JDSSV

Journal of Data Science, Statistics, and Visualisation

April 2024, Volume IV, Issue III. *doi: 10.52933/jdssv.v4i3.82*

Visual Interactive Parameter Selection for Temporal Blind Source Separation

Claudia Cappello
University of Salento, TU Wien

Nikolaus Piccolotto
TU Wien

Christoph Muehlmann
TU Wien

Markus Bögl
TU Wien

Peter Filzmoser
TU Wien

Silvia Miksch
TU Wien

Klaus Nordhausen
University of Jyväskylä

Abstract

Many fields of science and industry collect and analyze multivariate time-varying measurements, e.g., healthcare, geophysics, or finance. Such data is often high-dimensional, correlated, and noisy. Experts are interested in latent components of the dataset, but due to the properties above, these are difficult to obtain. Temporal Blind Source Separation (TBSS) is a suitable and well-established framework for these data. However, the wide choice of methods and their tuning parameters impede the effective use of TBSS in practice. Visual Analytics (VA) aims to create powerful analytic tools by combining the strengths of humans and computers. We designed, developed, and evaluated VA contributions in previous work to support TBSS-related analysis tasks. This paper highlights the benefits and opportunities of VA concepts for statistics-oriented problems. We demonstrate how their analysis workflow can be supported using an important TBSS application example with a real-world dataset of meteorological measurements in Italy.

Keywords: Time series, dimension reduction, model selection, visual analytics.

1. Introduction

Multivariate time series are commonly collected in many fields of science and industry. Geophysical time series are relevant, e.g., for meteorologists. Financial investors are interested in stocks and currency exchange rates. Medical professionals, on the other hand, often measure biomedical time series, such as electrocardiograms (ECG), magnetoencephalographies (MEG), or functional magnetic resonance images (fMRI). Common challenges in analyzing such datasets are correlated variables, noisy measurements, and the often high number of dimensions. Domain experts look for latent sources in the dataset. Such sources may be visually analyzed to learn about the measured phenomenon, e.g., recovering a fetus' heartbeat from its mother's ECG (de Lathauwer et al. 2000) to inspect it for diseases. Another use for latent sources is dimension reduction and modeling. Often, a few latent sources are sufficient to describe the structures and dynamics of the data, thus helping interpretability and future modeling. Modeling multivariate serial dependence is a challenge that benefits from reducing dimensions and decomposing the time series into components that could be modeled univariately. It is well known that ignoring serial dependence in dimension reduction might i) be inefficient and ii) miss important structures, indicating that standard multivariate methods, like principal component analysis (PCA), are not appropriate in such a context. On the other hand, blind source separation (BSS) is a general multivariate modeling technique with many different approaches, many of which are suitable for multivariate time series. However, these approaches often require selecting tuning parameters, making it crucial to compare different BSS approaches, which is quite challenging. BSS originated from signal processing but is nowadays a well-established multivariate method used in many application areas to gain better insights into processes driving the observed data.

However, even when a BSS model is specified, many different BSS approaches exist, which usually have tuning parameters that need to be specified by the user. A general challenge in BSS is that the models have many indeterminacies, and hence, it is difficult to compare the output of different BSS approaches or even the results from the same approach but using different parameter settings. Therefore, using BSS in practice is demanding, as choosing methods and tuning parameters is not straightforward. Further, data dimensionality and model indeterminacies are common challenges. Note also that, in general, not much research has been done concerning selecting these parameters. In a few cases, the focus was mainly on independent component analysis (ICA) in the context of multi-subject neuroscience experiments (Artoni et al. 2012, 2014), which might not easily transfer to other BSS models, methods, and applications.

Relief for these challenging tasks of visual analysis, interpretability, and modeling may come from Visual Analytics (VA). The general idea of VA is to combine the strengths and capabilities of humans with those of computers/machines by using visual representation and interaction techniques to enable effective communication between them. One definition of VA is by Keim et al. (2010), stating "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 datasets". By combining human reasoning with computers, VA provides methods, analytical workflows and software systems that combine these strengths and supply appropriate visual displays and interaction techniques so that communication between both sides can hap-

pen most effectively. VA solutions follow two popular mantras of VA and information visualization to achieve that goal: "Overview first, zoom and filter, then details-on-demand", the information-seeking mantra by Shneiderman (1996), and "analyze first - show the important - zoom, filter and analyze further - details on demand", the VA mantra by Keim et al. (2008).

Although many complex VA systems incorporate primarily basic visualizations curated within complex software environments enriched with powerful computation and analytics methods, they can also include more advanced data visualization techniques. Many of these are extensions or generalizations of basic data visualization techniques. For example, parallel coordinate plots (PCP) (Inselberg and Dimsdale 1990) can be considered multi-dimensional scatter plots, where parallel and equidistant axes represent multiple variables. A polygonal chain (polyline) represents a data tuple by linking the corresponding variable values (Aigner et al. 2023).

In prior work (Piccolotto et al. 2022a), the design and evaluation of such a VA solution for the temporal BSS method gSOBI (Miettinen et al. 2020) were discussed. However, the paper focused on visualization designs and data mining algorithms. It did not describe practical design decisions and considerations helpful in working with BSS. Neither did the paper describe a BSS- and VA-supported real-world data analysis scenario in detail. A tutorial-style document, which may be translated to other datasets, is needed to make VA-assisted BSS more prominent in modern data analysis tool kits. This paper intends to be that document, including the whole workflow from data preparation to parameter selection to interpretation of latent sources.

The paper is structured as follows. In Section 2, we recall various BSS models and reflect on the challenges of typical analysis tasks. Section 3 describes how custom interactive visualizations support these TBSS-related analysis tasks. In Section 4, we present a case study where we apply the interactive visualizations on a real-world environmental dataset from Italy. The benefits over plain R/RStudio are discussed separately in detail in Section 5, which concludes this paper.

2. Temporal Blind Source Separation

In general, BSS considers various statistical models for different forms of data, based on model-stringent estimators for the unmixing matrix. The basic BSS model states that

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{\mu}, \tag{1}$$

where $\boldsymbol{x}$ is a p-variate observable random vector, $\boldsymbol{z}$ a standardized latent p-vector with at uncorrelated or independent components, $\boldsymbol{A}$ is the non-random $p \times p$ mixing matrix and $\boldsymbol{\mu}$ is a non-random p-variate location vector. The goal of BSS is to estimate $\boldsymbol{z}$ based on $\boldsymbol{x}$ alone, i.e., to find an unmixing matrix $\boldsymbol{W}$ such that $\boldsymbol{W}(\boldsymbol{x} - \boldsymbol{\mu})$ recovers the latent components. The motivation of BSS is that the latent components are easier to interpret and model than the original variables, allowing deeper insights into the data. Clearly, the BSS problem is not solvable without further assumptions, and BSS approaches differ by the various assumptions they impose on $\boldsymbol{z}$. The most popular branch of BSS is ICA, where it is assumed that $\boldsymbol{z}$ consists of non-Gaussian independent components. However, many other BSS approaches target, e.g., time series or spatial data.

For general reviews, see for example, Nordhausen and Oja (2018); Nordhausen and Ruiz-Gazen (2022) and for a review focusing on BSS methods in the context of time series, see for example, Pan et al. (2022). In the following, we detail the commonly used model in TBSS in Section 2.1 and present the estimator. For this estimator, the parameters involved need to be set by the analyst. We highlight practical challenges and guidelines to support these choices in Section 2.3.

2.1. Independent Component Time Series Model

The model we will consider in more detail in this paper is the independent component time series model, which we denote by

$$\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{z}_t + \boldsymbol{\mu}, \quad t = 0, \pm 1, \pm 2, \ldots, \tag{2}$$

where $\boldsymbol{x}_t$ is an observable p-variate times series, the mixing matrix and location are as above and for the latent p-variate $\boldsymbol{z}_t$ the following assumptions are made:

IC1: $E(\boldsymbol{z}_t) = \boldsymbol{0}$ and $\mathrm{Cov}(\boldsymbol{z}_t) = \boldsymbol{I}_p$ for all t.

IC2: The components of $\boldsymbol{z}_t$ are independent.

We will refer to this model as the temporal blind source separation model. Such a model has been, e.g., considered for automatic artifacts removal for EEG data (Joyce et al. 2004), finding meaningful signals in brain imaging (Tang et al. 2000, 2005b; Tang 2010), monitoring buildings (Popescu and Manolescu 2007), wind speed forecasting (Firat et al. 2010), financial time series modeling (Nordhausen et al. 2021), or for analyzing the sediment cores from geological drilling projects (Bábek et al. 2022).

In TBSS, the location vector is usually not of interest. For the remainder of this paper, we assume without loss of generality that $\boldsymbol{\mu} = \boldsymbol{0}$. Moreover, the TBSS model is not well defined as the signs and order of the latent components are not fixed, i.e.,

$$\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{z}_t = (\boldsymbol{A}\boldsymbol{J}\boldsymbol{P})(\boldsymbol{P}^\top \boldsymbol{J}\boldsymbol{z}_t) = \boldsymbol{A}^*\boldsymbol{z}_t^*,$$

where $\boldsymbol{z}_t$ and $\boldsymbol{z}_t^*$ fulfill the model assumptions and yield the same process $\boldsymbol{x}_t$ for all $p \times p$ sign-change matrices $\boldsymbol{J}$ (diagonal matrices with ± 1 on its diagonal), and all $p \times p$ permutation matrices $\boldsymbol{P}$ (matrices with only zeros and ones and each row and column has exactly one value one). These indeterminacies are not considered a problem in practice. One of the reasons is that BSS is often used in a dimension reduction context, and it is assumed that only $q < p$ components are of interest. These q components are often hand-picked after visually inspecting all p components. Two approaches to formalizing such a dimension reduction approach that extends Model (2) have been suggested in the literature and are recalled as follows.

In the external noise model (ENM), one assumes

$$\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{z}_t + \boldsymbol{\epsilon}_t, \quad t = 0, \pm 1, \pm 2, \ldots, \tag{3}$$

where now $\boldsymbol{z}_t$ is q-variate following assumptions (IC1) and (IC2) while being independent of the p-variate white noise process $\boldsymbol{\epsilon}_t$. The mixing matrix $\boldsymbol{A}$ is accordingly $p \times q$. In the internal noise model (INM), on the other hand, one assumes that

$$\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{z}_t = \boldsymbol{A}(\boldsymbol{z}_{s,t}^\top, \boldsymbol{z}_{n,t}^\top)^\top, \quad t = 0, \pm 1, \pm 2, \ldots, \tag{4}$$

where $\boldsymbol{A}$ is the $p \times p$ mixing matrix and the q-variate signal of interest is $\boldsymbol{z}_{s,t}$ while the $p-q$-variate noise $\boldsymbol{z}_{n,t}$ is not of interest. Noise can be defined in many different terms as long as it is independent of the signal. Most commonly, it is modeled as a white noise process.

Note that Model (2) is not the classical ICA model, as there is no limitation on the distribution of the components as long as at least the first two moments are finite. Still, ICA methods are often used in this context, although they are not a natural choice as classical ICA methods are designed for independently identically distributed data. Therefore, ICA ignores information from serial dependence in the data. Further, in contrast to ICA, the TBSS model allows multiple Gaussian components.

2.2. gSOBI for TBSS

The method we will consider in the following to solve the TBSS problem is gSOBI (Miettinen et al. 2020) as it is a flexible method and has many other popular methods as special cases.

The gSOBI unmixing matrix $\boldsymbol{W}_{gS}$ is defined as the maximizer of

$$b \sum_{\tau \in \boldsymbol{k}_1} \sum_{i=1}^{p} (E(\boldsymbol{w}_i^\top \boldsymbol{x}_t \boldsymbol{w}_i^\top \boldsymbol{x}_{t+k}))^2 + (1-b) \sum_{\tau \in \boldsymbol{k}_2} \sum_{i=1}^{p} (E((\boldsymbol{w}_i^\top \boldsymbol{x}_t)^2 (\boldsymbol{w}_i^\top \boldsymbol{x}_{t+k})^2 - 1))^2,$$

under the constraint $\boldsymbol{W} \operatorname{Cov}(\boldsymbol{x}_t) \boldsymbol{W}^\top = \boldsymbol{I}_p$, with centered $\boldsymbol{x}_t$ and the parameter $b \in [0,1]$ and the sets $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$ which specify lag sets of interest. The vector $\boldsymbol{w}_i^\top$ denotes here the ith row of $\boldsymbol{W}$, $i = 1, \ldots, p$.

For specific values for b, $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$, special TBSS methods can be obtained.

- For $b = 1$ and $\boldsymbol{k}_1 = \{\tau_1\}$, one obtains AMUSE (Tong et al. 1990).
- For $b = 1$ and $\boldsymbol{k}_1 = \{\tau_1, \ldots, \tau_{\boldsymbol{k}_1}\}$, one obtains SOBI (Belouchrani et al. 1997; Miettinen et al. 2015, 2016).
- For $b = 0$ and $\boldsymbol{k}_2 = \{\tau_1, \ldots, \tau_{\boldsymbol{k}_2}\}$, one obtains vSOBI (Miettinen et al. 2020).

AMUSE and SOBI were developed with linear processes, like ARMA models, in mind, while vSOBI was developed in a framework of models exhibiting stochastic volatility features, like those present in GARCH models. SOBI can be seen as an extension of AMUSE by using more than one lag. It was recognized that choosing the optimal lag is difficult, and using several lags at once often gives better and more stable results. For more details about the three methods, see Appendix A. Therefore, "generalized" SOBI (gSOBI) can be seen as a linear combination of SOBI and vSOBI. The tuning parameter b gives either more weight to the first (SOBI) or the second part (vSOBI).

A user of gSOBI, therefore, has to choose b and which and how many lags should be included in the lag sets $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$. However, there are hardly any guidelines about choosing these three quantities, which greatly impact the results. Based on simulations, Miettinen et al. (2020) suggested to choose b somewhere above 0.5 but recommend values around 0.9, but they have no recommendations to choose $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$ except that they usually use a much smaller set of lags for $\boldsymbol{k}_2$ which seems however mainly due to the computational complexity.

For AMUSE and SOBI, some considerations about choosing the lags exist. Tang et al. (2005a) shows in the context of some neuroscience applications how crucial this choice is. Taskinen et al. (2016) suggest in the case of SOBI to choose a few different lag sets and then estimate for each of the sets the asymptotic covariance matrix of the estimated unmixing matrix; then choose as best set the one where the estimated covariance matrix has the smallest volume. However, this is already quite challenging for SOBI only and not feasible for gSOBI. Therefore, the general recommendation is to use sets of several different lags so that at the lags under consideration, the behavior of the latent time series is as different from each other as possible.

For proper use of gSOBI, our approach in the following is to use VA concepts to enhance the possibility of finding meaningful parameters for the huge set of possible combinations of the parameters b, $\boldsymbol{k}_1$, and $\boldsymbol{k}_2$.

Before outlining the strategy in the following section we would like to recall that an ENM or INM model is often of interest and that another decision must be made here. The general idea for ENM is that first, a classical principal component analysis would be performed where the first q components contain all the information for the TBSS, which is then subsequently performed. The reasoning is that for the population covariance matrix, the last $p-q$ eigenvalues equal the noise variance σ^2 while all other eigenvalues are larger. However, how to choose q based on the sample covariance matrix is still an open question that is mostly studied in the literature in an independent identically distributed framework (Wax and Kailath 1985; Zhao et al. 1986; Virta and Nordhausen 2019; Nordhausen et al. 2022), and due to lack of better methods, these are sometimes applied to a time series context. Scree plots seem most commonly applied to choose q. The INM is studied a bit better in the context of SOBI under the assumption that the noise is white noise. Then Matilainen et al. (2018); Virta and Nordhausen (2021) provide strategies to decide upon and select the signal components. Such methods are, however, not yet developed for vSOBI or gSOBI.

2.3. Some Thoughts on Practical Strategies and Tools Involved

In a typical application where gSOBI would be applied, the time series are many, and the sample size is large. The practitioner is mainly interested in achieving dimension reduction by performing TBSS and selecting a small number of meaningful, i.e., interpretable, components that sufficiently explain the data at hand. From our personal experience, we know that this is a tricky task. Many challenges impede thoroughly investigating the dataset and finding meaningful parameters, which we describe in the following paragraphs. As a consequence, usually only a couple of fits with different parameter settings are crudely compared, and subjectively, the best one is selected.

Comparing Fits Using Indices. As the signs and order of the components are not fixed, it is difficult to compare the different estimates/fits. To compare mixing matrices, one can use BSS performance measures (Nordhausen et al. 2011), which were developed for simulations where the mixing matrix $\boldsymbol{A}$ is known. The idea here is that for a perfect unmixing matrix, the product $\boldsymbol{WA}$ should be a scaled permutation matrix. Performance measures calculate the distance of $\boldsymbol{WA}$ to the target. Different measures

are available, and our preferred choice is the minimum distance (MD) index (Ilmonen et al. 2010), which is defined as

$$MD(\boldsymbol{W}, \boldsymbol{A}) = \frac{1}{\sqrt{p-1}} \inf_{\boldsymbol{C} \in \mathcal{C}} ||\boldsymbol{C}\boldsymbol{W}\boldsymbol{A} - \boldsymbol{I}_p||,$$

where $\mathcal{C}$ corresponds to the set of matrices where each row and column has exactly one non-zero element. The index takes values between 0 and 1, where 0 means a perfect separation. In a concrete analysis, $\boldsymbol{A}$ is, of course, unknown. Two competing unmixing matrices, $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ can be compared by computing $MD(\boldsymbol{W}_1, \boldsymbol{W}_2^{-1})$, which should be close to zero if they separate the components equally well. In our experience, extreme values of the MD index are rarely obtained, even with small changes in tuning parameters. Much more commonly, the index takes values around 0.4–0.8, which are not as easily interpreted. The index value then allows only relative assessments (e.g., $\boldsymbol{W}_1$ separates more like $\boldsymbol{W}_2$ than $\boldsymbol{W}_3$), and visual comparison is necessary to find the exact differences.

Comparing Fits By Their Loadings. The unmixing matrices are important when deciding about the interpretability of the independent components as they define the linear transformation from the data to the latent components. As the transformation is linear in its nature, the unmixing matrix can be interpreted as a loading matrix similar to PCA or factor analysis. Thus, practitioners can compare different loadings matrices and decide how well they characterize certain processes.

Comparing Fits By Their Components. Another approach to compare competing fits is to compare the resulting components. Again, the challenge is that they might be in a different order, with different signs, and generally not exactly the same.

One possibility is to order the components from all fits according to some general index of "interestingness" like kurtosis, skewness, autocorrelation measures, or stochastic volatility. If the index depends on signs, it might also be meaningful to fix first the signs of all components such that they are all right-skewed. Another option is to use absolute values of the index if the index reflects the sign (e.g., skewness). Ideally, such indices will sort similar components to the same positions, so visual inspection of the first few components from both fits should select relevant components. Of course, that cannot generally be assumed. Correlation matrices of components do not depend on component order, but they are reductive. Thus, inspection of matching components is necessary. More sophisticated visual support could make comparing matching components in alternative fits more efficient.

A clustering algorithm could also group components from all fits. Here, it is crucial that from each fit one component should be linked to the most similar component from *another* fit. The idea here is that, especially in a dimension reduction framework and under noise assumption, it could be assumed that the real signal components appear in all fits, while the noise components, as they are not well-defined, most likely differ between the different fits.

Selecting Lag Sets. Yet another challenging part is the selection of the lag sets. Many lags and combinations thereof are possible, and humans cannot reason about

all of them at once. In VA, a common strategy to deal with such amounts of data is quantification. Let $\text{ACov}_\tau(\boldsymbol{x}_t) = E\left((\boldsymbol{x}_t - E(\boldsymbol{x}_t))(\boldsymbol{x}_{t-\tau} - E(\boldsymbol{x}_{t-\tau}))^\top\right)$ denote the autocovariance matrix of $\boldsymbol{x}_t$ at lag τ. Then, the first idea is to keep those lags where the diagonal elements of $\boldsymbol{W}\,\text{ACov}_\tau(\boldsymbol{x}_t)\boldsymbol{W}^T$ have some big gaps as this means here the information between the latent components differs the most. Those lags where the diagonal elements are all very similar would be considered non-informative. This procedure works best for AMUSE/SOBI. For decision-making with real-world data, this concept should be extended to vSOBI and gSOBI and involve also higher order moments. One idea to do this is to consider the matrix of fourth cross-moments $\boldsymbol{B}_\tau$ which is defined for a centered time series as

$$\boldsymbol{B}_\tau(\boldsymbol{x}_t) = E(\boldsymbol{x}_{t+\tau}\boldsymbol{x}_t^\top\boldsymbol{x}_t\boldsymbol{x}_{t+\tau}^\top)$$

Under (IC1) and (IC2) for all lags τ, it holds that $\boldsymbol{B}(\boldsymbol{z}_t) = \boldsymbol{D}_\tau^*$ where $\boldsymbol{D}_\tau^*$ is again a diagonal matrix where the diagonal elements depend on τ (Matilainen et al. 2015). Thus, similar to the SOBI part now for the stochastic volatility part lag selection, the interesting or non-interesting lags can be based on the gaps between in the diagonal values of $\boldsymbol{W}\boldsymbol{B}_\tau(\boldsymbol{x}_t)\boldsymbol{W}^\top$. Also, as mentioned above, under the model, $\boldsymbol{W}\,\text{ACov}_\tau(\boldsymbol{x}_t)\boldsymbol{W}^\top$ and $\boldsymbol{W}\boldsymbol{B}_\tau(\boldsymbol{x}_t)\boldsymbol{W}^\top$ should be diagonal for all τ. However, for finite samples, estimated counterparts will hardly ever be diagonal, and an investigation regarding influential lags can also be based on the magnitude of the off-diagonal elements of the different matrix and lag combinations.

Summary. When selecting b, $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$ for gSOBI, keeping an overview over all these criteria is quite challenging, especially in the framework of R (R Core Team 2022) and RStudio (RStudio Team 2022), where the changes are not easily immediately visualized and measured. The following section introduces the VA software tool TBSSvis, which provides customized interactive visualizations for the TBSS-related tasks discussed in this section.

3. TBSSvis

TBSSvis (Piccolotto et al. 2022a) is an interactive VA software tool to support TBSS analysis in the context of gSOBI. It is available on GitHub (Piccolotto 2022), where detailed instructions on how to run TBSSvis are included. A video demo (using another dataset than that presented in Section 4) can be found on YouTube[1]. TBSSvis consists of a backend server written in R using **plumber** (Schloerke and Allen 2022) that handles all necessary computations. The outcomes of those are presented in a web-based frontend built using JavaScript (MDN Contributors 2022) and **D3** (Bostock et al. 2011). We designed TBSSvis for multivariate time series with up to length $T = 5\,000$ and dimension $p = 20$. These limits are not strict, but the convenience of using TBSSvis will gradually decrease with datasets beyond those.

TBSSvis was developed in a user-centered design process with (T)BSS experts over several months. Continuous discussions of what information to present when were accompanied by prototypes of increasing fidelity from paper sketches to digital mockups

[1] https://www.youtube.com/watch?v=iv919b12gek

to the final interactive software tool. TBSSvis may be understood as a visual interface to gSOBI. We chose gSOBI because of its flexibility and recency. TBSSvis was extensively evaluated in interviews with five (T)BSS experts who used the software tool on familiar datasets. They found TBSSvis to be a time-saver and very useful for applied work. The VA aspects in the design of TBSSvis are discussed in detail in Piccolotto et al. (2022a). We will present in this section only the most important views, visualizations, and concepts.

TBSSvis supports two main analysis phases crucial to TBSS model selection: parameter selection and result comparison (Section 2.3). It is possible to carry these out in existing tools, like R/RStudio, but it is not ideal. Due to its open-ended and exploratory nature, TBSS analysis is both i) visual as obtained components need inspection and comparison and ii) dependent on supporting data mining algorithms, e.g., clustering. Visual analytics is the science of how to combine these two aspects effectively. Hence, VA software tools present carefully designed interactive visualizations that efficiently and effectively support user tasks. R/RStudio, on the other hand, is a general-purpose programming environment and thus requires manual programming for visualizations and data mining algorithms. Further, it is not (without additional effort) possible to interactively modify these plots in ways relevant to the question at hand. Time spent on manual programming draws focus from the analysis. An advantage of R is its flexibility: Data may be transformed at will, and any available package may be used. Hence, any used parameter settings, latent components, and models obtained with TBSSvis can be exported to `RData` files and easily imported to R for further analysis.

3.1. Visual Analytics Basics

Like most VA contributions and software tools, TBSSvis uses basic data visualizations known from statistical graphs and exploratory data analysis. These graphs include scatter plots, line plots, bar charts, and so on. The major advancement stems from so-called *coordinated and multiple views* (CMV); user interaction techniques to link and highlight between multiple juxtaposed views (Roberts 2007). Each view may contain single or multiple data visualizations, whether basic or advanced. For such CMV, common interaction methods make such VA approaches useful and powerful. The most important include direct manipulation, brushing & linking, and dynamic queries (indirect manipulation). *Direct manipulation* allows users to interact directly with the graphical elements within each plot, e.g., to filter or select. *Brushing & linking* means that other views update after direct manipulation actions in one view, e.g., a user selects dots in a scatter plot which causes highlighting/selection of the same observations in all linked views that contain them (Roberts 2007). *Dynamic queries* are related to brushing & linking as they allow focusing on data of interest, e.g., by filtering the data. The difference to brushing & linking is that dynamic queries are an indirect manipulation approach, where users interact with sliders, menus, or buttons.

Because screen space is still limited and multivariate data can be large in observations and dimensions, it will often be difficult to show every data point in basic or multiple plots. Therefore, various computation and analytics approaches are often utilized for generating visualizations, e.g., aggregations, clustering, or similarity projections. Aggregations include, e.g., correlation matrices, histograms, or summary statistics.

Clustering is beneficial for finding groups of observations, and analysts may then reason about group representatives instead of individual items. Similarity projection, on the other hand, is commonly employed to generate 2D/3D visualizations of multivariate data for visual cluster analysis. Similar observations appear close to each other in the similarity projection. All these techniques are necessarily reductive as they do not show the full data, but they allow analysts to get an initial overview. Necessary screen space is commonly saved by hiding details at first and showing them only after user interactions. For example, when a multivariate time series is shown as individual line charts, each chart does not need to include the (identical) legend. Showing it on hover/mouse-over is sufficient. In the same spirit, it is common to make charts resizable (e.g., increase Y-axis space for line charts) or zoom into regions of interest (e.g., a contiguous time interval) with user interactions. Analysis on different levels of detail is thus facilitated.

3.2. Parameter Selection

The parameter space of gSOBI is gigantic as 2^{T+1} possible lag sets exist for a time series of length T. Hence, TBSSvis focuses primarily on those. TBSSvis employs two strategies to deal with lags: Organization and quantification. Parameter b may be chosen interactively with a slider.

TBSSvis relates each lag to the nearest calendar interval to allow the organization of lags. In other words, lags $24, 168, 720$ of hourly intervals become one day, one week, and one month. The benefit is two-fold. For one, it is a much more natural representation of time for humans. Second, it allows filtering lags based on temporal granularity, i.e., showing lags corresponding to weeks, months, or years. Doing so drastically reduces the amount of lags to think about.

However, the longer the time series at hand and the shorter its measurement interval, the more lags are to consider, even after filtering. Here is where the second strategy, quantification, comes in. The idea is to assign each lag a number indicating how interesting, relevant, or important it is. TBSSvis computes three such numbers. First, the autocorrelation of each input variable at all lags. Where input variables correlate highly, latent components might do that as well. Second, the eigenvalue difference of autocovariance matrices at all lags. This measure is supposed to be high when calculated on latent components, so the anticipation is that this might also hold for input variables for some datasets. Finally, we compute the diagonality of cross-moment matrices at all lags (for a given gSOBI solution). A better separation might be obtained by favoring lags where this diagonality was low. See Section 2.3 for detailed statistical considerations.

Figure 1 shows the relevant visualizations for lag selection. The parallel coordinates plot (PCP) contains the previously described quantified dimensions for each lag. It can be filtered to a given temporal granularity with the radio buttons on top. A subset of lags is selected by brushing along an axis. The available axes are the lags themselves (allows filtering for low/high lags directly), the temporal granule selected (allows filtering for low/high lags in relation to the calendar), and the quantified numbers described in the previous paragraph. A selection is depicted in more detail in the plot underneath the PCP (linking), which is a multivariate autocorrelation function plot (MACF). Each

box in the MACF represents the autocorrelation of all input variables (bars) at a given lag (box). Color saturation encodes the lag length, with less saturation meaning longer lags. Bars may be ordered by variable (alphabetically) or by value. The latter is the default, making predominantly positive or negative correlations of variables easily visible. By hovering over a lag in the MACF, it is highlighted and may be inspected in even greater detail with the line chart and scatter plot below (brushing & linking). The line chart shows a user-selected input variable, while the scatter plot shows that variable vs. itself at the currently highlighted lag. The lag length is shown as a horizontal line in the line chart, further simplifying reasoning about the lag. These multiple views are intended to be used in sequence, where the analyst first filters all lags to a subset (PCP, Figure 1-A), inspects the subset more closely to identify individual lags as candidates to include in a parameter setting (MACF, Figure 1-B), and finally inspects (line chart and scatterplot, Figure 1-C) and selects individual lags (Figure 1-C and D).

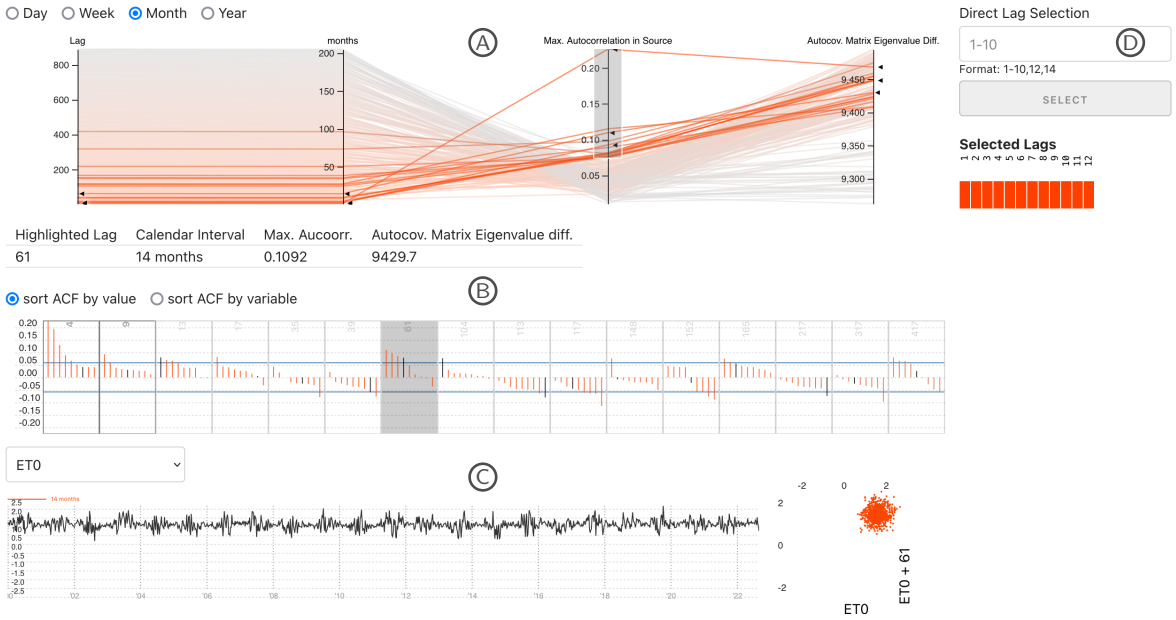


Figure 1: Lag Selection view of TBSSvis. A parallel coordinates plot allows to filter lags to those that are interesting, e.g., those where an input variable exhibits the highest autocorrelation (A). A multivariate ACF plot shows input variables at filtered lags (B). Individual lags may be highlighted with hover, thus updating the scatter plot underneath (C). Interaction may be circumvented by direct selection of lags with R-like syntax (D).

3.3. Result Exploration

To assess candidate TBSS models qualitatively, analysts need to compare their results and parameter settings on various levels of detail. Such comparisons often start at a very high level, where analysts are interested in the TBSS method used (SOBI, vSOBI, or gSOBI) and a coarse description of used lag sets (mostly short, mostly long, or something else). Attention then shifts to the components, where analysts want to know which are common to many models and which were obtained only by a few. Having identified those, very detailed comparisons become more relevant. E.g., which lags were

used in one model but not the other, how do individual components of selected models compare, and what are the differences in their loadings. TBSSvis provides bespoke interactive visualizations to support these tasks. We will introduce them here and demonstrate their use in Section 4.

Model Overview. An overview of candidate models may be obtained from a tabular representation (Figure 2). Each model is automatically named by a unique identifier. Two columns hold the $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$ lag set, respectively, shown as a histogram with five bins. A column between the two shows the b parameter as a triangle on a line. The table is sorted by b. Gray background in the first column encodes whether the gSOBI implementation converged and latent components were obtained.

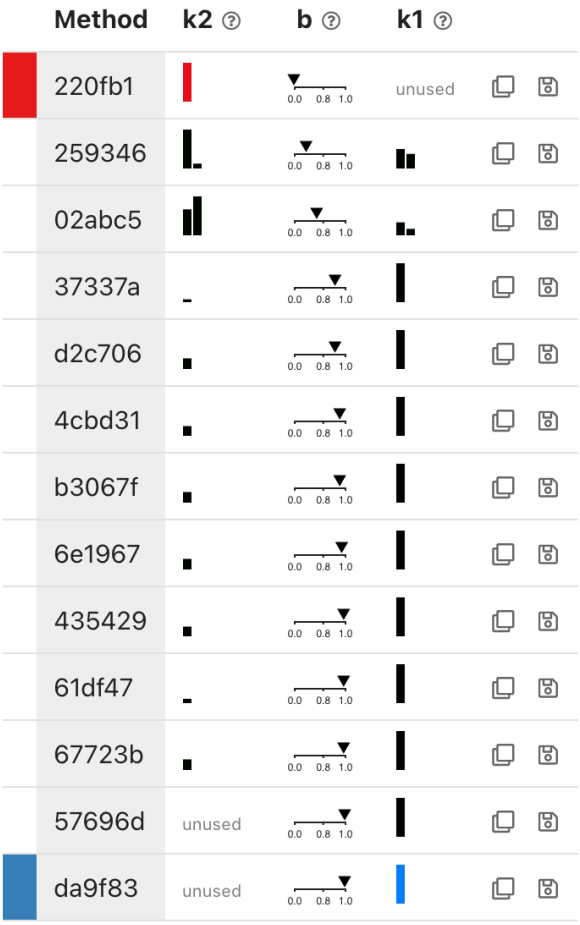


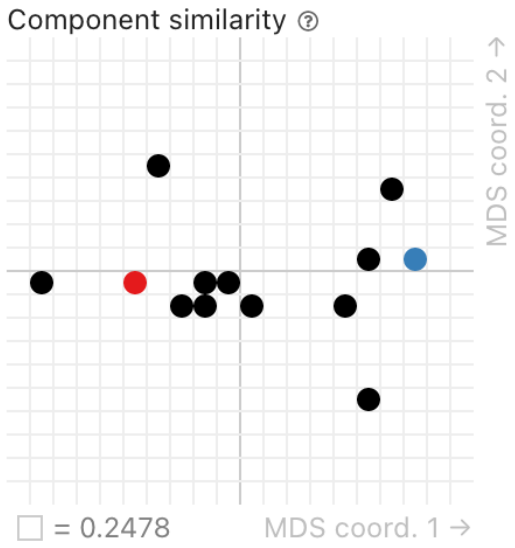
Figure 2: Tabular overview of alternative models. Gray background color shows that the gSOBI implementation converged for this parameterization. The extreme cases SOBI (blue) and vSOBI (red) were selected. Histograms in the $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$ columns show bins with a width of 500 lags.

Component Overview. Equally necessary is an overview of obtained components (Section 2.3), which TBSSvis provides in two ways (Figure 3). First, a similarity projection shows the similarity of models via their MD index (Ilmonen et al. 2010). Dots

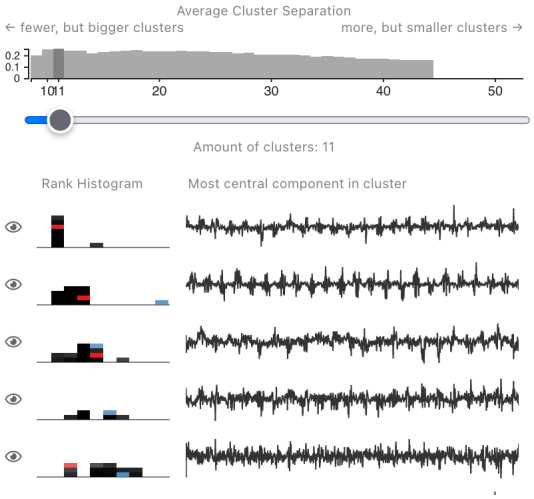
close to each other represent models with similar unmixing matrices and, thus, latent components. The plot is discretized to avoid occlusion (Figure 3a).

Second, TBSSvis applies a custom clustering scheme (Piccolotto et al. 2022a) that respects the set structure of latent components, unlike popular clustering algorithms such as k-means. Specifically, the clustering scheme groups by component similarity (quantified by absolute Pearson correlation) but only groups components from distinct models into the same cluster (Figure 3b). It builds upon constrained clustering techniques, such as Wagstaff et al. (2001). *Cannot-link* constraints, i.e., pairs of elements that must not be grouped into the same cluster, are added for all pairs of latent components of the same model. TBSSvis uses a k-medoids-like formulation of the problem for increased generality as k-means requires distances with triangle inequality, and its cluster representatives are not actual observations. TBSSvis iteratively adapts an unconstrained k-medoids solution obtained from the **cluster** R package by moving components to the next-best cluster as long as constraints are violated. Pseudocode for the set-aware clustering scheme can be found in Piccolotto et al. (2022a).

Representatives of these clusters are ordered by a user-selected quantification strategy (Section 2.3) and plotted in a list. TBSSvis provides kurtosis, absolute skewness, and periodicity (Vlachos et al. 2005). The boxes left of the cluster representatives represent components in that cluster. The boxes' opacity encodes variation in the cluster by difference to the cluster medoid. The horizontal position of the box encodes the component's order according to the selected feature compared to other components from the same model. Thus, very saturated boxes mark little variation in the cluster. The number of clusters (k of k-medoids) can be adjusted with a slider above. A bar chart on top of the slider depicts the clustering quality. Its peak thus marks the most likely number of meaningfully different components obtained in all models.



(a) Similarity projection of latent components with Multidimensional Scaling.



(b) Set-aware clustering of latent components (top 5 components according to kurtosis).

Figure 3: Overview of latent components obtained by different models.

Tuning Parameter Comparison. To allow comparison of tuning parameter settings of different models, we use triangles on a shared line for b. Lag sets can be compared via interweaved histograms, a visualization specifically designed for that purpose. By construction, distinct lag sets appear interweaved (few shared columns), while similar lag sets do not and display many shared columns. The user can regulate the amount of detail via the bin size.

Component Comparison. Detailed comparison of components is supported by plotting involved components side-by-side, one column per model. A slope graph may be added between columns, connecting similar components of adjacent models (Figure 4). Line thickness encodes three classes of similarity (quantified by absolute Pearson correlation), and no line is visible for component pairs with an absolute correlation below 0.5. The number of singular and thick lines (similar components) or many and thin lines (dissimilar components) makes it easy to distinguish the common parts from the rest.

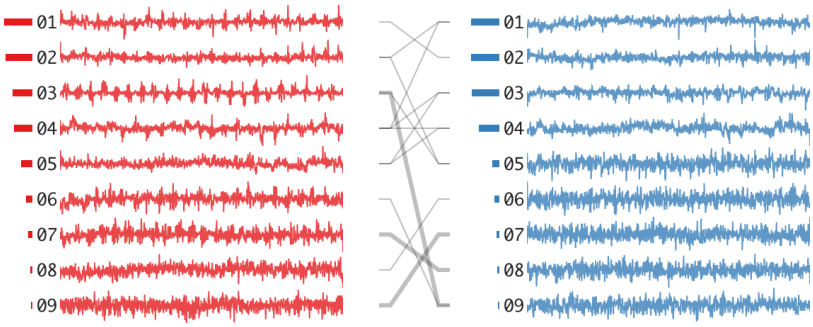


Figure 4: Slope graphs support the comparison of components from alternative models. Thick lines connect the most similar pairs, in this case 3-9, 7-8 and 9-7 of the red and blue solution, respectively. The Pearson correlation coefficient is shown on mouse-over. Bars left of the numbers encode the value of the component's quantified interestingness, e.g., kurtosis.

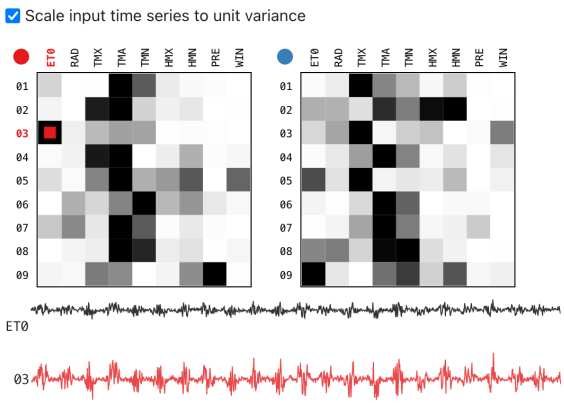


Figure 5: Visualization of unmixing matrices. Loadings are visible on mouse-hover. The third component of the red solution and the ET0 input variable were selected for visual comparison.

Unmixing Matrix and Loadings Comparison. Loadings are crucial for the interpretation of latent components. TBSSvis visualizes the unmixing matrix in a matrix visualization (Figure 5) to identify important input variables, where rows correspond to latent components and columns to input variables. Each row is color-coded separately with a grey color scale, where black marks input variables with the highest absolute value. Involved components and input variables are shown below after a cell is selected.

4. Case Study

This part examines an environmental dataset with the introduced TBSS methods and TBSSvis. All subsequent results are obtained with TBSSvis, version v1.0.1. The backend server runs R version 4.0.3 (R Core Team 2022) with the packages summarized in Appendix B.

4.1. Description of the Dataset

The data analyzed in the present case study consist of evapotranspiration levels (mm), solar radiation (MJ/m^2), minimum, average, and maximum temperature ($°C$), minimum and maximum humidity (%), wind speed (m/s) and precipitation (mm) weekly averages for 23 years (2000–2022) with a total of 1 184 temporal observations (Table 1). Note that evapotranspiration is a synthetic variable based on temperature-related variables and solar radiation (Hargreaves and Samani 1985). The measurements refer to one station belonging to the monitoring system of Veneto Meteorological Service[2] located in the Belluno district (Italy) at a high altitude (1 642 m above sea level). Moreover, the lowest weekly mean values occurred in the winter period and the highest mean values in the summer period. Hence, we removed the annual periodicity and used the weekly averages computed by the `decompose` R function (Kendal 1983) before applying BSS methods, which requires the stationarity of the time series.

Table 1: Description of the nine climate and meteorological variables

Name	Description
ET0	Evapotranspiration
RAD	Radiation
TMX	Maximum Temperature
TMA	Average Temperature
TMN	Minimum Temperature
HMX	Maximum Humidity
HMN	Minimum Humidity
PRE	Precipitation
WIN	Wind Velocity

[2]The raw data were obtained via a request to the Environmental Protection Agency of Veneto Region, who collects climate and weather data from 180 monitoring stations (https://www.arpa.veneto.it/bollettini/storico). In particular, the evapotranspiration levels were estimated according to the Hargreaves equation (Hargreaves and Samani 1982).

Appendix C contains an R script that prepares the considered dataset in CSV format. Figure 6 depicts the whole considered deseasonalized dataset in detail. From a holistic point of view, ET0 and RAD show strong oscillations during summer, whereas the values seem more stable during winter. In contrast, the opposite behavior is observed in the humidity (HMN and HMX). These effects hint at the existence of volatility in the observed time series. The temperature-related variables (TMX, TMN, and TMA) naturally show the same patterns overall, with the minimum temperature having the strongest oscillations. Interestingly, the temperature variables show a different behavior between 2006 and 2008. The winter of 2007 especially shows high values, meaning the winter was unusually warm. On the other hand, the summers of 2006 and 2007 were unusually cold, which is needed to ensure a constant overall annual mean temperature. Note that the nine variables are on different scales, which needs to be accounted for when comparing the loadings for the different BSS methods.

In what follows, the different steps required in finding gSOBI tuning parameter settings are described. In particular, after a first evaluation of the SOBI and vSOBI results, the user has to set usable lags by evaluating the MACF plots and usable values of the parameter b. Finally, the alternative gSOBI solutions have to be compared.

4.2. Comparing SOBI and vSOBI

In the first analysis step, the results of the SOBI and vSOBI methods are compared with the default values as they give a first impression of the features detected by the BSS based on sole second-order and sole fourth-order dependence. In particular, the first 12 lags ($\boldsymbol{k}_1 = \{1, \ldots, 12\}$) and the first 3 lags ($\boldsymbol{k}_2 = \{1, 2, 3\}$) are used for SOBI and vSOBI, respectively. In Figure 7, the latent time series for SOBI and vSOBI are depicted. The MD index between both loadings matrices equals 0.7, indicating substantially different loadings for both methods.

For the SOBI results in blue (Figure 7), only the last five time series seem to oscillate around the zero mean evenly and might thus be viewed as non-informative white noise. The first SOBI component has substantial outliers for 2006, 2012, 2014, 2016, and 2018 around the new year. The loadings both for the raw and scaled data indicate that this component is mainly formed by the temperature-related variables (i.e., TMX, TMA, TMN). In particular, the raw loadings weight TMX to 1.4, TMA to -0.9, and TMN to -0.6 which can be interpreted as the difference between the maximum and the sum of the average and minimum temperature. Hence, the positive outliers are mainly formed by a higher average maximum temperature, and the negative ones by the opposite effect.

The first and the third series show strong time-varying oscillations for the vSOBI latent time series in red (Figure 7). Large oscillations are evident during winter for component one and during summer for component three. By inspecting the raw and the scaled loadings (left panel of Figure 8b) the third latent component is mainly formed by ET0, which shows the same oscillations during summer (see first time series in Figure 6). More interesting is the pattern of the first latent component for the vSOBI result (Figure 7): the raw loadings (left panel of Figure 8b) hint that ET0, TMA, and TMN form this component with values of -5.3, 2.0 and -1.4. These loadings can be translated to the difference between the evapotranspiration and the deviation of the minimum

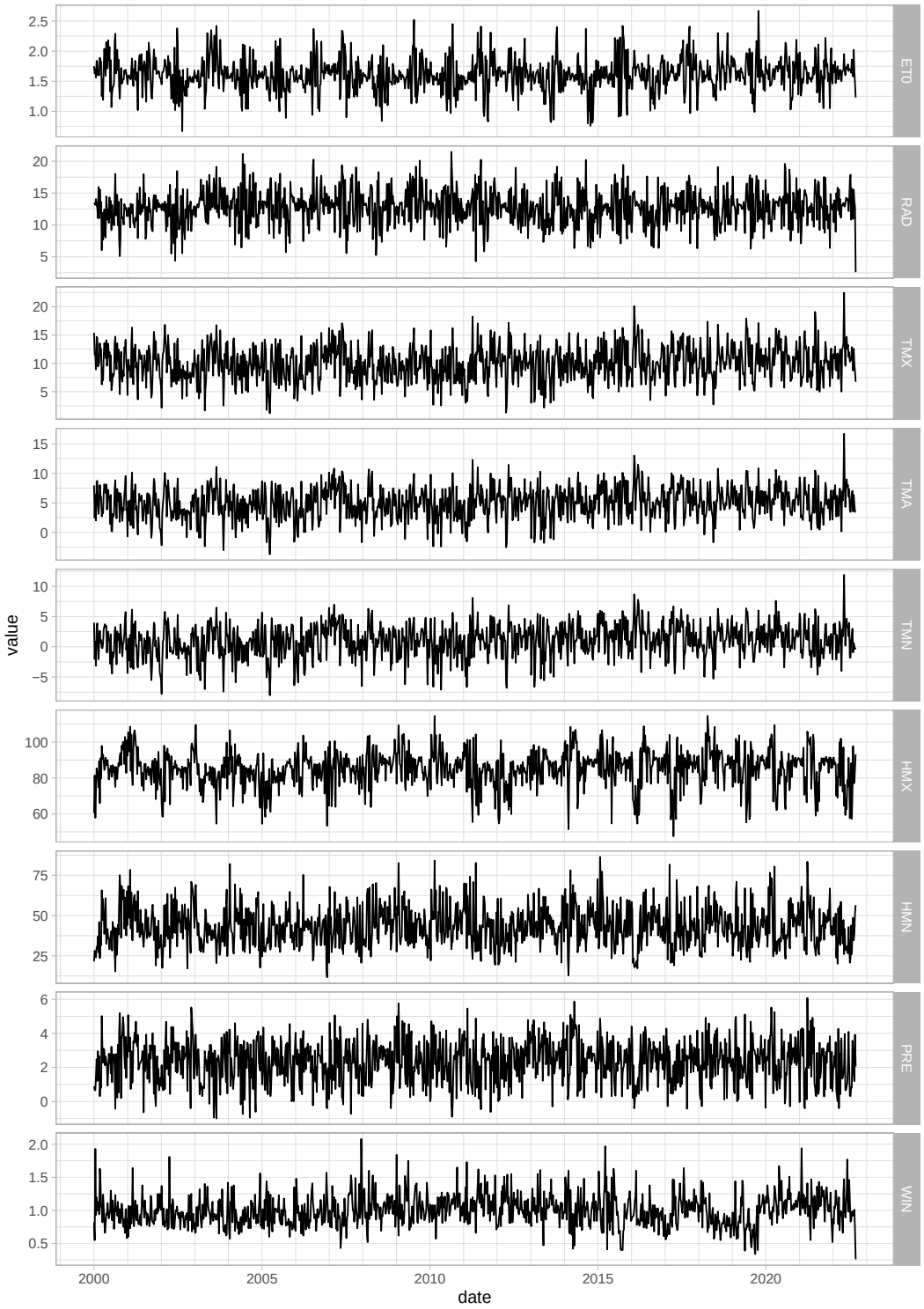


Figure 6: Overview of the nine climate and meteorological time series.

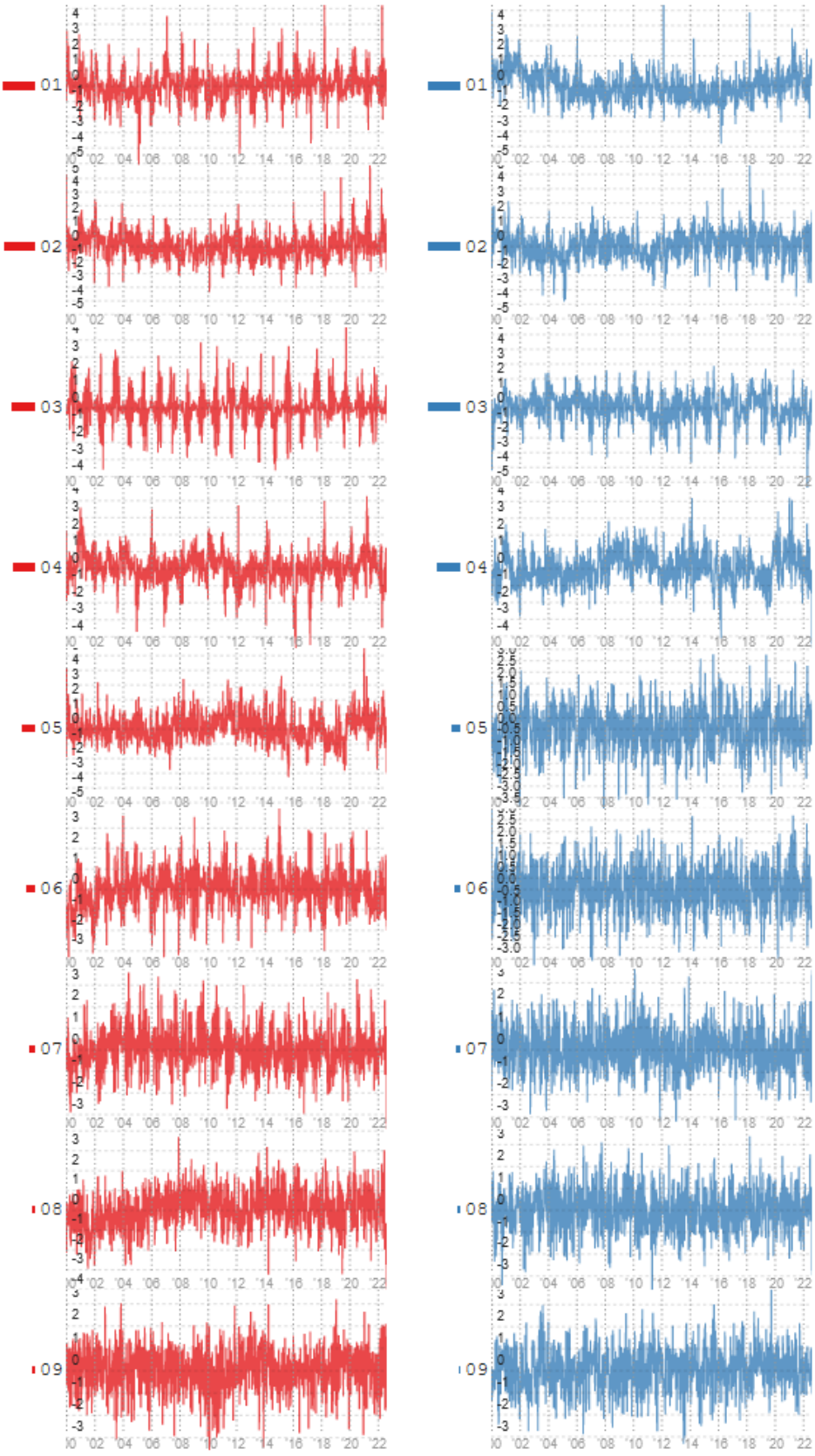
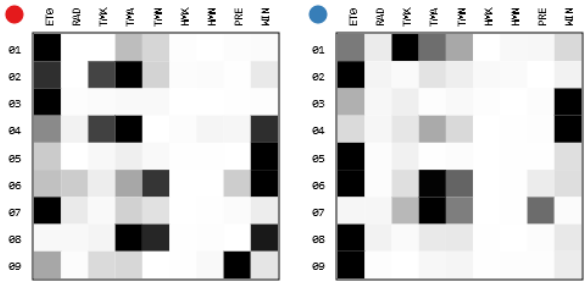
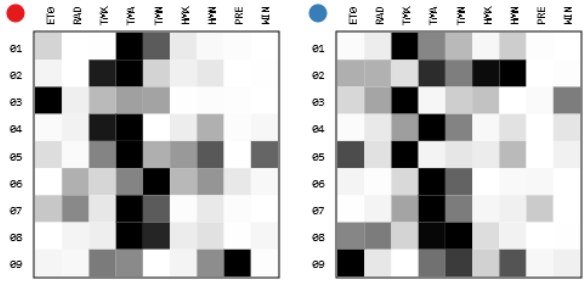


Figure 7: Latent time series for vSOBI with parameters $\boldsymbol{k}_2 = \{1, 2, 3\}$ (red) and for SOBI with parameters $\boldsymbol{k}_1 = \{1, \ldots, 12\}$ (blue). The order is according to the kurtosis values.



(a) Loadings matrices for the raw data



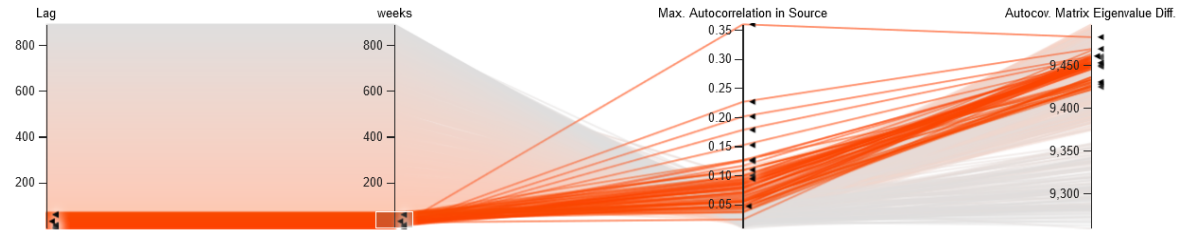
(b) Loadings matrices for the scaled data

Figure 8: Loadings matrices for vSOBI with parameters $\boldsymbol{k}_2 = \{1, 2, 3\}$ (red) and for SOBI with parameters $\boldsymbol{k}_1 = \{1, \ldots, 12\}$ (blue).

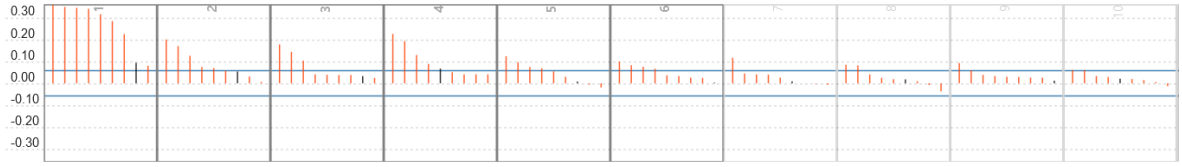
from the average temperature. This deviation is very unstable in the summer months and very stable in the winter months. Note that this relevant pattern was not observed in the original variables ET0, TMA, and TMN. In total, both SOBI and vSOBI results deliver meaningful latent components with the default set of lags. Hence, the gSOBI variant with a customized lag setting might gain more insights.

4.3. Lag Selection

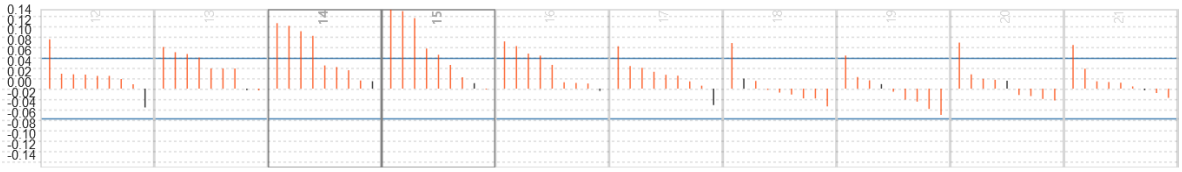
The default lag selection might not be suitable for every situation as they are based on the vague assumptions that the first few lags are the most informative. Nevertheless, they act as an educated first guess. One strategy to select usable lags is to investigate the MACF function and choose lags where several variables show substantial autocorrelation. Figure 9a depicts the PCP, which hints that the maximum autocorrelation is found in the shortest lags. The ACF values for the input variable for the first ten lags are shown in Figure 9b. The blue lines depict the 95% confidence interval for the case of pure white noise. Hence, values that are outside these lines can be considered to exhibit significant autocorrelation. By evaluating the MACF plots, we chose the lags $\boldsymbol{k}_1 = \{1, \ldots, 6, 14, 15, 59, 60, 61\}$ as these seem to show substantial autocorrelation and a high difference in the eigenvalues of the autocovariance in several variables. The last three lags of the determined set (59–61 weeks) very roughly correspond to a time separation of one year, which might indicate that the dataset contains annual effects. For the lag set $\boldsymbol{k}_2$, we keep the first three weeks as already substantial oscillations were found.



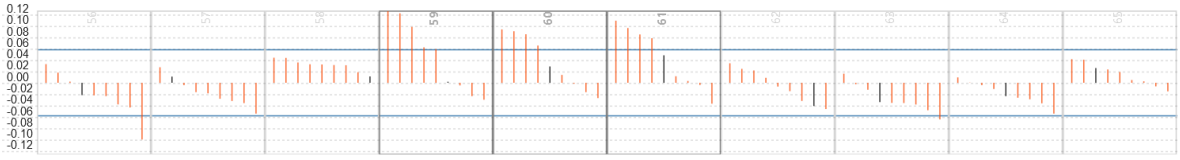
(a) PCP depicting the lags, corresponding maximum autocorrelation and the ACF eigenvalue differences in the data.



(b) MACF for lags 1–10



(c) MACF for lags 12–21

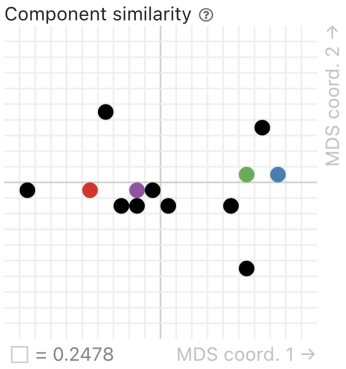


(d) MACF for lags 56–65

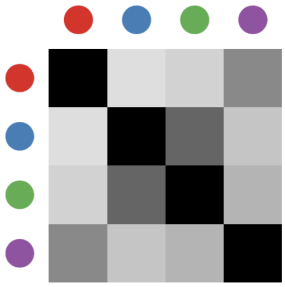
Figure 9: PCP and MACF plots for the environmental dataset under study.

4.4. Selection of the Parameter b

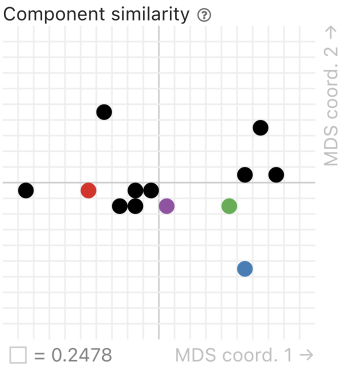
Parameter b determines the ratio between SOBI ($b = 1$) and vSOBI ($b = 0$). As vSOBI accounts for fourth-order temporal dependence, it will automatically surpass the second-order dependence of SOBI (power of two versus power of four). Thus, to equally distribute between SOBI and vSOBI, values of b close to one will be needed.
To investigate the effect of the parameter b, we keep the $\boldsymbol{k}_1$ either to the default ($\boldsymbol{k}_1 = \{1, \ldots, 12\}$) or the custom one ($\boldsymbol{k}_1 = \{1, \ldots, 6, 14, 15, 59, 60, 61\}$), $\boldsymbol{k}_2$ lag set unchanged and set b to the values 0.9, 0.95, 0.97 and 0.99 and compare the results to the sole SOBI and vSOBI cases. Note that in the case study, only values of the parameter b above 0.9 are considered, as values below 0.9 only showed the sole vSOBI results based on initial investigations. This behavior indicates that fourth-order effects are highly present in the dataset. Figure 10a and Figure 10b depict the different settings of b for the default lag setting, while panels Figure 10c and Figure 10d show different b for the custom lag setting, found in Section 4.3. Specifically, these plots show matrices of pairwise MD indices and clustering of the results based on correlations between the



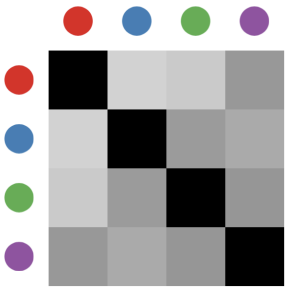
(a) Component similarity default lags



(b) MD indices default lags



(c) Component similarity custom lags



(d) MD indices custom lags

Figure 10: Results for different choices of gSOBI parameter settings (dots in a and c). Variations of b parameter are highlighted for the default lag set $\boldsymbol{k}_1 = \{1, \ldots, 12\}$ in a–b: $b = 0$, $b = 0.95$, $b = 0.99$ and $b = 1$. Variations of b parameter are highlighted for the custom lag set $\boldsymbol{k}_1 = \{1, \ldots, 6, 14, 15, 59, 60, 61\}$ in c–d: $b = 0$, $b = 0.97$, $b = 0.99$ and $b = 1$. Parameter settings of the remaining points are listed in Appendix D.

components.

By evaluating the results for the default lag setting (left panel Figure 10a), it is evident that a parameter of $b = 0.95$ is still very close to the sole vSOBI ($b = 0$) and a parameter of $b = 0.99$ is closer to the sole SOBI ($b = 1$). For the custom lag selection (Figure 10c) it is interesting to see that the SOBI solution $b = 1$ is far from all other computed results. Regarding the MD index, the $b = 0.97$ and $b = 0.99$ solutions are still close to the SOBI solution $b = 1$ (compare right panels in Figure 10c and Figure 10d). When the parameter b is changed, the solutions lie in the middle of the sole vSOBI solution and the SOBI solution for both lag settings in the component similarity plot. This behavior hints that these solutions combine features found by both methods. The solutions obtained by the custom lag setting appear to be more spread out in the component similarity plot. For this reason, we investigate the corresponding gSOBI solutions ($b = 0.97$ and $b = 0.99$) in more detail.

4.5. Results of gSOBI

In the previous section, two alternative gSOBI solutions with the parameters $\boldsymbol{k}_1 =$

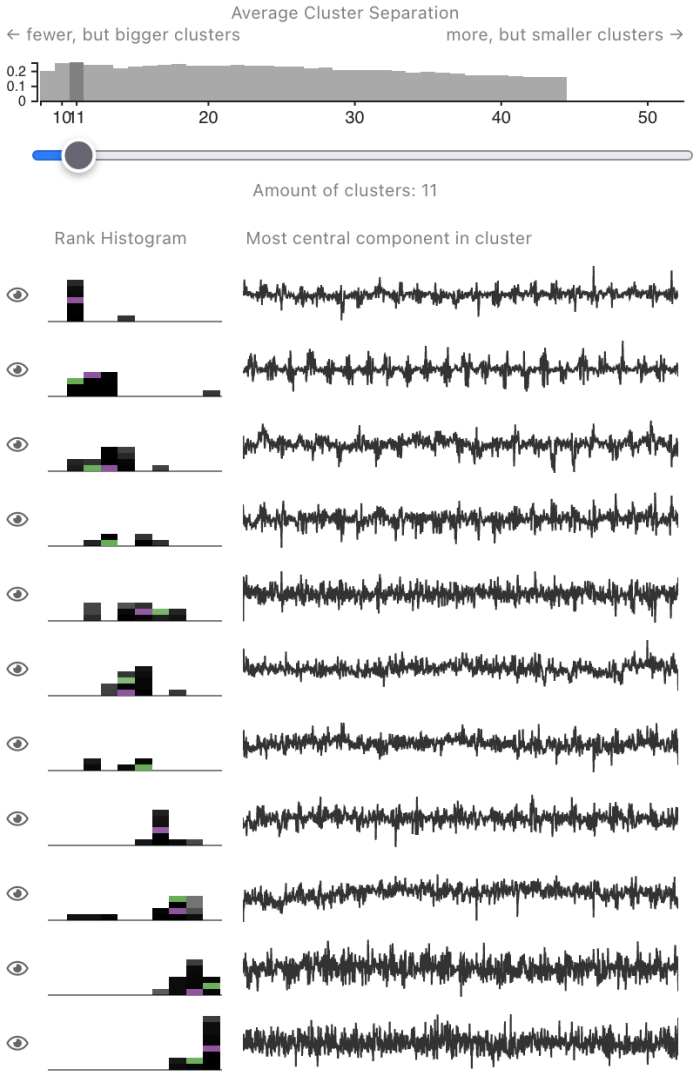


Figure 11: Clustering for different results. The colors depict gSOBI with weights $b = 0.97$ and $b = 0.99$ and lag sets $\boldsymbol{k}_1 = \{1, \ldots, 6, 14, 15, 59, 60, 61\}$, $\boldsymbol{k}_2 = \{1, 2, 3\}$ (same for both colors).

$\{1, \ldots, 6, 14, 15, 59, 60, 61\}$, $\boldsymbol{k}_2 = \{1, 2, 3\}$ and $b = 0.97$ and $b = 0.99$ have been found. These results have to be compared in more detail as the custom lag setting uses lags with the highest autocorrelation, and the two b values lie in between the two extremes SOBI and vSOBI. The latent components (of all computed results) are clustered in Figure 11 to investigate the differences between the two gSOBI results. The number of optimal clusters equals eleven, as that is where the clustering quality is best (bar chart on top of the slider). From the rank histograms in Figure 11, it can be observed that the components listed as first and eighth are only found in solution $b = 0.97$, while only $b = 0.99$ captures components listed as fourth and seventh, as the respective histograms contain either a green or purple box, but not both. All components and their correlation are depicted in Figure 12 ordered by kurtosis value, which shows that the first and second $b = 0.99$ components equal the second and third of the $b = 0.97$ results. Especially the first green and second purple components show again time-

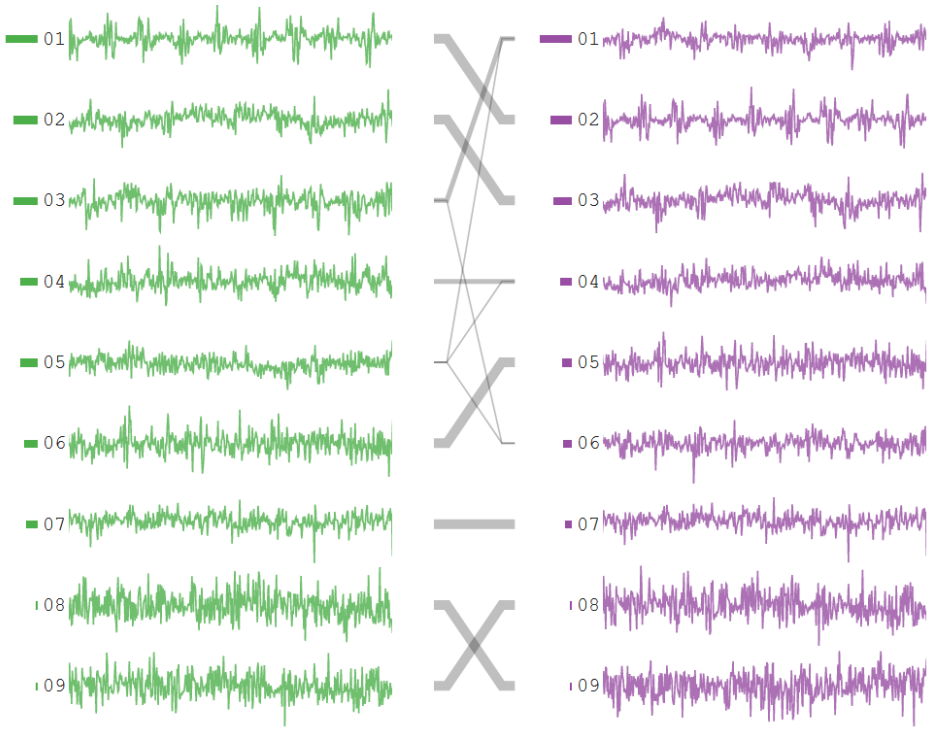
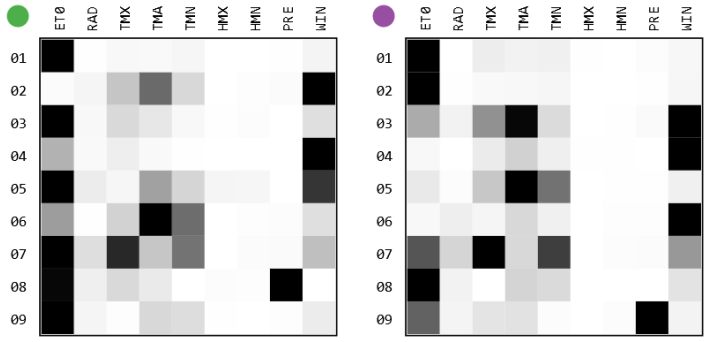


Figure 12: Latent time series and correlation lines for gSOBI with weights $b = 0.97$ and $b = 0.99$ and lag sets $\boldsymbol{k}_1 = \{1, \ldots, 6, 14, 15, 59, 60, 61\}$, $\boldsymbol{k}_2 = \{1, 2, 3\}$ (same for both colors). The order is according to kurtosis values.

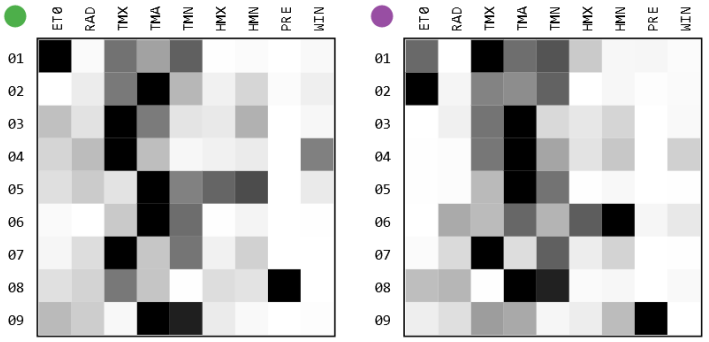
varying oscillations, which are high during the summer months and low during winter. By inspecting the corresponding loadings matrix (Figure 13), we can see that this component is similarly formed by evapotranspiration (ET0) and temperature-related variables. Similarly, the second green and third purple components show similarities in their patterns and loadings compared to the fourth latent component of the sole gSOBI results. The first purple and third green components show a medium correlation. The purple component again reveals time-varying oscillation with the highest volatility in the winter (in contrast to purple component two).

5. Discussion and Conclusions

TBSS can serve as a tool to gain insights into the data at hand as it combines a straightforward interpretation scheme (loadings and scores) and delivers latent components that are serially uncorrelated and maximize certain criteria important to time series analysis: SOBI maximizes autocorrelation, vSOBI maximizes fourth-order serial dependence and gSOBI is a linear combination of these two quantities. While the theoretical implications of TBSS are well-studied, the challenge lies in finding suitable tuning parameter settings for the dataset at hand. SOBI and vSOBI need one set of lags; therefore, gSOBI needs two lag sets and the vSOBI/SOBI ratio parameter b. On the order of 2^T, gSOBI tuning parameter settings are possible for a time series of length T, thus requiring systematic and efficient ways to traverse this huge parameter space and related latent components. We described VA as a promising remedy to these



(a) Loadings matrices for the raw data



(b) Loadings matrices for the scaled data

Figure 13: Loadings matrices for gSOBI with $b = 0.97$ and $b = 0.99$ and lag sets $\boldsymbol{k}_1 = \{1, \ldots, 6, 14, 15, 59, 60, 61\}$, $\boldsymbol{k}_2 = \{1, 2, 3\}$ (same for both colors).

problems in Section 1 and Section 3.

In the presented case study, numerous parameter settings were investigated in the first step. Based on those, three lag sets for SOBI and for vSOBI and three different b parameters with all lag set combinations for gSOBI were investigated in more detail. In total, roughly thirty solutions were investigated in under an hour. The time effort necessary to carry out the same analysis steps in R/RStudio would exceed that of TBSSvis by far. That becomes especially clear when considering that TBSSvis is interactive, allowing quick comparisons between different results. Moreover, the quality of the results is immediately visible as all important quantities (e.g., scores, loadings matrix, correlation, and MD to other results) are shown in custom-made visualizations to support the task at hand. Qualitatively, TBSSvis offers an interactive surface to investigate multivariate time series through the ACF function, lagged fourth moments, time series plots, and lagged scatter plots. On the other hand, plots of different quantities that are specific to the TBSS framework are interactively visualized. The ability to switch easily between these two surfaces mimics the general BSS analysis framework as the time series specific quantities influence the choice of the parameters, and consequently, the results, leading again to refinements of the parameters. Interactively carrying out this analysis cycle further enhances the ability to scan the TBSS methods' parameter space conveniently.

Finally, BSS methods were also extended to other kinds of data. For example, those

observed at different geographical positions, i.e., spatial BSS (Nordhausen et al. 2015; Bachoc et al. 2020). Following the successful application of VA principles to TBSS, it stands to reason that similar benefits may be gained for other BSS methods, which we could already show for spatial BSS tuning parameter selection (Piccolotto et al. 2022b). Thus, a logical next research step is to develop BSS theory and VA tools for multivariate data exhibiting temporal *and* spatial dependencies.

Acknowledgments

We are grateful for the valuable comments made by the referees and the editor which helped to improve the paper considerably. This work was supported by the Austrian Science Fund (FWF) under grant P31881-N32, Vienna Science and Technology Fund (WWTF) under grant [10.47379/ICT19047] and COST Action HiTEc (CA21163).

References

Aigner, W., Miksch, S., Schumann, H., and Tominski, C. (2023). *Visualization of Time-Oriented Data*. Springer, London, UK, DOI: 10.1007/978-1-4471-7527-8.

Artoni, F., Gemignani, A., Sebastiani, L., Bedini, R., Landi, A., and Menicucci, D. (2012). ErpICASSO: A Tool for Reliability Estimates of Independent Components in EEG Event-Related Analysis. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 368–371. DOI: 10.1109/EMBC.2012.6345945.

Artoni, F., Menicucci, D., Delorme, A., Makeig, S., and Micera, S. (2014). RELICA: A Method for Estimating the Reliability of Independent Components. *NeuroImage*, 103:391–400, DOI: 10.1016/j.neuroimage.2014.09.010.

Bachoc, F., Genton, M. G., Nordhausen, K., Ruiz-Gazen, A., and Virta, J. (2020). Spatial Blind Source Separation. *Biometrika*, 107:627–646, DOI: 10.1093/biomet/asz079.

Belouchrani, A., Abed Meraim, K., Cardoso, J.-F., and Moulines, E. (1997). A Blind Source Separation Technique Based on Second Order Statistics. *IEEE Transactions on Signal Processing*, 45:434–444, DOI: 10.1109/78.554307.

Bostock, M., Ogievetsky, V., and Heer, J. (2011). D3: Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, ISSN: 1077-2626, DOI: 10.1109/TVCG.2011.185.

Brown, C. (2022). ***hash***: *Full Featured Implementation of Hash Tables/Associative Arrays/Dictionaries*, https://CRAN.R-project.org/package=hash. R package version 2.2.6.2.

Bábek, O., Fačevicová, K., Žídek, M., Sedláček, J., Muehlmann, C., Nordhausen, K., and Hron, K. (2022). X-Ray Fluorescence Scanning of Soft and Wet-Sediment Cores

in Terrestrial Environments; a Robust Blind Source Separation Approach. *Journal of Geochemical Exploration*, 243:107106, DOI: 10.1016/j.gexplo.2022.107106.

Carnell, R. (2022). ***lhs**: Latin Hypercube Samples*, https://CRAN.R-project.org/package=lhs. R package version 1.1.5.

Clarkson, D. (1988). A Least Squares Version of Algorithm AS 211: The F-G Diagonalization Algorithm. *Applied Statistics*, 37:317–321, DOI: 10.2307/2347359.

de Lathauwer, L., de Moor, B., and Vandewalle, J. (2000). Fetal electrocardiogram extraction by blind source subspace separation. *IEEE Transactions on Biomedical Engineering*, 47(5):567–572, DOI: 10.1109/10.841326.

Drost, H. (2018). **Philentropy**: Information theory and distance quantification with R. *Journal of Open Source Software*, 3(26):765, DOI: 10.21105/joss.00765. R package version 0.7.0.

Eddelbuettel, D., Lucas, A., Tuszynski, J., Bengtsson, H., Urbanek, S., Frasca, M., Lewis, B., Stokely, M., Muehleisen, H., Murdoch, D., Hester, J., Wu, W., Kou, Q., Onkelinx, T., Lang, M., Simko, V., Hornik, K., Neal, R., Bell, K., de Queljoe, M., Suruceanu, I., Denney, B., Schumacher, D., , and Chang., W. (2020). ***digest**: Create Compact Hash Digests of R Objects*, https://CRAN.R-project.org/package=digest. R package version 0.6.27.

Firat, U., Engin, S. N., Saraclar, M., and Ertuzun, A. B. (2010). Wind Speed Forecasting Based on Second Order Blind Identification and Autoregressive Model. In *Ninth International Conference on Machine Learning and Applications*, pages 686–691. IEEE, DOI: 10.1109/ICMLA.2010.106.

Grolemund, G. and Wickham, H. (2011). Dates and times made easy with **lubridate**. *Journal of Statistical Software*, 40(3):1–25, DOI: 10.18637/jss.v040.i03. R package version 1.8.0.

Hargreaves, G. and Samani, Z. (1985). Reference crop evapotranspiration from temperature. *Applied Engineering in Agricolture*, (1):96–99, DOI: 10.13031/2013.26773.

Hargreaves, G. H. and Samani, Z. A. (1982). Estimating Potential Evapotranspiration. *Journal of Irrigation and Drainage Division*, 108(3):223–230, DOI: 10.1061/JRCEA4.0001390.

Henry, L. and Wickham, H. (2020). ***purrr**: Functional Programming Tools*, https://CRAN.R-project.org/package=purrr. R package version 0.3.4.

Illner, K., Miettinen, J., Fuchs, C., Taskinen, S., Nordhausen, K., Oja, H., and Theis, F. J. (2015). Model Selection Using Limiting Distributions of Second-Order Blind Source Separation Algorithms. *Signal Processing*, 113:95–103, DOI: 10.1016/j.sigpro.2015.01.017.

Ilmonen, P., Nordhausen, K., Oja, H., and Ollila, E. (2010). A New Performance Index for ICA: Properties, Computation and Asymptotic Analysis. In Vigneron, V., Zarzoso, V., Moreau, E., Gribonval, R., and Vincent, E., editors,

Latent Variable Analysis and Signal Separation, Lecture Notes in Computer Science, pages 229–236, Berlin, Heidelberg. Springer, ISBN: 978-3-642-15995-4, DOI: 10.1007/978-3-642-15995-4_29.

Inselberg, A. and Dimsdale, B. (1990). Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry. In *Proceedings of the First IEEE Conference on Visualization: Visualization '90*, pages 361–378. DOI: 10.1109/VISUAL.1990.146402.

Izrailev, S. (2022). ***tictoc**: Functions for Timing R Scripts, as Well as Implementations of "Stack" and "StackList" Structures*, https://CRAN.R-project.org/package=tictoc. R package version 1.1.

Joyce, C. A., Gorodnitsky, I. F., and Kutas, M. (2004). Automatic Removal of Eye Movement and Blink Artifacts from EEG Data Using Blind Component Separation. *Psychophysiology*, 41(2):313–325, DOI: 10.1111/j.1469-8986.2003.00141.x.

Keim, D., Kohlhammer, J., Ellis, G., and Mansmann, F. (2010). *Mastering the Information Age Solving Problems with Visual Analytics*. Eurographics Association, DOI: 10.2312/14803.

Keim, D. A., Mansmann, F., Schneidewind, J., Thomas, J., and Ziegler, H. (2008). Visual Analytics: Scope and Challenges. In *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*. Springer, Berlin, Heidelberg, DOI: 10.1007/978-3-540-71080-6_6.

Kendal, J.K. Stuart, A. (1983). *The Advanced Theory of Statistics*. Griffin, London.

Komsta, L. and Novomestky, F. (2022). ***moments**: Moments, Cumulants, Skewness, Kurtosis and Related Tests*, https://CRAN.R-project.org/package=moments. R package version 0.14.1.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., and Hornik, K. (2019). ***cluster**: Cluster Analysis Basics and Extensions*, https://CRAN.R-project.org/package=cluster. R package version 2.1.0.

Matilainen, M., Croux, C., Miettinen, J., Nordhausen, K., Oja, H., Taskinen, S., and Virta, J. (2020). ***tsBSS**: Blind Source Separation and Supervised Dimension Reduction for Time Series*, https://CRAN.R-project.org/package=tsBSS. R package version 0.5.6.

Matilainen, M., Miettinen, J., Nordhausen, K., Oja, H., and Taskinen, S. (2017). On Independent Component Analysis with Stochastic Volatility Models. *Austrian Journal of Statistics*, 46:57–66, DOI: 10.17713/ajs.v46i3-4.671.

Matilainen, M., Nordhausen, K., and Oja, H. (2015). New Independent Component Analysis Tools for Time Series. *Statistics & Probability Letters*, 105:80–87, DOI: 10.1016/j.spl.2015.04.033.

Matilainen, M., Nordhausen, K., and Virta, J. (2018). On the Number of Signals in Multivariate Time Series. In Deville, Y., Gannot, S., Mason, R., Plumbley, M., and Ward, D., editors, *Latent Variable Analysis and Signal Separation*, pages 248–258. Springer, DOI: 10.1007/978-3-319-93764-9_24.

MDN Contributors (2022). *JavaScript Reference*. Mozilla Corporation, Mountain View, CA, USA, https://developer.mozilla.org/en-US/docs/Web/JavaScript.

Miettinen, J., Illner, K., Nordhausen, K., Oja, H., Taskinen, S., and Theis, F. (2016). Separation of Uncorrelated Stationary Time Series Using Autocovariance Matrices. *Journal of Time Series Analysis*, 37:337–354, DOI: 10.1111/jtsa.12159.

Miettinen, J., Matilainen, M., Nordhausen, K., and Taskinen, S. (2020). Extracting Conditionally Heteroskedastic Components using Independent Component Analysis. *Journal of Time Series Analysis*, 41:293–311, DOI: 10.1111/jtsa.12505.

Miettinen, J., Nordhausen, K., Oja, H., and Taskinen, S. (2012). Statistical Properties of a Blind Source Separation Estimator for Stationary Time Series. *Statistics & Probability Letters*, 82(11):1865–1873, DOI: 10.1016/j.spl.2012.06.025.

Miettinen, J., Nordhausen, K., Oja, H., and Taskinen, S. (2014). Deflation-based Separation of Uncorrelated Stationary Time Series. *Journal of Multivariate Analysis*, 123:214–227, DOI: 10.1016/j.jmva.2013.09.009.

Miettinen, J., Taskinen, S., Nordhausen, K., and Oja, H. (2015). Fourth Moments and Independent Component Analysis. *Statistical Science*, 30(3):372–390, DOI: 10.1214/15-STS520.

Muehlmann, C., Nordhausen, K., and Virta, J. (2021). ***SpatialBSS**: Blind Source Separation for Multivariate Spatial Data*, https://CRAN.R-project.org/package=SpatialBSS. R package version 0.11-0.

Muehlmann, C., Piccolotto, N., De Iaco, S., and Nordhausen, K. (2022). ***SpaceTimeBSS**: Blind Source Separation for Multivariate Spatio-Temporal Data*, https://CRAN.R-project.org/package=SpaceTimeBSS. R package version 0.2-0.

Müller, K. and Wickham, H. (2020). ***tibble**: Simple Data Frames*, https://CRAN.R-project.org/package=tibble. R package version 3.0.4.

Nordhausen, K., Matilainen, M., Miettinen, J., Virta, J., and Taskinen, S. (2021). Dimension Reduction for Time Series in a Blind Source Separation Context Using R. *Journal of Statistical Software*, 98:1–30, DOI: 10.18637/jss.v098.i15.

Nordhausen, K. and Oja, H. (2018). Independent Component Analysis: A Statistical Perspective. *WIREs: Computational Statistics*, 10:e1440, DOI: 10.1002/wics.1440.

Nordhausen, K., Oja, H., Filzmoser, P., and Reimann, C. (2015). Blind Source Separation for Spatial Compositional Data. *Mathematical Geosciences*, 47(7):753–770, DOI: 10.1007/s11004-014-9559-5.

Nordhausen, K., Ollila, E., and Oja, H. (2011). On the Performance Indices of ICA and Blind Source Separation. In *2011 IEEE 12th International Workshop on Signal Processing Advances in Wireless Communications*, pages 486–490. DOI: 10.1109/SPAWC.2011.5990458.

Nordhausen, K. and Ruiz-Gazen, A. (2022). On the Usage of Joint Diagonalization in Multivariate Statistics. *Journal of Multivariate Analysis*, 188:104844, DOI: 10.1016/j.jmva.2021.104844.

Nordhausen, K., Taskinen, S., and Virta, J. (2022). Signal dimension estimation in BSS models with serial dependence. In *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pages 1–7. DOI: 10.1109/ICECCME55909.2022.9988152.

Pan, Y., Matilainen, M., Taskinen, S., and Nordhausen, K. (2022). A Review of Second-Order Blind Identification Methods. *WIREs Computational Statistics*, 14:e1550, DOI: 10.1002/wics.1550.

Piccolotto, N. (2022). ***TBSSvis***, DOI: 10.5281/zenodo.7315612, https://github.com/npiccolotto/tbss-vis. version v1.0.1.

Piccolotto, N., Bögl, M., Gschwandtner, T. Muehlmann, C., Nordhausen, K., Filzmoser, P., and Miksch, S. (2022a). **TBSSvis**: Visual Analytics for Temporal Blind Source Separation. *Visual Informatics*, 6(4):51–66, DOI: 10.1016/j.visinf.2022.10.002.

Piccolotto, N., Bögl, M., Muehlmann, C., Nordhausen, K., Filzmoser, P., and Miksch, S. (2022b). Visual parameter selection for spatial blind source separation. *Computer Graphics Forum*, 41(3):157–168, DOI: 10.1111/cgf.14530.

Popescu, T. D. and Manolescu, M. (2007). Blind Source Separation of Traffic-Induced Vibrations in Building Monitoring. In *IEEE International Conference on Control and Automation*, pages 2101–2106. IEEE, DOI: 10.1109/ICCA.2007.4376731.

R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, https://www.R-project.org/.

Roberts, J. C. (2007). State of the Art: Coordinated & Multiple Views in Exploratory Visualization. In *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2007)*, pages 61–71. DOI: 10.1109/CMV.2007.20.

RStudio Team (2022). ***RStudio**: Integrated Development Environment for R*. RStudio, PBC, Boston, MA, http://www.rstudio.com/.

Schloerke, B. and Allen, J. (2022). ***plumber**: An API Generator for R*, https://www.rplumber.io,https://github.com/rstudio/plumber.

Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages*, pages 336–343, Boulder, Colorado, USA. IEEE, DOI: 10.1109/VL.1996.545307.

Tang, A. (2010). Applications of Second Order Blind Identification to High-Density EEG-Based Brain Imaging: A Review. In Zhang, L., Lu, B.-L., and Kwok, J., editors, *Advances in Neural Networks - ISNN 2010: 7th International Symposium on Neural Networks, Part II*, pages 368–377. Springer, DOI: 10.1007/978-3-642-13318-3_46.

Tang, A. C., Liu, J.-Y., and Sutherland, M. T. (2005a). Recovery of Correlated Neuronal Sources From EEG: The Good and Bad Ways of Using SOBI. *NeuroImage*, 28:507–519, DOI: 10.1016/j.neuroimage.2005.06.062.

Tang, A. C., Pearlmutter, B. A., Zibulevsky, M., and Carter, S. A. (2000). Blind Source Separation of Multichannel Neuromagnetic Responses. *Neurocomputing*, 32:1115–1120, DOI: 10.1016/S0925-2312(00)00286-1.

Tang, A. C., Sutherland, M. T., and McKinney, C. J. (2005b). Validation of SOBI Components from High-Density EEG. *Neuroimage*, 25(2):539–553, DOI: 10.1016/j.neuroimage.2004.11.027.

Taskinen, S., Miettinen, J., and Nordhausen, K. (2016). A More Efficient Second Order Blind Identification Method for Separation of Uncorrelated Stationary Time Series. *Statistics & Probability Letters*, 116:21–26, DOI: 10.1016/j.spl.2016.04.007.

Tong, L., Soon, V., Huang, Y., and Liu, R. (1990). AMUSE: A New Blind Identification Algorithm. In *Proceedings of IEEE International Symposium on Circuits and Systems*, pages 1784–1787. IEEE, DOI: 10.1109/ISCAS.1990.111981.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition.

Virta, J. and Nordhausen, K. (2019). Estimating the Number of Signals Using Principal Component Analysis. *Stat*, 8:e231, DOI: 10.1002/sta4.231.

Virta, J. and Nordhausen, K. (2021). Determining the Signal Dimension in Second Order Source Separation. *Statistica Sinica*, 31:135–156, DOI: 10.5705/ss.202018.0347.

Vlachos, M., Yu, P., and Castelli, V. (2005). On Periodicity Detection and Structural Periodic Similarity. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 449–460. Society for Industrial and Applied Mathematics, DOI: 10.1137/1.9781611972757.40.

Wagstaff, K., Cardie, C., Rogers, S., and Schroedl, S. (2001). Constrained K-means Clustering with Background Knowledge. In *Proceedings of the International Conference on Machine Learning*, pages 577–584, San Francisco, CA, USA. Morgan Kaufmann, DOI: 10.5555/645530.655669.

Wax, M. and Kailath, T. (1985). Detection of Signals by Information Theoretic Criteria. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33:387–392, DOI: 10.1109/TASSP.1985.1164557.

Wickham, H. (2007). Reshaping Data with the **reshape** Package. *Journal of Statistical Software*, 21(12):1–20, DOI: 10.18637/jss.v021.i12. R package version 1.4.4.

Wickham, H. (2021). ***pryr****: Tools for Computing on the Language*, https://CRAN.R-project.org/package=pryr. R package version 0.1.5.

Wickham, H., François, R., Henry, L., and Müller, K. (2020). ***dplyr**: A Grammar of Data Manipulation*, https://CRAN.R-project.org/package=dplyr. R package version 1.0.2.

Zeileis, A. and Grothendieck, G. (2005). **zoo**: S3 Infrastructure for Regular and Irregular Time Series. *Journal of Statistical Software*, 14(6):1–27, DOI: 10.18637/jss.v014.i06. R package version 1.8-8.

Zhao, L., Krishnaiah, P., and Bai, Z. (1986). On Detection of the Number of Signals in Presence of White Noise. *Journal of Multivariate Analysis*, 20:1–25, DOI: 10.1016/0047-259X(86)90017-5.

A. Some more details on TBSS methods used

As mentioned in Section 2.2, gSOBI has as special cases AMUSE, SOBI and vSOBI. In the following, we give some more details about these methods.

A.1. AMUSE and SOBI

One of the first methods suggested to exploit serial dependence in a BSS framework was AMUSE (Tong et al. 1990). Let $\mathrm{ACov}_\tau(\boldsymbol{x}_t)$ denote the autocovariance matrix of $\boldsymbol{x}_t$ at lag τ which reduces for $\tau = 0$ to the covariance matrix, i.e., $\mathrm{ACov}_0(\boldsymbol{x}_t) = \mathrm{Cov}(\boldsymbol{x}_t)$. For a lag $\tau \geq 1$ the AMUSE unmixing matrix $\boldsymbol{W}_A$ is obtained as the matrix which simultaneously diagonalizes $\mathrm{Cov}(\boldsymbol{x}_t)$ and $\mathrm{ACov}_\tau(\boldsymbol{x}_t)$ in the following way

$$\boldsymbol{W}_A \,\mathrm{Cov}(\boldsymbol{x}_t)\boldsymbol{W}_A^\top = \boldsymbol{I}_p \quad \text{and} \quad \boldsymbol{W}_A \,\mathrm{ACov}_\tau(\boldsymbol{x}_t)\boldsymbol{W}_A^\top = \boldsymbol{D},$$

where $\boldsymbol{D}$ is a diagonal matrix with decreasing diagonal elements $d_1, \ldots, d_p$. AMUSE can be obtained via a generalized eigenvalue decomposition and it is well defined if all diagonal elements in $\boldsymbol{D}$ are distinct, which translates to the assumptions:

IC3: $\mathrm{ACov}_\tau(\boldsymbol{z}_t) = \boldsymbol{D}_\tau$ for all τ, where $\boldsymbol{D}_\tau$ denotes a diagonal matrix whose diagonal elements depend on τ.

IC4: For at least one τ the diagonal elements of $\boldsymbol{D}_\tau$ are distinct.

The statistical properties of AMUSE are discussed in Miettinen et al. (2012) and it is well known that the performance of AMUSE depends heavily on the choice of τ which should be chosen so that the gaps between the diagonal elements of $\boldsymbol{D}_\tau$ are maximized. Finding a good lag τ is quite challenging and AMUSE is therefore used often simply with $\tau = 1$.

However, in practical applications, the preferred approach is not to have to choose one lag but rather a set of distinct lags $\boldsymbol{k} = \{\tau_1, \ldots, \tau_K\}$ and then jointly diagonalizing the corresponding K autocovariance matrices. This approach is known as SOBI (Belouchrani et al. 1997) and the SOBI unmixing matrix $\boldsymbol{W}_S$, is defined as the maximizer of

$$\sum_{k=1}^{K}\sum_{i=1}^{p}(\boldsymbol{w}_i^\top \,\mathrm{ACov}_{\tau_k}(\boldsymbol{x}_t)\boldsymbol{w}_i)^2,$$

under the constraint that $\boldsymbol{W}\,\mathrm{Cov}(\boldsymbol{x}_t)\boldsymbol{W}^\top = \boldsymbol{I}_p$, where $\boldsymbol{w}_i^\top$ denotes the ith row of $\boldsymbol{W}$, $i = 1, \ldots, p$. Usually the order of the components is fixed so that the pseudo-eigenvalues $\sum_{k=1}^{K}\sum_{i=1}^{p}(\boldsymbol{w}_i^\top \,\mathrm{ACov}_{\tau_k}(\boldsymbol{x}_t)\boldsymbol{w}_i)^2$ are in descending order.

For the estimation purpose, there are many algorithms for this joint diagonalization problem as discussed, for example, in Illner et al. (2015). Using an algorithm is necessary as for finite samples the $K + 1$ matrices involved will usually not commute due to measurement error and the problem is thus formulated as a minimization problem. Two matrices as used in AMUSE can always be simultaneously diagonalized. The algorithm used for the joint diagonalization has also influence on the statistical properties of the estimator as discussed in Miettinen et al. (2014, 2016), and in the paper, we will

always use in the case of joint diagonalization the algorithm based on Jacobi rotations (Clarkson 1988) which seems to be the most common choice. Independent from that, SOBI is well-defined under (IC1-IC3) and the additional assumption

IC4: The diagonal elements of $\sum_i \boldsymbol{D}_i$ are strictly decreasing.

The assumption (IC4) means that the spectrum of the independent times series differs and the lag set used in SOBI should be chosen such that for each component there is at least one lag for which the autocorrelation differs from all other components' autocorrelation. SOBI with one lag reduces to AMUSE but when using more lags it is much more general. The previous condition is often evaluated by the simplified approximation that the sum of the pseudo-eigenvalues $\sum_{k=1}^{K}\sum_{i=1}^{p}(\boldsymbol{w}_i^\top \operatorname{ACov}_{\tau_k}(\boldsymbol{x}_t)\boldsymbol{w}_i)^2$ should be distinct. The general motivation for SOBI vs AMUSE is that even if AMUSE has a good lag chosen adding worse lags in SOBI will make the unmixing only minimal worse, but the chance of having good lags chosen in SOBI is much more likely than relying on one specific lag of AMUSE. This is also supported by the asymptotic findings in Miettinen et al. (2014, 2016).

Tang et al. (2005a) argue that the lag selection in SOBI is still crucial and show the considerable effects in some neuro-science applications when considering different lag sets. But there are no guidelines for practitioners and a common default choice is to use the $K = 12$ first lags. These are for example shown to be bad in the context of the examples of Tang et al. (2005a). Taskinen et al. (2016) suggest choosing a few different lag sets and then estimate for each of the sets, the asymptotic covariance matrix of the estimated unmixing matrix, and choose as best set the one where the estimated covariance matrix has the smallest volume. The limitation of this approach is that the number of lag sets should be small as estimation of the covariance matrix is computationally expensive and requires much stricter assumptions as specified in (IC1)–(IC4) and also large sample sizes.

Note that the model specified by (IC1)–(IC3) is often called the second-order source separation (SOS) model as it can be separated using second-order information alone.

A.2. vSOBI

Loosely, AMUSE and SOBI work best if the latent components have some kind of autoregressive moving average (ARMA) structure but are not very successful for economic and financial times series. The reason is that by using only second-order information, components with stochastic volatility features cannot be detected. To overcome this issue, Matilainen et al. (2017) reformulated SOBI by introducing a non-linearity function G into the optimization problem above and yield vSOBI, with the corresponding unmixing matrix $\boldsymbol{W}_{vS}$, as the maximizer of

$$\sum_{k=1}^{K}\sum_{i=1}^{p}(E(G(\boldsymbol{w}_i^\top \boldsymbol{x}_t)G(\boldsymbol{w}_i^\top \boldsymbol{x}_{t+\tau_k}) - (E(G(\boldsymbol{w}_i^\top \boldsymbol{x}_t)E(G(\boldsymbol{w}_i^\top \boldsymbol{x}_{t+\tau_k})))^2,$$

where $\boldsymbol{x}_t$ is centered and again the constraint $\boldsymbol{W}\operatorname{Cov}(\boldsymbol{x}_t)\boldsymbol{W}^\top = \boldsymbol{I}_p$ applies. The unmixing matrix $\boldsymbol{W}_{vS}$ can be computed for finite data using a fixed-point algorithm, and G can be any twice continuously differentiable function and common choices are

$G(x) = x^2$ or $G(x) = \log(\cosh(x))$. vSOBI works under the framework (IC1) and (IC2) and when it fails depends on the lag set chosen and on the function G. In general, it focuses on higher order information and was designed to detect stochastic volatility features and is not very efficient in a pure SOS model.

However, as most time series exhibit second order and higher order dependence, vSOBI was not much investigated and the idea of gSOBI is to combine information exploited by AMUSE/SOBI and by vSOBI.

B. R packages used on backend server

Table 2 summarizes the R which are used to derive the results described in Section 4.

Table 2: Summary of the used R packages on the backend server.

Name	Version	Reference
cluster	2.1.0	(Maechler et al. 2019)
digest	0.6.27	(Eddelbuettel et al. 2020)
dplyr	1.0.2	(Wickham et al. 2020)
hash	2.2.6.2	(Brown 2022)
lhs	1.1.5	(Carnell 2022)
lubridate	1.8.0	(Grolemund and Wickham 2011)
MASS	7.3-53	(Venables and Ripley 2002)
moments	0.14.1	(Komsta and Novomestky 2022)
Philentropy	0.7.0	(Drost 2018)
pryr	0.1.5	(Wickham 2021)
purrr	0.3.4	(Henry and Wickham 2020)
reshape2	1.4.4	(Wickham 2007)
SpatialBSS	0.11-0	(Muehlmann et al. 2021)
tibble	3.0.4	(Müller and Wickham 2020)
tictoc	1.1	(Izrailev 2022)
tsBSS	0.5.6	(Matilainen et al. 2020)
zoo	1.8-8	(Zeileis and Grothendieck 2005)

C. Data Preparation

The following R code is used to derive a CSV file with the considered dataset in Section 4 from the package **SpaceTimeBSS** version 0.2-0 (Muehlmann et al. 2022). The time series of the dataset are depicted in Figure 6.

```
library("SpaceTimeBSS")
data("meteo_veneto")

dat <- meteo_veneto[meteo_veneto$sp.ID == 3, ]
dat$x <- NULL
```

```
dat$y <- NULL
dat$sp.ID <- NULL
dat$timeIndex <- as.Date("2000-01-01") + (dat$timeIndex - 1) * 7
names(dat) <- c("date", "ETO", "RAD", "TMX", "TMA",
                "TMN", "HMX", "HMN", "PRE", "WIN")

write.csv(dat, "meteo_veneto_ts.csv",  row.names = FALSE)
```

D. Parameter Settings Used for Figures

Table 3: All parameter settings used for investigations in Section 4.

b	Lag set $\boldsymbol{k}_1$	Lag set $\boldsymbol{k}_2$
0	—	1, 2, 3
0.95	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	1, 2, 3
0.99	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	1, 2, 3
1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	—
0.9	1, 2, 3, 4, 5, 6, 14, 15, 59, 60, 61	1, 2, 3
0.95	1, 2, 3, 4, 5, 6, 14, 15, 59, 60, 61	1, 2, 3
0.97	1, 2, 3, 4, 5, 6, 14, 15, 59, 60, 61	1, 2, 3
0.99	1, 2, 3, 4, 5, 6, 14, 15, 59, 60, 61	1, 2, 3
1	1, 2, 3, 4, 5, 6, 14, 15, 59, 60, 61	—
0.9	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 120, 140, 160, 180, 200, 220, 240, 260, 280, 300	1, 2, 3
0.99	1, 2, 3, 4, 12, 36, 52, 104, 208	1
0.49	52, 151, 190, 196, 249, 434, 436, 486, 532, 534, 613, 621, 634, 650	11, 107, 109, 166, 250, 253, 256, 327, 337, 362, 367, 388, 394, 435, 461, 494, 502, 867
0.67	141, 455, 872	374, 402, 420, 480, 521, 571, 574, 619, 791, 824

Affiliation:

Claudia Cappello
Dipartimento di Scienze dell'Economia
University of Salento
Via per Monteroni (Complesso Ecotekne)
73100 Lecce, Italy
E-mail: claudia.cappello@unisalento.it

Nikolaus Piccolotto, Markus Bögl, Silvia Miksch
Centre for Visual Analytics Science and Technology
TU Wien
Favoritenstraße 9-11
1040 Vienna, Austria
E-mail: nikolaus.piccolotto@tuwien.ac.at, markus.boegl@tuwien.ac.at, silvia.miksch@tuwien.ac.at

Christoph Muehlmann, Peter Filzmoser
Institute of Statistics & Mathematical Methods in Economics
TU Wien
Wiedner Hauptstrasse 8-10
1040 Vienna, Austria
E-mail: christoph.muehlmann@tuwien.ac.at, peter.filzmoser@tuwien.ac.at

Klaus Nordhausen
Department of Mathematics and Statistics
University of Jyväskylä
P.O. Box 35 (MaD)
40014 University of Jyväskylä, Finland
E-mail: klaus.k.nordhausen@jyu.fi
URL: http://klausnordhausen.com/

Journal of Data Science, Statistics, and Visualisation https://jdssv.org/
published by the International Association for Statistical Computing
http://iasc-isi.org/

ISSN 2773-0689
April 2024, Volume IV, Issue III *Submitted:* 2022-12-01
doi:10.52933/jdssv.v4i3.82 *Accepted:* 2024-01-09