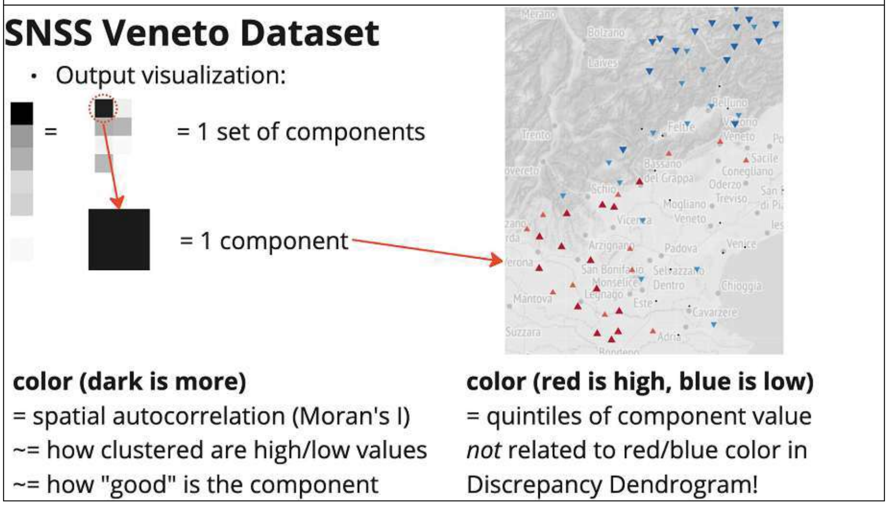
color (red is high, blue is low)
= quintiles of component value
not related to red/blue color in
Discrepancy Dendrogram!



SNSS Veneto Dataset

- Variables: unmixing matrix **w**, kernels **k**, regionalizations **r**
- 6 kernels x 7 regionalizations = 42 combinations
- Dissimilarity measures d(., .):
  - **w**: MD Index
  - **r**: Percentage of location pairs that have unequal relation (in same region or not)
  - **k**: Norm of difference of the spatial kernel matrices

0, 30, 60, 90 km



SNSS Veneto Dataset

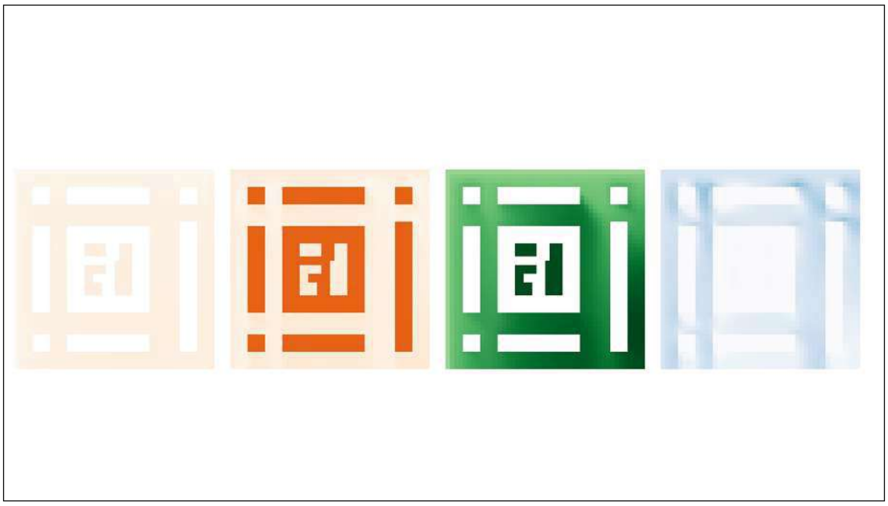
- Output visualization:

= = 1 set of components

= 1 component

color (dark is more)
= spatial autocorrelation (Moran's I)
~= how clustered are high/low values
~= how "good" is the component

color (red is high, blue is low)
= quintiles of component value
not related to red/blue color in
Discrepancy Dendrogram!



Online Material

All supplemental material for P5 may also be found on the open access web page of the article: https://doi.org/10.1109/TVCG.2023.3327203 (accessed 15th May, 2024). Supplemental material not reproduced in this thesis:

- Video Demo: https://doi.org/10.1109/TVCG.2023.3327203/mm1 (accessed 15th May, 2024).