Master Thesis

Blind Source Separation for Compositional Time Series

Gregor Fischer
01403086

under the supervision of
Prof. Klaus Nordhausen

10.05.2020

Contents

1 Abstract

This thesis shows how blind source separation methods for time-series can be applied to compositional time series. In many applications data sets are of compositional nature, meaning that the relative values of the variables are of interest instead of the absolute ones. Blind source separation (BSS) is a popular modelling approach for multivariate time-series, since it aims to decompose them into latent sources on which univariate modelling is possible. Compositional time-series are per definition multivariate. Moreover, in their isometric-log-ratio-coordinate representation, on which the BSS models are built, they are multivariate if the number of compositions is greater than two. Therefore blind source separation is very useful for compositional time-series. Our methodology is illustrated on a real world data set: Absorption data from a stream in Lower Austria. In the study of dissolved organic matter, ratios of absorption coefficients have been used to indicate the quality of dissolved organic matter in various environments, yielding compositional time series data, on which our new method can be applied.

2 Introduction

2.1 Introduction

Blind source separation (BSS) is a well established method for analysing multivariate time series, since it has proven useful in modelling and interpretation, see for example Miettinen et al. [2017] or Miettinen et al. [2018]. In BSS a multivariate time series gets decomposed into latent sources, on which for instance univariate time series models can be fitted. BSS can also be used for dimension reduction of time series data. Real world application of blind source separation can be found among others in signal processing [Belouchrani and Amin, 1998]. When the time-series is of compositional nature, the statistical methodology of BSS and compositional data analysis have to be combined. Compositional data is data where instead of the absolute values of the variables, the relative ones are of interest. One of the first works in the field of compositional data analysis has been Aitchison [1982]. Since then this field has been gaining importance, see for example Pawlowsky-Glahn and Egozcue [2001], Pawlowsky-Glahn et al. [2015] and Filzmoser and Templ [2018], leading to an increased awareness that many multivariate data sets are of compositional nature.

In this thesis we will show how blind source separation can be applied on compositional data. We will then apply this concept on a concrete data set. In Section 3 we will describe statistical methodology on compositional data analysis. This will include two log-ratio transformations from the simplex to the Euclidean vector space. Subsection 3.3 will be about principal component analysis for compositional data. Section 4 will contain an introduction to blind source separation for time-series. This section will include three different blind source separation models: the second order source separation model, blind source separation for time-series with stochastic volatility and non-stationary source separation. Detailed descriptions of how sources can be estimated in these blind source separation models can be found in the subsections of the corresponding models. We will also present the novel blind source separation method by Nordhausen et al. [2020], that is a hybrid method that can be applied to all three blind source separation models. Therefore this method is suggested if it is unclear which blind source separation model to choose. In Section 5 we will explain how the methodology of compositional data and blind source separation for time series can be combined by fitting blind source separation models on the transformed compositions. Subsection 5.1 will further explain how principal component analysis can be added in order to be able to apply blind source separation to highly correlated compositional time series as well. In Section 6 there will be a detailed description on how we apply the new method on absorption data. At a stream near Petzenkirchen in Lower Austria absorption coefficients have been measured every ten minutes in a time period from January 2014 to December 2014. In the study of dissolved organic matter, ratios of absorption coefficients are useful to model the quality of dissolved organic matter, making the data set a compositional time series. This yields the necessity of our new methodology. Subsection 6.8 will contain a short outlook on how our results can be interpreted, though a more detailed description will be left to the experts on the field. In the appendix we will show R-code, that was implemented in order to apply the methods on the data set.

2.2 Notation

We want to dedicate a short chapter to the notation of this thesis, since a lot of different vectors and matrices will appear. In order to distinguish the different vectors and matrices we will use big letters in bold if an object is a matrix, small letters in bold if the object is a vector and small normal sized letters if the object is a scalar:

$\boldsymbol{V}$	matrix
$\boldsymbol{v}$	vector
v	scalar.

We use subindices in brackets to indicate elements of vectors and elements, rows or columns in matrices.

$V_{[i,j]}$	(i,j)-th element of the matrix $\boldsymbol{V}$
$\boldsymbol{V}_{[.,i]}$	i-th column of $\boldsymbol{V}$
$\boldsymbol{V}_{[i,.]}$	i-th row of $\boldsymbol{V}$
$v_{[i]}$	i-th element of the vector $\boldsymbol{v}$.

In time series we use the subindex t to indicate the time index in a time series:

$$\boldsymbol{x}_t, t \in T \quad \text{multivariate time-series}$$

In this thesis we deal with matrices that depend on some parameter τ. In these cases, we will write the matrix with a subindex τ: $\boldsymbol{S}_\tau$. Hence the notation with subindices in brackets for elements of a matrix in order not to overload the subindex notation.

In this thesis we also use transformations of vectors, for example a so called clr-transformation is applied on a compositional vector $\boldsymbol{x}$. We will use the abbreviation of the transformation as a subscript to distinguish the original vector from the transformed vector:

$$\begin{aligned} \boldsymbol{x} &\quad \text{original vector,} \\ \boldsymbol{x}^{clr} &\quad \text{clr-transformed vector.} \end{aligned} \tag{1}$$

3 Compositional data

3.1 Definition and explanations

In this section we will describe what compositional data is, and what the approach to such data from a statistics point of view looks like.

A compositional data vector is a vector, where the entries are proportions of some whole unit. Therefore it is assumed that the sum over the entries is constant.

Definition 1 (Compositional vector)
A vector is compositional, if its entries sum up to an arbitrary but fixed constant C:

$$\boldsymbol{x} \in \mathbb{R}^{p+1} \quad | \quad x_{[i]} > 0 \quad \sum_{i=1}^{p+1} x_{[i]} = C.$$

In Biology or Chemistry data sets are often of compositional nature, for instance when a concentration of an element in a liquid is measured. If the units were measured in $\frac{ml}{l}$ the constant C in Def. 1 would be 1000. Sometimes the constant is normed to 1. We will soon show, that the exact value of this constant does not matter at all.

This strict definition is useful, since it provides a sample space for compositional data, which leads to a methodology on how to apply statistical analysis on compositional data. However, for a definition of a data set to be compositional this definition is by no means to be taken literally.

Filzmoser and Templ [2018] give the following definition on compositional data analysis: *Compositional data analysis needs to be applied whenever the relative values of variables in a data set are of interest, instead of the absolute ones.* So whether a data set is compositional or not, depends on the purpose of the analysis instead of the actual data set.

The following example illustrates this point. Consider a data set with two variables: expenses and savings of a household. The sum of both variables represents the income of a household. The income might not be constant throughout all households. So the strict definition in Def. 1 does not hold. However, if one is interested in the proportion of the income a household saves, compositional data analysis needs to be applied. One could divide both variables by the income, in order to meet Def. 1. But it is important to emphasize that one does not "make" a data set compositional by doing so. From a compositional data analysis point of view an observation (1200, 800) would be considered the same as (2400, 1600). Therefore, a data vector $\boldsymbol{x}$ fulfilling Def. 1 with $C = 1$ ($\sum_{i=1}^{p+1} x_{[i]} = 1$), like (0.6, 0.4), is just a representation of all possible observation where the expanses are 1.5 times as big as the savings. Two data vectors, where one is a scaled version of the other one, are called compositionally equivalent.

3.2 Transformations of compositions, Aitchison geometry on the simplex

As discussed in the previous section, Def. 1 provides a sample space for compositional data. A sample space of a (p+1)-dimensional compositional vector is the (p+1)-dimensional unit simplex $\mathbb{S}^{p+1}$.

Definition 2 (Unit simplex)
The unit simplex $\mathbb{S}^{p+1}$ is defined as

$$\mathbb{S}^{p+1} := \{\boldsymbol{x} \in \mathbb{R}^{p+1} \quad | \quad x_{[i]} > 0 \quad \sum_{i=1}^{p+1} x_{[i]} = 1\}.$$

A lot of methodology in multivariate statistics like principal component analysis or cluster analysis are based on the Euclidean geometry of the $\mathbb{R}^{p+1}$. However, the standard Euclidean geometry is a poor choice for compositional data, since it violates many principles of compositional data analysis, such as scale invariance, perturbation invariance and subcompositional dominance (Aitchison et al. [2000]). Scale invariance, perturbation invariance and subcompositional dominance are principals of compositional data analysis that every statistical method on compositional data should fulfil. Scale invariance means, that the outcome should not depend on the unit of a compositional vector $\boldsymbol{x} = [x_{[1]}, .., x_{[p+1]}]^\top$. Therefore any compositional vector $\boldsymbol{x}$ should be treated the same as a compositionally equivalent vector $\lambda\boldsymbol{x}$, $\lambda > 0$. Perturbation invariance means, that the information contained in a compositional vector $\boldsymbol{x} = [x_{[1]}, \ldots, x_{[p+1]}]^\top$ should be the same as in a compositional vector where the compositions have been permuted $\boldsymbol{x}^{perm} = [x_{[\pi(1)]}, \ldots, x_{[\pi(p+1)]}]^\top$, π is a permutation of the integers 1 to $p + 1$. Subcompositional dominance is a request on the metric between compositional vectors. Any metric $\Delta_{p+1}(\boldsymbol{x}, \boldsymbol{y})$ between two compositional vectors with $p + 1$ parts should fulfil

$$\Delta_{p+1}(\boldsymbol{x}, \boldsymbol{y}) \geq \Delta_b(\boldsymbol{x}_{[1,\dots,b]}, \boldsymbol{y}_{[1,\dots,b]}), \tag{2}$$

where $\boldsymbol{x}_{[1,\dots b]}$ contains just the compositions 1 to b from the vector $\boldsymbol{x}$ ($b < p+1$). Subcompositional dominance can be seen as the equivalent to the triangle inequality of the Euclidean metric of the $\mathbb{R}^p$.

Since most of the time in statistics the methodology is based on the Euclidean metric of the $\mathbb{R}^{p+1}$, one way to deal with compositional data analysis is to transform the compositional data from $\mathbb{S}^{p+1}$ to $\mathbb{R}^{p+1}$ or to $\mathbb{R}^p$. This transformation should be consistent with the above defined principles of compositional data analysis. Since the ratios between the variables is what matters, any reasonable transformation should be a function of the ratios of the compositions. Aitchison [1982] proposed log-ratios. Due to the property of the natural logarithm $\ln\left(\frac{x_{[i]}}{x_{[j]}}\right) = -\ln\left(\frac{x_{[j]}}{x_{[i]}}\right)$, log-ratios are a good choice for building such a transformation. In contrary to just ratios, log-ratios implement a nice symmetric structure in the space of the transformed compositions. Consider a compositional vector with just two compositions $x_{[1]}$ and $x_{[2]}$. With the transformation $x = \ln(\frac{x_{[1]}}{x_{[2]}})$ the positive real numbers represent the space where $x_{[1]}$ is bigger, the negative real numbers represent the space where $x_{[2]}$ is bigger. The number 0 represents the point where the two compositions are exactly equal. For every $y \in \mathbb{R}$, $-y$ represents the point where the proportions are the other way around.

If the dimension of the compositional vector is bigger than two, it is not that trivial to decide which log-ratios to use. One popular transformation is the centered-log-ratio (clr) transformation.

Definition 3 (Clr-transformation)
The clr-transformation is a mapping from the simplex $\mathbb{S}^{p+1}$ to $\mathbb{R}^{p+1}$. The i-th coordinate of the clr-transformed vector is obtained as

$$\begin{array}{rccc} clr\text{:} & \mathbb{S}^{p+1} & \rightarrow & \mathbb{R}^{p+1} \\ & \boldsymbol{x} & \rightarrow & \boldsymbol{x}^{clr} \end{array}$$

$$\boldsymbol{x}^{clr}_{[i]} := \ln\left(\frac{x_{[i]}}{(\prod_{i=1}^{p+1} x_{[i]})^{\frac{1}{p+1}}}\right).$$

One advantage of this transformation is its interpretability. The i-th coordinate of the transformed vector is simply the log-ratio between the i-th composition and the geometric mean of all compositions. The i-th coordinate of the clr-transformed vector can also be seen as the average log-ratio of the i-th composition to all the other ones. Using the properties of the natural logarithm one can easily verify

$$\ln\left(\frac{x_{[i]}}{(\prod_{i=1}^{p+1} x_{[i]})^{\frac{1}{p+1}}}\right) = \frac{1}{p+1}\left(\ln\left(\frac{x_{[i]}}{x_{[1]}}\right) + \ln\left(\frac{x_{[i]}}{x_{[2]}}\right) + \ldots + \ln\left(\frac{x_{[i]}}{x_{[p+1]}}\right)\right). \tag{3}$$

One major drawback of this transformation is that the clr-vector is in $\mathbb{R}^{p+1}$, but describes something that is really of dimension p. Using the laws of the natural logarithm yields

$$\sum_{i=1}^{p+1} \boldsymbol{x}_{[i]}^{clr} = \sum_{i=1}^{p+1} \ln\left(\frac{x_{[i]}}{(\prod_{i=1}^{p+1} x_{[i]})^{\frac{1}{p+1}}}\right) = \ln\left(\frac{\prod_{i=1}^{p+1} x_{[i]}}{\left((\prod_{i=1}^{p+1} x_{[i]})^{\frac{1}{p+1}}\right)^{p+1}}\right) = \ln(1) = 0. \tag{4}$$

Hence a clr-transformed compositional vector lies on the p-dimensional hyperplane in $\mathbb{R}^{p+1}$:

Definition 4 (Clr hyperplane)
The clr-transformed compositional vectors lie on the following hyperplane in $\mathbb{R}^{p+1}$*:*

$$\mathbb{H}^{clr} = \{\boldsymbol{x} \in \mathbb{R}^{p+1} | \quad \sum_{i=1}^{p+1} x_{[i]} = 0\}.$$

Let $\boldsymbol{X}$ be a matrix, whose rows represent observations of compositional vectors. Furthermore let $\boldsymbol{X}^{clr}$ be the clr-transformed matrix, whose rows are the clr-transformed compositional vectors from $\boldsymbol{X}$. A consequence of the clr-transformed compositions laying on $\mathbb{H}^{clr}$ is that $\boldsymbol{X}^{clr}$ does not have full rank. That can be an issue in many multivariate statistics methods. For example in classical least squares regression one would need to invert the matrix $\boldsymbol{X}^{clr\top}\boldsymbol{X}^{clr}$, which is not invertible.

A transformation from the unit simplex $\mathbb{S}^{p+1}$ to the $\mathbb{R}^p$ is the isometric log-ratio transformation. The idea of this transformation is to build an orthonormal basis on the p-dimensional hyperplane $\mathbb{H}^{clr}$ from Def. 4 and express the clr-transformed compositions within that orthonormal basis. Since there are infinitely many possibilities for building such a basis, the isometric log-ratio transformations can be considered a class of transformations. One particular choice for such an orthonormal basis leads to Def. 5.

Definition 5 (Isometric log-ratio transformation)
The isometric log-ratio transformation is a mapping from $\mathbb{S}^{p+1}$ *to* $\mathbb{R}^p$*. The i-th coordinate of the ilr-transformed vector can be obtained as:*

$$\begin{aligned} ilr\colon \quad \mathbb{S}^{p+1} &\rightarrow \mathbb{R}^p \\ \boldsymbol{x} &\rightarrow \boldsymbol{x}^{ilr} \end{aligned}$$

$$\boldsymbol{x}_{[i]}^{ilr} := \left(\frac{p+1-i}{p-i+2}\right)^{\frac{1}{2}} \log\left(\frac{x_{[i]}}{(\prod_{j=i+1}^{p+1} x_{[j]})^{\frac{1}{p+1-i}}}\right).$$

Details on the derivation and interpretation of this orthonormal basis are given by Fišerová and Hron [2011]. The ilr-transformation avoids the linear dependency at the cost of some interpretability. The i-th coordinate of the ilr-transformed compositional vector can be seen as the average log-ratio of composition $x_{[i]}$ to all the "remaining" ones: $x_{[i+1]} \dots x_{[p+1]}$. Note that the composition $x_{[1]}$ takes a special role here from an interpretation point of view. The first coordinate of this ilr-transformation represents the average log-ratio between the composition $x_{[1]}$ and all the other ones. While the second coordinate for instance does not take into account the log-ratio $\ln\left(\frac{x_{[2]}}{x_{[1]}}\right)$. Therefore this type of ilr-transformation is referred to as pivot-coordinates by Filzmoser and Templ [2018]. One might choose a different "pivot" than the composition $x_{[1]}$, $x_{[2]}$ for instance, and use the first ilr-cooridnate to express the average log-ratio between composition $x_{[2]}$ and all other compositions. Since every permutation of the indices $\{1, \dots, p+1\}$ yields a different pivot-coordinates, there are $(p+1)!$ possibilities for such type of basis.

This is not a contradiction to perturbation invariance. In the ilr-transformation in Def. 5 the variable $x_{[1]}$ stands out just from an interpretation point of view. In fact every log-ratio can be expressed as a linear combination of ilr-coordinates. This also applies to the clr-coefficients:

Theorem 1 (Representation of centered log-ratios with the ilr-basis) *For the ilr-transformation in Def. 5 and the matrix $\boldsymbol{V}$ defined in Eq. 5 it holds that*

$$\boldsymbol{x}^{\boldsymbol{clr}} = \boldsymbol{V}\boldsymbol{x}^{\boldsymbol{ilr}}$$
$$\boldsymbol{x}^{\boldsymbol{ilr}} = \boldsymbol{V}^{\top}\boldsymbol{x}^{\boldsymbol{clr}}.$$

Let $\boldsymbol{V}_{[.,j]}$ denote the j-th column of $\boldsymbol{V} = (\boldsymbol{V}_{[.,1]}, \dots, \boldsymbol{V}_{[.,p]})$. $\boldsymbol{V}_{[.,j]}$ is given as

$$\boldsymbol{V}_{[.j]} = \left(\frac{p+1-j}{p-j+2}\right)^{\frac{1}{2}} \left(0, ..., 0, 1, -\frac{1}{p+1-j}, ..., -\frac{1}{p+1-j}\right)^{\top}. \qquad (5)$$

There is a close link between the isometric log-ratio transformation and the so-called Aitchinson geometry on the simplex, as it is referred to by Pawlowsky-Glahn and Egozcue [2001].

Definition 6 (Aitchinson geometry on the simplex) *The following operations: perturbation $\oplus$ (as addition), powering $\odot$ (as multiplication) and Aitchinson inner product $(.,.)_A$ define a vector space with inner product on the (unit) simplex.*

$$\boldsymbol{x} \oplus \boldsymbol{y} := (x_{[1]}y_{[1]}, ..., x_{[p+1]}y_{[p+1]})^{\top}$$
$$\alpha \odot \boldsymbol{x} := (x_{[1]}^{\alpha}, ..., x_{[p+1]}^{\alpha})^{\top}$$
$$(\boldsymbol{x}, \boldsymbol{y})_A := \frac{1}{2D}\sum_{i=1}^{p+1}\sum_{j=1}^{p+1} \ln\left(\frac{x_{[i]}}{x_{[j]}}\right) \ln\left(\frac{y_{[i]}}{y_{[j]}}\right)$$

Egozcue et al. [2015] for example show that these operations indeed yield a vector space with inner product.

An interesting observation is that the zero element of this vector space is the vector where all compositions are the same. Note, that these operations do not necessarily map onto the unit simplex, for example $(0.8, 0.2) \oplus (0.6, 0.4) = (0.48, 0.08)$ which does not lie on the unit simplex. The compositionally equivalent point on the unit simplex is about $(0.86, 0.14)$. Since from a compositional point of view these two vectors are the same, norming the output of the perturbation or powering is not necessary.

Theorem 2 (Isometry of the ilr-transformation)
For two compositional vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{S}^{p+1}$ and $\alpha \in \mathbb{R}$ it holds that:

$$\begin{aligned} ilr(\boldsymbol{x} \oplus \boldsymbol{y}) &= ilr(\boldsymbol{x}) + ilr(\boldsymbol{y}) \\ ilr(\alpha \odot \boldsymbol{x}) &= \alpha \ ilr(\boldsymbol{x}) \\ (\boldsymbol{x}, \boldsymbol{y})_A &= (ilr(\boldsymbol{x}), ilr(\boldsymbol{y})), \end{aligned}$$

where $(.,.)$ denotes the Euclidean inner product in $\mathbb{R}^p$.

Theorem 2 from Egozcue et al. [2003] is very useful to interpret linear transformations of the ilr-transformed compositions on the simplex $\mathbb{S}^{p+1}$.

3.3 Principal component analysis for compositional data

One example of a tool from multivariate statistics that is applied on the transformed compositions is principal component analysis (PCA). Once the compositional vectors are transformed form $\mathbb{S}^{p+1}$ to either $\mathbb{R}^p$ with the ilr-transformation or the $\mathbb{R}^{p+1}$ with the clr-transformation, PCA can be applied. As discussed in the previous section, the Euclidean metric is not the right tool for the compositional data vectors. On the ilr-transformed compositions for instance, PCA can be applied via an eigenvalue decomposition of the variance covariance matrix $\boldsymbol{\Sigma}$.

Definition 7 (Variance-covariance matrix)
The variance-covariance matrix for a stochastic vector $\boldsymbol{x} = [x_{[1]}, \dots, x_{[p]}]^\top \in \mathbb{R}^p$ is defined as

$$\boldsymbol{\Sigma} = \mathbb{E}\left[(\boldsymbol{x} - \mathbb{E}(\boldsymbol{x}))(\boldsymbol{x} - \mathbb{E}(\boldsymbol{x}))^\top\right].$$

Since the variance covariance matrix is a symmetric positive (semi)-definit matrix, it is possible to express it as in Th. 3.

Theorem 3 (Eigenvalue decomposition of $\boldsymbol{\Sigma}$)
The variance-covariance matrix $\boldsymbol{\Sigma}$ of a stochastic vector $\boldsymbol{x}$ with finite second moments can be expressed as

$$\boldsymbol{\Sigma} = \boldsymbol{\Gamma}\boldsymbol{D}\boldsymbol{\Gamma}^\top,$$

where $\boldsymbol{D}$ is a diagonal matrix containing the non-negative eigenvalues of $\boldsymbol{\Sigma}$ in decreasing order. The matrix $\boldsymbol{\Gamma}$ is called the loadings matrix whose columns are the orthonormal eigenvectors.

Definition 8 (Scores in PCA)
The projections of the vector $\boldsymbol{x}$ on the eigenvectors in $\boldsymbol{\Gamma}$ are called the scores $\boldsymbol{s}$ which can be obtained via

$$\boldsymbol{s} = \boldsymbol{\Gamma}^\top \boldsymbol{x}.$$

Due to the fact that the eigenvectors in $\boldsymbol{\Gamma}$ are orthonormal, the variance covariance matrix of the scores, $\boldsymbol{\Sigma}(\boldsymbol{s})$, is diagonal:

$$\boldsymbol{\Sigma}(\boldsymbol{s}) = \boldsymbol{\Gamma}^\top\boldsymbol{\Gamma}\boldsymbol{D}\boldsymbol{\Gamma}^\top\boldsymbol{\Gamma} = \boldsymbol{D}. \tag{6}$$

The variance of the scores are the eigenvalues of $\boldsymbol{\Sigma}$, contained in $\boldsymbol{D}$, which are sorted such that the first score has the highest variance, the second score has the second highest variance and so on.

In order to perform PCA on a data matrix, the variance-covariance matrix has to be estimated. In this thesis we used the classical estimator for the variance-covariance matrix and the robust covariance minimum determinant estimator.

Definition 9 (classical estimator of $\boldsymbol{\Sigma}$)
Let $\boldsymbol{X}_{[i,.]} \in \mathbb{R}^p$ be the n rows of the data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times p}$. Then the variance-covariance matrix can be estimated as

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{X}_{[i,.]} - \bar{\boldsymbol{x}})(\boldsymbol{X}_{[i,.]} - \bar{\boldsymbol{x}})^\top,$$
$$\bar{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{X}_{[i,.]}.$$

Definition 10 (Scatter matrix)
Let $\boldsymbol{x} \in \mathbb{R}^p$ be a stochastic vector. A matrix valued functional $\boldsymbol{S}(\boldsymbol{x})$ is called a scatter matrix, iff it is positive definite and for all vectors $\boldsymbol{b}$ and full-rank $(p \times p)$-matrices $\boldsymbol{A}$ it holds true that

$$\boldsymbol{S}(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}) = \boldsymbol{A}\boldsymbol{S}(\boldsymbol{x})\boldsymbol{A}^\top. \tag{7}$$

This property is called the affine equivariance property of the scatter matrix.

The variance-covariance matrix $\boldsymbol{\Sigma}$ is an example of a scatter matrix. Another scatter functional is the robust covariance minimum determinant estimator:

Definition 11 (covariance minimum determinant estimator)
Let $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ be a data matrix, whose n rows $\boldsymbol{X}_{[i,.]}$ represent realisations of the stochastic vector $\boldsymbol{x}$. The covariance minimum determinant estimator is a scatter $\boldsymbol{S}$. It is obtained as the classical variance covariance estimator from Def. 9 using a subset of the data matrix ($h < n$ rows). That way, for all possible subsets H of size h one obtains an estimator $\hat{\boldsymbol{\Sigma}}_H$. The minimum covariance determinant estimator is the one with the lowest determinant.

Since this estimator is calculated using only h rows of the data matrix it is very resistant to outliers. More details on this estimator are discussed in Hubert and Debruyne [2010].

If the stochastic vector $\boldsymbol{x}$ follows an elliptical symmetric distribution and the covariance matrix $\boldsymbol{\Sigma}$ of $\boldsymbol{x}$ exists, then every scatter matrix $\boldsymbol{S}$ is proportional to the covariance-matrix [Oja et al., 2006]:

$$\boldsymbol{\Sigma} \propto \boldsymbol{S} \quad \forall \boldsymbol{S}. \tag{8}$$

That means in that case, every scatter matrix will have the same eigenvectors as the covariance matrix. Therefore the estimated scores will be the same if a scatter matrix is used in PCA. This justifies using the covariance-minimum determinant estimator in PCA.

When performing PCA, one must not forget that it is performed on the transformed compositions and not on the actual compositions. Theorem 2 shows what this linear transformation in the ilr-space looks like on the simplex. In Filzmoser and Templ [2018] plots of these loadings are shown in the compositional space with the help of a ternary diagram. A ternary diagram is a way of showing 3-dimensional compositions in a 2D-diagram. Another way of interpreting the scores is to transform the loadings to clr-space with the help of the matrix $\boldsymbol{V}$ from Th. 1:

$$\boldsymbol{s} = \boldsymbol{\Gamma}\boldsymbol{x}^{ilr} = \boldsymbol{\Gamma}\boldsymbol{V}^{\top}\boldsymbol{x}^{\boldsymbol{clr}}. \tag{9}$$

4 Blind source separation

4.1 Basic model and explanations

Let $\boldsymbol{x}_t = [x_{t[1]}, \ldots, x_{t[p]}]^{\top} \in \mathbb{R}^p, t \in T$, be a p-variate time-series. In the basic blind source separation model one assumes that the observable time series $\boldsymbol{x}_t$ is a linear combination of the so called sources $\boldsymbol{z}_t \in \mathbb{R}^p$:

Definition 12 (Blind source separation, basic model)
In the basic blind source separation model one assumes the following relationship between the observable time series $\boldsymbol{x}_t$ and the sources $\boldsymbol{z}_t$:

$$\boldsymbol{x}_t = \boldsymbol{\mu} + \boldsymbol{\Omega}\boldsymbol{z}_t, \tag{10}$$

where the sources $\boldsymbol{z}_t$ fulfil certain assumptions, such that they can be estimated from the observable time series alone. All blind source separation models have one common assumption on $\boldsymbol{z}_t$:

$$\mathbb{E}(\boldsymbol{z}_t) = \mathbf{0} \quad \textit{and} \quad \mathbb{E}(\boldsymbol{z}_t^\top \boldsymbol{z}_t) = \boldsymbol{I}. \tag{11}$$

Apart from $\boldsymbol{\Omega} \in \mathbb{R}^{p \times p}$ having full rank, there are no further assumptions on the matrix $\boldsymbol{\Omega}$. Under the assumptions in Eq. 11, the parameter $\boldsymbol{\mu}$ in Eq. 10 has to be the mean of the time series $\boldsymbol{x}_t$. Since it has no further influence on the estimation of the sources, it can also be assumed to be zero.

Definition 13 (Unmixing matrix)
The goal of blind source separation (BSS) is to obtain the sources $\boldsymbol{z}_t$ via

$$\boldsymbol{z}_t = \boldsymbol{\Omega}^{-1}(\boldsymbol{x}_t - \boldsymbol{\mu}). \tag{12}$$

However in BSS the matrix $\boldsymbol{\Omega}^{-1}$ can only be obtained up to permutation and sign-changes of its rows.

A matrix $\boldsymbol{\Lambda}$ that is a solution of

$$\tilde{\boldsymbol{z}}_t = \boldsymbol{\Lambda}(\boldsymbol{x}_t - \boldsymbol{\mu}), \tag{13}$$

such that $\tilde{\boldsymbol{z}}_t$ meets all the assumptions on the sources in the model, is called an unmixing matrix. For an acquired unmixing matrix $\boldsymbol{\Lambda}$ yielding $\tilde{\boldsymbol{z}}_t$, see Eq. 13, changing the signs or permuting the latent time series $\tilde{z}_{t[i]}$ does not violate the assumptions made in the BSS model. Therefore multiplying $\boldsymbol{\Lambda}$ with a sign-change matrix $\boldsymbol{J}$ and or a permutation matrix $\boldsymbol{P}$,

$$\tilde{\boldsymbol{\Lambda}} = \boldsymbol{J}\boldsymbol{P}\boldsymbol{\Lambda} \tag{14}$$

still gives a valid unmixing matrix. Hence the problem of finding the true unmixing matrix $\boldsymbol{\Omega}^{-1}$ is not uniquely solvable. Every obtained unmixing matrix $\boldsymbol{\Lambda}$ and sources $\tilde{\boldsymbol{z}}_t$ will fulfil

$$\begin{aligned} \boldsymbol{\Omega}^{-1} &= \boldsymbol{J}\boldsymbol{P}\boldsymbol{\Lambda}, \\ \boldsymbol{z}_t &= \boldsymbol{J}\boldsymbol{P}\tilde{\boldsymbol{z}}_t, \end{aligned}$$

where $\boldsymbol{J}$ is an unknown sign change matrix and $\boldsymbol{P}$ is an unknown permutation matrix.

In order to estimate an unmixing matrix $\boldsymbol{\Lambda}$, further assumptions than the one in Eq. 11 need to be made on the sources. Different further assumptions on the sources lead to different BSS models. These assumptions also yield nice properties for $\boldsymbol{z}_t$, which makes them easier to model and interpret than $\boldsymbol{x}_t$.

4.2 The second order source separation model

The most established approach of BSS models for time series is the second order source separation (SOS) model. In this model one additional assumption is that the sources $\boldsymbol{z}_t$ are weakly stationary.

Definition 14 (Weakly stationarity)
Let $\boldsymbol{x}_t, t \in T$, be a p-variate time-series. Then $\boldsymbol{x}_t$ is called weakly stationary if and only if for all $t, s, t+h, s+h \in T$ the following assumptions hold.

$$\mathbb{E}[\boldsymbol{x}_t \boldsymbol{x}_t^\top] < \infty, \tag{15}$$
$$\mathbb{E}[\boldsymbol{x}_t] = \mathbb{E}[\boldsymbol{x}_s], \tag{16}$$
$$\mathbb{E}[\boldsymbol{x}_t \boldsymbol{x}_s^\top] = \mathbb{E}[\boldsymbol{x}_{t+h} \boldsymbol{x}_{s+h}^\top]. \tag{17}$$

Equations 16 and 17 imply

$$Cov(\boldsymbol{x}_t, \boldsymbol{x}_s) = Cov(\boldsymbol{x}_{t+h}, \boldsymbol{x}_{s+h}), \tag{18}$$

which means, the auto-covariance-matrix of the random vectors $\boldsymbol{x}_t$ and $\boldsymbol{x}_s$ does not depend on the times t and s, but only on the lag they are apart. That leads to the following definition:

Definition 15 (Autocovariance-matrix at lag $\boldsymbol{\tau}$)
The autocovariance-matrix $\boldsymbol{S}_\tau$ at lag τ of a weakly stationary time-series $\boldsymbol{x}_t$ is defined as

$$\boldsymbol{S}_\tau(\boldsymbol{x}_t) = \mathbb{E}\left[(\boldsymbol{x}_t - \mathbb{E}(\boldsymbol{x}_t))(\boldsymbol{x}_{t+\tau} - \mathbb{E}(\boldsymbol{x}_t))^\top\right].$$

With the above definition the assumptions on the sources in the SOS model can be stated.

Definition 16 (Second order source separation)
In the second order source separation model the sources $\boldsymbol{z}_t$ are assumed to be weakly stationary. Furthermore the following two assumptions are made on the sources:

$$\text{(SOS 1)} \quad \mathbb{E}(\boldsymbol{z}_t) = \boldsymbol{0} \quad \textit{and} \quad \mathbb{E}(\boldsymbol{z}_t^\top \boldsymbol{z}_t) = \boldsymbol{I} \tag{19}$$
$$\text{(SOS 2)} \quad \boldsymbol{S}_\tau(\boldsymbol{z}_t) \quad \textit{is diagonal for} \quad \tau = 1, 2, 3, \ldots \tag{20}$$

The assumption (SOS1) in Def. 16 means that the sources $\boldsymbol{z}_t$ have variance 1 and are uncorrelated at a fixed time. Note that for $\boldsymbol{x}_t^{st} = \boldsymbol{\Sigma}^{-\frac{1}{2}}(\boldsymbol{x}_t - \boldsymbol{\mu})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and the variance-covariance matrix of $\boldsymbol{x}_t$, $\boldsymbol{x}_t^{st}$ fulfils assumption (SOS1). This process is called whitening of the time series:

Definition 17 (Whitening of the time series)
Let $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ be the mean and the variance covariance matrix of the time-series $\boldsymbol{x}_t$. Then the whitened time series $\boldsymbol{x}_t^{st}$ is obtained as

$$\boldsymbol{x}_t^{st} = \boldsymbol{\Sigma}^{-\frac{1}{2}}(\boldsymbol{x}_t - \boldsymbol{\mu}).$$

Therefore the inverse square root of the variance-covariance matrix is the first building block for constructing an unmixing matrix $\boldsymbol{\Lambda}$ from Def. 13. The assumption (SOS2) means that there is no correlation between the latent sources $z_{t[i]}, \quad i = 1, \dots, p$ at any time.

Note that multiplying the whitened time series $\boldsymbol{x}_t^{st}$ with any orthonormal matrix $\boldsymbol{U}^\top$ does not violate assumption (SOS1). Therefore the idea of many second order source separation methods, and many BSS methods in general, is to find such a matrix $\boldsymbol{U}^\top$ to meet assumption (SOS2). The final unmixing matrix is then obtained as

$$\boldsymbol{\Lambda} = \boldsymbol{U}^\top \boldsymbol{\Sigma}^{-\frac{1}{2}}. \tag{21}$$

The property of the latent time series $z_{t[i]}$ being uncorrelated among each other justifies fitting time series models on each univariate source. This approach is much easier than modelling complex dependencies on the correlated time series $\boldsymbol{x}_t$.

In this thesis we will present two BSS methods for the SOS model, the AMUSE (algorithm for multiple unknown signals extraction) [Tong et al., 1990] and the SOBI (Second order blind identification)-method.

BSS Method 1 (AMUSE)
Let $\boldsymbol{x}_t$ be the observable time-series following the second order source separation model. In the AMUSE method, sources $\tilde{\boldsymbol{z}}_t$ are obtained by

$$\tilde{\boldsymbol{z}}_t = \boldsymbol{\Lambda}_\tau (\boldsymbol{x}_t - \boldsymbol{\mu}), \tag{22}$$

where $\boldsymbol{\Lambda}_\tau$ fulfils

$$\begin{aligned} \boldsymbol{\Lambda}_\tau \boldsymbol{\Sigma} \boldsymbol{\Lambda}_\tau^\top &= \boldsymbol{I} \\ \boldsymbol{\Lambda}_\tau \boldsymbol{S}_\tau(\boldsymbol{x}_t) \boldsymbol{\Lambda}_\tau^\top &= \boldsymbol{D}. \end{aligned} \tag{23}$$

$\boldsymbol{D}$ is a diagonal matrix with increasing elements on the diagonal, $\boldsymbol{S}_\tau$ is the auto-covariance matrix for a chosen lag τ and $\boldsymbol{\Sigma}$ is the variance covariance matrix of $\boldsymbol{x}_t$.

As discussed for instance in Nordhausen et al. [2020], the choice of the lag τ has a huge impact on the outcome of the AMUSE method. The SOBI method from [Belouchrani et al., 1997] tries to avoid this dependency by jointly diagonalising a set of auto-covariance matrices.

BSS Method 2 (SOBI)
Let $\boldsymbol{x}_t$ be the observable time series. The SOBI method first whitens the time series,

$$\boldsymbol{x}_t^{st} = \boldsymbol{\Sigma}^{-\frac{1}{2}} (\boldsymbol{x}_t - \boldsymbol{\mu}).$$

Then, the orthonormal matrix $\boldsymbol{U}$ is obtained as the matrix whose columns $\boldsymbol{U}_{[i,.]}$ maximize

$$\sum_{\tau \in L} \sum_{i=1}^{p} (\boldsymbol{U}_{[i,.]}^{\top} \boldsymbol{S}_{\tau}(\boldsymbol{x}_t^{st}) \boldsymbol{U}_{[i,.]})^2 \tag{24}$$

for a chosen set of lags $L = \{\tau_1, \dots \tau_k\}$. Using the above obtained matrix $\boldsymbol{U}$, the SOBI unmixing matrix is then acquired as

$$\boldsymbol{\Lambda} = \boldsymbol{U}^{\top} \boldsymbol{\Sigma}^{-\frac{1}{2}}.$$

Finding the matrix $\boldsymbol{\Lambda}_\tau$ in Eq. 23 in the AMUSE method is a joint diagonalisation problem for the matrices $\boldsymbol{\Sigma}$ and $\boldsymbol{S}_\tau(\boldsymbol{x}_t)$. Joint diagonalisation problems are frequently encountered within the BSS framework [Tichavsky and Yeredor, 2009]. In Miettinen et al. [2017] various joint diagonalisation problems in the BSS framework are presented as well as algorithms for approximate joint diagonalisation. Different requirements have to be fulfilled by the matrices whether there are 2 or more than 2 matrices that have two be diagonalised.

The problem of diagonalising two matrices, like in the AMUSE case, has a solution, if $\boldsymbol{\Sigma}$ is a positive definite and symmetric matrix, and $\boldsymbol{S}_\tau$ is a symmetric matrix. In the SOS model these requirements are fulfilled both for the theoretical matrices $\boldsymbol{\Sigma}$ and $\boldsymbol{S}_\tau$, and for their estimators $\hat{\boldsymbol{\Sigma}}$ from Def. 9 and $\hat{\boldsymbol{S}}_\tau$ from Def. 34. If the diagonal elements in $\boldsymbol{D}$ from Eq. 23 are distinct then the unmixing matrix $\boldsymbol{\Lambda}_\tau$ is unique up to permutation and sign changes of the rows [Miettinen et al., 2017]. These diagonal elements are the autocovariance-functions of the latent sources $z_{t[i]}$ at lag τ. Therefore a requirement for the AMUSE method is that no two latent sources $z_{t[i]}$ and $z_{t[j]}$ have the same autocovariance for the chosen lag τ.

The problem of jointly diagonalising more than two matrices arises in the SOBI case. In the SOBI method one needs to solve the joint diagonalisation problem:

$$\boldsymbol{U} \boldsymbol{\Sigma}(\boldsymbol{x}_t) \boldsymbol{U}^{\top} = \boldsymbol{I}_p \tag{25}$$

and for $\tau \in L$

$$\boldsymbol{U} \boldsymbol{S}_\tau(\boldsymbol{x}_t) \boldsymbol{U}^{\top} = \boldsymbol{D}_\tau. \tag{26}$$

Diagonalising a set of $k > 2$ symmetric matrices is only possible if the matrices commute [Miettinen et al., 2017], meaning

$$\boldsymbol{S}_{\tau_i} \boldsymbol{S}_{\tau_j} = \boldsymbol{S}_{\tau_j} \boldsymbol{S}_{\tau_i}. \tag{27}$$

In the SOS model this is the case for the true theoretical matrices $\boldsymbol{S}_\tau$, however for estimated ones, this is not necessarily the case. Therefore in the SOBI method an approximate joint diagonalisation algorithm is used. The maximisation problem in Method 2 is such an approximate joint diagonalisation approach, for more details see Miettinen et al. [2017]. This means in the SOBI method, the estimated autocovariance matrices $\hat{\boldsymbol{S}}_\tau(\hat{\boldsymbol{z}}_t)$ will not be true diagonal matrices, but the approximate joint diagonalisation algorithm makes them as diagonal as possible.

For the SOBI algorithm the variance-covariance matrix $\boldsymbol{\Sigma}$, the location vector $\boldsymbol{\mu}$ and the auto-covariance matrices $\boldsymbol{S}_\tau$ for the lags $\tau \in L$ need to be estimated. Estimators for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ have been shown already in the section about PCA. In the second order source separation model we have to use the symmetrisized estimator for the auto-covariance matrix $\boldsymbol{S}_\tau$.

Definition 18 (Symmetrisized estimator for $\boldsymbol{S}_\tau$)
Let $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ be a data matrix whose n rows, $\boldsymbol{X}_{[i,.]}$ represent realisations of the p-variate time-series $\boldsymbol{x}_t$. Then the symmetrisized estimator $\hat{\boldsymbol{S}}_\tau$ for $\boldsymbol{S}_\tau(\boldsymbol{x}_t)$ is acquired as

$$\hat{\boldsymbol{A}}_\tau = \frac{1}{n-\tau}\sum_{i=1}^{n-\tau}(\boldsymbol{X}_{[i,.]} - \bar{\boldsymbol{x}})(\boldsymbol{X}_{[i+\tau,.]} - \bar{\boldsymbol{x}})^\top \tag{28}$$

$$\hat{\boldsymbol{S}}_\tau = \frac{1}{2}\left(\hat{\boldsymbol{A}}_\tau + \hat{\boldsymbol{A}}_\tau^\top\right). \tag{29}$$

The estimator $\hat{\boldsymbol{A}}_\tau$ in Eq. 28 is the classical estimator for the auto-covariance matrix, $\bar{\boldsymbol{x}}$ is the arithmetic mean. The second calculation in Eq. 29 is called symmetrisizing the classical estimator for the autocovariance matrix. Since the matrix $\boldsymbol{S}_\tau$ is symmetric, the estimator for it should also be a symmetric matrix.

A common way to robustify the SOBI-method is to replace the auto-covariance matrix with the spatial sign auto-covariance matrix, which will be defined in the following Def. 19, and whiten the data with a robust estimator for the variance-covariance matrix. For more details on robustification of the SOBI method view Ilmonen et al. [2015]. Their paper contains a review of robustification of SOBI as well as suggestions for improvements.

Definition 19 (Spatial sign auto-covariance matrices)
Let $\boldsymbol{x}_t$ be a centered time series. The spatial sign auto-covariance matrix is defined as

$$\boldsymbol{R}_\tau = \mathbb{E}\left[\frac{\boldsymbol{x}_t}{||\boldsymbol{x}_t||}\frac{\boldsymbol{x}_{t+\tau}^\top}{||\boldsymbol{x}_{t+\tau}||}\right].$$

It can be estimated using the following estimator:

Definition 20 (Symmetrisized estimator for $\boldsymbol{R}_\tau$)
Let $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ be a data matrix whose n rows, $\boldsymbol{X}_{[i,.]}$ represent realisations of the centered p-variate time-series $\boldsymbol{x}_t$. Then the classical estimator $\hat{\boldsymbol{R}}_\tau$ for $\boldsymbol{R}_\tau(\boldsymbol{x}_t)$ is acquired as

$$\boldsymbol{A}_\tau = \frac{1}{|T|-\tau}\sum_{i=1}^{|T|-\tau}\frac{\boldsymbol{X}_{[i,.]}}{||\boldsymbol{X}_{[i,.]}||}\frac{\boldsymbol{X}_{[i+\tau,.]}^\top}{||\boldsymbol{X}_{[i+\tau,.]}||} \tag{30}$$

$$\hat{\boldsymbol{R}}_\tau = \frac{1}{2}\left(\boldsymbol{A}_\tau + \boldsymbol{A}_\tau^\top\right). \tag{31}$$

BSS Method 3 (robust SOBI)
In the robust SOBI method the time series $\boldsymbol{x}_t$ are robust whitened, using the robust covariance minimum determinant estimator $\hat{\boldsymbol{\Sigma}}_{mcd}$, and the corresponding location $\hat{\boldsymbol{\mu}}_{mcd}$. Furthermore let $\boldsymbol{U}$ be the orthonormal matrix , whose columns $\boldsymbol{U}_{[.,i]}$ maximize

$$\sum_{\tau \in L} \sum_{i=1}^{p} (\boldsymbol{U}_{[.,i]}^{\top} \boldsymbol{R}_{\tau}(\boldsymbol{x}_t^{st}) \boldsymbol{U}_{[.,i]})^2. \tag{32}$$

for a chosen set of lags $L = \{\tau_1, \dots \tau_k\}$. Once again the final unmixing matrix is acquired as

$$\boldsymbol{\Lambda} = \boldsymbol{U}^{\top} \boldsymbol{\Sigma}_{mcd}^{-\frac{1}{2}}$$

It is important to point out, that in this method we need the additional assumption that the observable time series $\boldsymbol{x}_t$ has a symmetrical distribution.

4.3 The stochastic independent component model

The stochastic independent component model, introduced by Matilainen et al. [2015], was designed to be able to model time series with stochastic volatility within the BSS framework. In this model the following assumptions are put on the sources:

Definition 21 (Stochastic independent component model)
Let $\boldsymbol{z}_t$ be the sources of the blind source separation model. In the stochastic independent component model, $\boldsymbol{z}_t$ is assumed to fulfil the following assumptions:

$$\mathbb{E}(\boldsymbol{z}_t \boldsymbol{z}_t^{\top}) = \boldsymbol{I}_p \quad \text{and } \mathbb{E}(\boldsymbol{z}_t) = \boldsymbol{0}. \tag{33}$$

Furthermore, each component of $\boldsymbol{z}_t$ exhibits stochastic volatility features and has finite fourth moments and crossmoments. No two components are identical at all lags.

Time series with stochastic volatility is a class of time series models, where the variance of the process is a random variable, modelled with a stochastic process. The basic SV model assumes that the time series z_t fulfils the following assumptions: [Shephard and Andersen, 2009]

Definition 22 (Time series with stochastic volatility)
Let z_t be a univariate time-series. The basic stochastic volatility model assumes that the time series z_t fulfils

$$z_t = \sigma_t \epsilon_t, \tag{34}$$

where σ_t and ϵ_t are independent. The process σ_t is non-negative and the process ϵ_t is an autoregressive process with mean zero and variance σ^2.

Different SV models come from different modelling of σ_t and ϵ_t. A famous SV model for example is the GARCH(p,q) model, where it is assumed that σ_t fulfils the difference equation:

$$\sigma_t^2 = \delta + \alpha_1 x_{t-1}^2 + \ \dots \ + \alpha_q x_{t-q}^2 + \beta_1 \sigma_{t-1}^2 + \ \dots \ + \beta_p \sigma_{t-p}^2, \tag{35}$$

and ϵ_t is an IID-process with mean 0, see Def. 23. For more details, view for example He and Teräsvirta [1999]. An IID-Process is a sequence of identically distributed random variables:

Definition 23 (IID-Process)
A process $\epsilon_t, t \in T$, is called an iid-process with mean μ and variance σ^2, iff for every pairwise different time points $t_1, \dots, t_k, \in T$, the random variables $\epsilon_{t_1}, \dots, \epsilon_{t_k}$, are independent and identically distributed (iid), having a mean of μ and a variance of σ^2.

One method to estimate sources in a stochastic volatility independent component model is with the gFOBI (generalized fourth order blind identification) method. Instead of the autocovariance matrices as used in SOBI, it uses the fourth order cross-moments matrices, as defined in the following Def. 24 of the whitened time series in the joint diagonalisation problem. For more details view for example [Matilainen et al., 2017].

Definition 24 (Fourth order cross-moment matrix)
Let $\boldsymbol{x}_t^{st}$ be a p-variate whitened time-series. The fourth order cross-moment matrix $\boldsymbol{B}_\tau$ at lag τ of $\boldsymbol{x}_t^{st}$ is defined as

$$\boldsymbol{B}_\tau = \mathbb{E}[\boldsymbol{x}_{t+\tau}^{st} \boldsymbol{x}_t^{st\top} \boldsymbol{x}_t^{st} \boldsymbol{x}_{t+\tau}^{st\top}]. \tag{36}$$

Let $x_{t[i]}^{st}$ be the i-th element of the p-variate time series $\boldsymbol{x}_t^{st}$. The (i,j)-th element of $\boldsymbol{B}_\tau$, $\boldsymbol{B}_{\tau[i,j]}$ is given as

$$\boldsymbol{B}_{\tau[i,j]} = \mathbb{E}((\sum_{l=1}^{p} x_{t[l]}^{st2}) x_{t+\tau[i]}^{st} x_{t+\tau[j]}^{st}). \tag{37}$$

With the just defined fourth-order cross-moment matrix, the gFOBI-method from Matilainen et al. [2015] can be described.

BSS Method 4 (gFOBI)
Let $\boldsymbol{x}_t$ be the observed time-series. The gFOBI algorithm first whitens the time-series $\boldsymbol{x}_t$ and then uses the fourth order cross-moment matrices at lags $\tau \in L$ of the whitened time-series in the joint diagonalisation problem. Let $\boldsymbol{U}$ be the orthonormal matrix whose columns $\boldsymbol{U}_{[.,i]}$ maximize

$$\sum_{\tau \in T} \sum_{i=1}^{p} (\boldsymbol{U}_{[.,i]}^{\top} \boldsymbol{B}_\tau(\boldsymbol{x}_t^{st}) \boldsymbol{U}_{[.,i]})^2 \tag{38}$$

for a chosen lag set $L = \{\tau_1, ..., \tau_k\}$. The gFOBI unmixing matrix is then obtained as

$$\boldsymbol{\Lambda} = \boldsymbol{U}^\top \boldsymbol{\Sigma}^{-\frac{1}{2}}.$$

In order to calculate a gFOBI model one needs to estimate the matrices $\boldsymbol{B}_\tau$.

Definition 25 (Estimator for the fourth order cross-moment matrix) *Let $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ be the data matrix whose n rows $\boldsymbol{X}_{[i,.]}$ represent realisations of the time-series $\boldsymbol{x}_t$. Furthermore let $\boldsymbol{X}^{st}$ be the whitened data matrix*

$$\boldsymbol{X}^{st} = \boldsymbol{X}\hat{\boldsymbol{\Sigma}}^{-\frac{1}{2}\top} \tag{39}$$

with elements $X^{st}_{[i,j]}$. Then the (k,j)-th element of the estimated fourth order cross-moment matrix $\hat{\boldsymbol{B}}_{\tau[k,j]}$ is given as

$$\hat{\boldsymbol{B}}_{\tau[k,j]} = \frac{1}{|T| - \tau} \sum_{i=1}^{|T|-\tau} \Big(\sum_{l=1}^{p} (X^{st}_{[i,l]})^2\Big) X^{st}_{[i+\tau,k]} X^{st}_{[i+\tau,j]}. \tag{40}$$

4.4 Non-stationary source separation

In the Non-stationary model, the assumption of weakly stationarity is relaxed. In this model the variance of the sources can change over time.

Definition 26 (Non stationary source separation model) *In the non stationary source separation model the following assumptions on the sources $\boldsymbol{z}_t$ hold:*

$$\mathbb{E}(\boldsymbol{z}_t) = \boldsymbol{0} \quad \forall t, \tag{41}$$

$$\mathbb{E}(\boldsymbol{z}_t \boldsymbol{z}_t^\top) \quad \textit{is positive definite and diagonal} \quad \forall t, \tag{42}$$

$$\mathbb{E}(\boldsymbol{z}_t \boldsymbol{z}_{t+\tau}^\top) \textit{ is diagonal} \quad \forall t, \tau. \tag{43}$$

Eq. 43 replaces the weakly stationarity assumption and allows for the autocovariance matrix to change over time. However, also in this model there should be no correlation between the sources at any time. In Choi and Cichocki [2000] three different methods are presented that find an unmixing matrix in the non-stationary source separation model. One of them is reffered to as NSS-JD method and involves their whitening approach using the auto-covariance matrix at a lag $\tau \neq 0$ as well as autocovariance matrices $\boldsymbol{S}_{T_i,\tau}$ at lag τ and time period $T_i \subset T$, which are defined in Def. 27. In this thesis the adjustment of the method as presented in Nordhausen et al. [2020], where the auto-covariance matrix is used for whitening.

Definition 27 (Autocovariance matrix for time period $\boldsymbol{T_i}$) *Let $\boldsymbol{x}_t^{st}, t \in T$, be a p-variate whitened time series and $T_i \subset T$ be a finite time interval. The autocovariance matrix for $\boldsymbol{x}_t^{st}$ at lag τ for time period T_i: $\boldsymbol{S}_{T_i,\tau}$ is defined as*

$$\boldsymbol{S}_{T_i,\tau} = \mathbb{E}(\boldsymbol{x}_t^{st} \boldsymbol{x}_{t+\tau}^{st\top} | \quad t, t+\tau \in T_i). \tag{44}$$

With this definition the NSS-JD method can be described:

BSS Method 5 (NSS-JD)
Let $\boldsymbol{x}_t$ be the observable time series in a non-stationary blind source separation model. The NSS-JD method first whitens the time series:

$$\boldsymbol{x}_t^{st} = \boldsymbol{\Sigma}^{-\frac{1}{2}}(\boldsymbol{x}_t - \boldsymbol{\mu}). \tag{45}$$

Then for the chosen lag τ and K disjunct time intervals T_i:

$$T = \dot{\bigcup}_{i=1}^{K} T_i, \tag{46}$$

the auto-covariance matrices $\boldsymbol{S}_{T_i,\tau}$ at lag τ for each time interval are used in the joint diagonalisation problem. Let $\boldsymbol{U}$ be the matrix, whose columns $\boldsymbol{U}_{[.,i]}$ maximize

$$\sum_{i=1}^{K}\sum_{i=1}^{p}(\boldsymbol{U}_{[.,i]}^{\top}\boldsymbol{S}_{T_i,\tau}(\boldsymbol{x}_t^{st})\boldsymbol{U}_{[.,i]})^2. \tag{47}$$

Then the NSS-JD unmixing matrix $\boldsymbol{\Lambda}$ is defined as

$$\boldsymbol{\Lambda} = \boldsymbol{U}^{\top}\boldsymbol{\Sigma}^{-\frac{1}{2}}. \tag{48}$$

4.5 A new BSS method

Nordhausen et al. [2020] recently published a novel BSS method, the NSS-SOBI-gFOBI-method. It is a hybrid method that can be used for the second order source separation model, for the stochastic independent component model and for the non stationary source separation (NSS) model. Therefore it is suggested, when the underlying BSS model is not known.

The NSS-SOBI-gFOBI-method uses both the autocorrelation matrices $\boldsymbol{S}_\tau$ and the fourth order cross-moment matrices $\boldsymbol{B}_\tau$ in the joint diagonalisation problem. It also uses the idea of dividing the time period T into K non-overlapping time periods:

$$T = \dot{\bigcup}_{i=1}^{K} T_i, \tag{49}$$

and uses the autocovariance matrix $\boldsymbol{S}_{T_i,\tau}$ and the fourth order cross-moment matrix $\boldsymbol{B}_{T_i,\tau}$ for each time interval T_i. The fourth order cross-moment matrix $\boldsymbol{B}_{T_i,\tau}$ at lag τ and time interval T_i is defined in a similar way as $\boldsymbol{S}_{T_i,\tau}$:

Definition 28 (Fourth order cross-moment matrix for time period $\boldsymbol{T_i}$)
Let $\boldsymbol{x}_t^{st}, t \in T$, be a p-variate whitened time series. The fourth order cross-moment matrix for $\boldsymbol{x}_t^{st}$ at lag τ for time period T_i, $\boldsymbol{B}_{T_i,\tau}$ is defined as

$$\boldsymbol{B}_{T_i,\tau} = \mathbb{E}[\boldsymbol{x}_{t+\tau}^{st}\boldsymbol{x}_t^{st\top}\boldsymbol{x}_t^{st}\boldsymbol{x}_{t+\tau}^{st\top}| \quad t, t+\tau \in T_i)] \tag{50}$$

With the above definition the NSS-SOBI-gFOBI method can be described as:

BSS Method 6 (NSS-SOBI-gFOBI)
Let $\boldsymbol{x}^{st}$ be the whitened time-series. Furthermore let $L_s = \{\tau_{s,1}, \dots, \tau_{s,L_s}\}$ be the lag set for the autocovariance matrices and $L_b = \{\tau_{b,1}, \dots, \tau_{b,L_b}\}$ be the lag set for the fourth order cross-moment matrices.

Let

$$\boldsymbol{V}_{T_i,j} = \left\{ \begin{array}{ll} \boldsymbol{S}_{T_i,\tau_{s,j}} & \textit{for } j = 1, \dots, L_s \\ \boldsymbol{B}_{T_i,\tau_{b,j-L_s}} & \textit{for } j = L_s + 1, \dots, L_s + L_b \end{array} \right\}$$

denote the matrices used in the joint diagonalisation problem. Then the matrix $\boldsymbol{U}$ can be defined as the orthonormal matrix, whose columns $\boldsymbol{U}_{[.,i]}$ maximize

$$\sum_{l=1}^{K} \sum_{j=1}^{L_s+L_b} \sum_{i=1}^{p} (\boldsymbol{U}_{[.,i]}^{\top} \alpha_j \boldsymbol{V}_{T_i,j} \boldsymbol{U}_{[.,i]})^2. \tag{51}$$

The NSS-SOBI-gFOBI-unmixing-matrix is then obtained as

$$\boldsymbol{\Lambda} = \boldsymbol{U}^{\top} \boldsymbol{\Sigma}^{-\frac{1}{2}}.$$

Nordhausen et al. [2020] introduced weights α_j in the maximisation problem. This is because of the fact that matrices in the joint diagonalisation problem come from different families. Since the fourth order cross-moment matrices tend to have a bigger determinant than the auto-covariance matrices, one might run the risk that the fourth order cross-moment matrices dominate in the maximisation problem if all matrices are given the same weight. Nordhausen et al. [2020] suggest

$$\begin{aligned} \alpha_j &= \frac{1}{p+2} \quad \text{if } \boldsymbol{V}_{T_i,j} \text{ is a fourth order cross-moment matrix} \\ \alpha_j &= 1 \quad \text{otherwise} \end{aligned} \tag{52}$$

or

$$\begin{aligned} \alpha_j &= \frac{1}{\max|\boldsymbol{V}(T_i,j)|} \quad \text{if } \max|\boldsymbol{V}(T_i,j)| > 1 \\ \alpha_j &= 1 \quad \text{otherwise} \end{aligned} \tag{53}$$

as weights. The weights in Eq. 52 are suggested since scaling the fourth-moments matrix,$\boldsymbol{B}_0$, in a multivariate normal model by $\frac{1}{p+2}$ yields a consistent estimator for the covariance matrix [Nordhausen et al., 2020]. The idea of the weights defined in Eq. 53 is to downweight large matrices while not upweighting zero-matrices. In the recently submitted paper Nordhausen et al. [2020], time series of different BSS-models were simulated and the sources were estimated using the different BSS methods. Then the estimated sources were compared to the simulated ones using a suitable comparison metric. Even though their new hybrid method did not beat the method of the corresponding BSS model, it was significantly better than the methods from the other BSS models. Therefore this method is suggested if it is unclear which BSS model is suitable.

5 Blind source separation for compositional time series

In this section we are going to combine the methods of compositional data analysis and blind source separation.

Definition 29 (Compositional time series)
A time series $\boldsymbol{x}_t, t \in T$, is a compositional time series, if for every $s \in T$, $\boldsymbol{x}_s$ is a compositional vector.

In previous papers of compositional time-series, time series models were built on the log-ratio transformed data. Then the inverse transformation was applied for instance on the predictions in the transformed space, in order to obtain predictions in terms of compositional data on the simplex, view for example Silva and Smith [2001] or Brunsdon and Smith [1998], where repeated surveys were modelled. More recent work on compositional time series can be found for instance in Dawson et al. [2014], Bergman and Holmquist [2014] or in Kynčlová et al. [2015].

This is also the approach we suggest for blind source separation for compositional time-series: Apply a log-ratio transformation first and then build a blind source separation model on the transformed data. We suggest to use the isometric log-ratio transformation in order to avoid the singularity issue, and to use the linear relationship between the ilr-transformation and the clr-transformation to obtain loadings in clr-space which can be interpreted in terms of the original compositions.

Definition 30 (Ilr- and Clr- transformed time series)
Let $\boldsymbol{x}_t \in \mathbb{S}^{p+1}, t \in T$, be a compositional time series. By applying the ilr-transformation from Eq. 5 one obtains the ilr-transformed time-series $\boldsymbol{x}_t^{ilr}$ as

$$\boldsymbol{x}_t^{ilr} = ilr(\boldsymbol{x}_t),$$

and the clr-transformed time series $\boldsymbol{x}_t^{clr}$ by using the clr-tranformation from Eq. 3:

$$\boldsymbol{x}_t^{clr} = clr(\boldsymbol{x}_t). \tag{54}$$

Then a blind source separation model can be built using the ilr-transformed time-series $\boldsymbol{x}_t^{ilr}$ as the observable time series in the BSS model:

$$\boldsymbol{z}_t = \boldsymbol{\Omega}\boldsymbol{x}_t^{ilr} + \boldsymbol{\mu}. \tag{55}$$

For principal component analysis of compositional data both the ilr- and the clr-transformation can be used to transform the data from the simplex to the $\mathbb{R}^{p+1}$ or the $\mathbb{R}^p$. However, for blind source separation for compositional time-series, an ilr-transformation has to be used. The reason is, that the whitening process

$$\boldsymbol{x}_t^{st} = \boldsymbol{\Sigma}^{-\frac{1}{2}}(\boldsymbol{x}_t - \boldsymbol{\mu})$$

requires the covariance matrix $\boldsymbol{\Sigma}$ to be invertible, which is not the case for the clr-transformed time series. Like in PCA for compositional data one must not forget that the blind source separation model is built on the transformed compositions.

Let $\boldsymbol{\Lambda}$ be an estimated unmixing matrix from a BSS model yielding

$$\tilde{\boldsymbol{z}}_t = \boldsymbol{\Lambda}(\boldsymbol{x}_t^{ilr} - \hat{\boldsymbol{\mu}}^{ilr}), \tag{56}$$

where $\hat{\boldsymbol{\mu}}^{ilr}$ is the estimated mean of the ilr-transformed time series. Using the linear transformation $\boldsymbol{V}$ between the ilr-transformed time series and the clr-transformed time series (see Th. 1), we obtain an unmixing matrix for the clr-transformed time series:

$$\tilde{\boldsymbol{z}}_t = \boldsymbol{\Lambda}(\boldsymbol{x}_t^{ilr} - \hat{\boldsymbol{\mu}}^{ilr}) = \boldsymbol{\Lambda}\boldsymbol{V}^\top(\boldsymbol{x}_t^{clr} - \hat{\boldsymbol{\mu}}^{clr}) = \tilde{\boldsymbol{\Lambda}}(\boldsymbol{x}_t^{clr} - \hat{\boldsymbol{\mu}}^{clr}). \tag{57}$$

The estimated mean $\hat{\boldsymbol{\mu}}^{clr}$ of the clr-transformed time-series from Eq. 57 has to be obtained as

$$\hat{\boldsymbol{\mu}}^{clr} = \boldsymbol{V}^\top \hat{\boldsymbol{\mu}}^{ilr} \tag{58}$$

for Eq. 57 to be true. The clr-unmixing matrix $\tilde{\boldsymbol{\Lambda}} \in \mathbb{R}^{(p+1)\times p}$ contains the information of how the clr-transformed time series is linearly combined into the estimated sources $\tilde{\boldsymbol{z}}_t$. Using the clr-loadings, the estimated sources $\tilde{\boldsymbol{z}}_t$ are easier to interpret in terms of the original compositions, since it is easier to trace back the compositions from the clr-transformation. Recall that the i-th coordinate of the clr-transformed compositional vector is the average log-ratio of composition $x_{t[i]}$ and all the other ones. So for instance, if the (i,j)-th element $\tilde{\boldsymbol{\Lambda}}_{[i,j]}$ of the clr-loadings matrix $\tilde{\boldsymbol{\Lambda}}$ has a high absolute value compared to all the other elements in the i-th row, it means that the latent estimated source $\tilde{z}_i$ represents the relative value between composition $x_{t[i]}$ and the other compositions.

We want to emphasise on the fact that the estimated sources $\tilde{\boldsymbol{z}}_t$ do not depend on the chosen ilr-transformation. Every ilr-transformation is just a representation of the compositions within an orthonormal basis in the $\mathbb{R}^p$, and the results of the BSS methods we described does not depend on the choice of basis. Let $\boldsymbol{x}_t, t \in T$, be a p-variate time series, and let $\tilde{\boldsymbol{x}}_t$ be the same time series expressed within a different basis. Then

$$\tilde{\boldsymbol{x}}_t = \boldsymbol{W}\boldsymbol{x}_t \tag{59}$$

holds for some full-rank matrix $\boldsymbol{W}$. For instance, in the SOS model with the joint diagonalisation approach as in the SOBI or in the JADE method, the unmixing matrix $\mathbf{\Lambda}$ fulfils the affine equivariance property [Miettinen et al., 2014]. That means, if

$$\tilde{\boldsymbol{x}}_t = \boldsymbol{W}\boldsymbol{x}_t, \tag{60}$$

is a linear transformation of the time series $\boldsymbol{x}_t$ with some full-rank matrix $\boldsymbol{W}$, the estimated unmixing matrix in the SOS model (using the methods in this thesis) for the transformed time-series $\tilde{\boldsymbol{x}}_t$, $\mathbf{\Lambda}_W$, fulfils

$$\mathbf{\Lambda}_W = \boldsymbol{P}\boldsymbol{J}\mathbf{\Lambda}\boldsymbol{W}^{-1}, \tag{61}$$

where $\boldsymbol{P}$ is a permutation matrix and $\boldsymbol{J}$ is a sign change-matrix. Since in BSS the unmixing matrix can only be obtained up to a permuation and a sign change matrix anyway, the choice of basis has no influence on the estimation of the sources.

5.1 Blind source separation for highly correlated compositional time series

In blind source separation for compositional time series the ilr-transformation from the simplex $\mathbb{S}^{p+1}$ to $\mathbb{R}^p$ is used to avoid the issue of the singularity of the variance covariance matrix. In case the ilr-data is highly correlated, a reduction of dimensionality needs to be applied on the ilr-transformed data before it can be used as the observable time-series in the BSS models.

Definition 31 (Scores-time-series)
Let $\boldsymbol{x}_t^{ilr} \in \mathbb{R}^p, t \in T$, be an ilr-transformed compositional time-series with a singular variance covariance matrix $\mathbf{\Sigma}(\boldsymbol{x}_t^{ilr})$. Performing PCA on the ilr-time-series yields the diagonal matrix $\boldsymbol{D}$ and the loading matrix $\mathbf{\Gamma}$ fulfilling

$$\mathbf{\Sigma}(\boldsymbol{x}_t^{ilr}) = \mathbf{\Gamma}\boldsymbol{D}\mathbf{\Gamma}^\top, \tag{62}$$

as well as scores time series

$$\boldsymbol{s}_t = \mathbf{\Gamma}^\top \boldsymbol{x}_t^{ilr}. \tag{63}$$

Recall that the scores are ordered in such a way that the first score has the highest variance, the second source has the second highest variance and so on. In the case that $\mathbf{\Sigma}$ is singular, there will be scores $s_{t[i]}$ that have zero variance. In order to perform a dimension reduction use only the first $l < p$ scores, such that the used scores have a positive variance. There are certain rules for choosing $l < p$ components in PCA. Let $d_1 > d_2 > \ldots > d_p$ be the diagonal elements of $\boldsymbol{D}$. A popular rule of choosing l components is to choose them in a way, such that the cummulative explained variance is bigger than a threshold c, i.e.

$$\frac{\sum_{i=1}^{l} d_i}{\sum_{j=1}^{p} d_j} > c. \tag{64}$$

Let $\boldsymbol{s}_t^l$ be the first $l < p$ scores series chosen and $\mathbf{\Gamma}_l \in \mathbb{R}^{p\times l} = [\mathbf{\Gamma}_{[.,1]}, \ldots, \mathbf{\Gamma}_{[.,l]}]$ be the matrix that contains only the first l eigenvectors of the loadings matrix fulfilling

$$\boldsymbol{s}_t^l = \boldsymbol{\Gamma}_l^\top \boldsymbol{x}_t^{ilr}. \tag{65}$$

These l scores can now be used as the observable time series in the BSS models:

$$\boldsymbol{s}_t^l = \boldsymbol{\mu} + \boldsymbol{\Omega} \boldsymbol{z}_t, \tag{66}$$

since the singularity issue is resolved because $\det(\boldsymbol{\Sigma}(\boldsymbol{s}_t^l)) = \prod_{j=1}^l d_j > 0$. Now applying a BSS-method of choice leads to an estimated unmixing matrix $\boldsymbol{\Lambda}$ as well as estimated sources $\tilde{\boldsymbol{z}}_t$:

$$\tilde{\boldsymbol{z}}_t = \boldsymbol{\Lambda}(\boldsymbol{s}_t^l - \hat{\boldsymbol{\mu}^l}), \tag{67}$$

where $\hat{\boldsymbol{\mu}}^l$ is an estimated mean of the first l scores $\boldsymbol{s}_t^l$. The loadings matrix $\boldsymbol{\Gamma}_l$ can be used to obtain an unmixing matrix in terms of the ilr-transformed time series:

$$\tilde{\boldsymbol{z}}_t = \boldsymbol{\Lambda}(\boldsymbol{s}_t^l - \hat{\boldsymbol{\mu}^l}) = \boldsymbol{\Lambda}\boldsymbol{\Gamma}_l^\top \boldsymbol{x}_t^{ilr}. \tag{68}$$

Using the matrix $\boldsymbol{V}$ from Eq. 5 we obtain the unmixing matrix in terms of the clr-tranformed time series:

$$\begin{aligned} \tilde{\boldsymbol{z}}_t &= \boldsymbol{\Lambda}(\boldsymbol{s}_t^l - \hat{\boldsymbol{\mu}^l}) = \boldsymbol{\Lambda}\boldsymbol{\Gamma}_l^\top (\boldsymbol{x}_t^{ilr} - \hat{\boldsymbol{\mu}}^{ilr}) \\ &= \boldsymbol{\Lambda}\boldsymbol{\Gamma}_l^\top \boldsymbol{V}^\top (\boldsymbol{x}_t^{clr} - \hat{\boldsymbol{\mu}}^{ilr}) = \tilde{\boldsymbol{\Lambda}}(\boldsymbol{x}_t^{clr} - \hat{\boldsymbol{\mu}}^{ilr}). \end{aligned} \tag{69}$$

Again, the clr-unmixing matrix $\tilde{\boldsymbol{\Lambda}}$ can be used for an easier interpretation of the sources $\tilde{\boldsymbol{z}}_t$.

6 Application on spectrometric data

In this section we explain how we apply our new methodology to absorption data measured at a small stream in Lower Austria.

6.1 Data acquisition

The data was collected in 2014 by Matthias Pucher from the University of Natural Resources and Life Sciences in Vienna. Employees of the department for water management installed a spectrometer (s::can spectro::lyser by Messtechnik GmbH in Austria) at a stream near Petzenkrichen in Lower Austria, see figure 1, taken with permission from Eder et al. [2010]. The device contains a light source and a sensor. Between them there is a 15 cm wide gap. The light source emits light of different wavelengths. The water in the stream running through the gap absorbs a fraction of the emitted light and the sensor records what is left. The physical units used to describe the absorbed fraction of the light are the absorbance A and the absorption coefficient a. The absorbance A is defined as

$$A = \log_{10}(I_0/I), \tag{70}$$

where I_0 is the omitted radiation and I is the radiation measured at the reflector. The variable A is non-dimensional since the units of I_0 and I cancel each other out. The absorption coefficient a is defined as

$$a = \frac{\ln(10)A}{l} \tag{71}$$

where l is the distance between the light source and the sensor in meters (0.15 in our case). Hence the unit of a is m^{-1}. In Eq. 71 the constant $\ln(10)$ has the effect that the ratio is in terms of the natural logarithm. With the absorption coefficient, measurements can be compared more easily between different studies, since the absorption coefficient does not depend on the length of the gap between the light source and the sensor. A more detailed discussion of these physical units is given by Hu et al. [2002].

6.2 Dissolved organic matter

Spectrometric data is used to model dissolved organic matter (DOM). DOM is defined as the fraction of organic matter that passes through a 0.45 nm filter (Perdue and Ritchie [2003]). DOM has a massive influence on various aspects of freshwater eco-systems, for example on microbial communities (Docherty et al. [2006]) or on optical properties of the water (Li and Hur [2017]). From a data analysis point of view, a lot of things have been done to model DOM with spectrometric data. You et al. [1999] use absorption ratios (ratios of absorption coefficients of different wavelengths) to analyse the molecular weight of dissolved organic carbon in soil. In Ikeya and Watanabe [2003] absorbance ratios were used to model the degree of humification in humic acids. For more applications of absorption coefficients and absorption ratios to analyse DOM view Li and Hur [2017].

6.3 Data preparation

In our case, the spectrometer measured the absorption coefficient a, see Eq. 71, every ten minutes with light of different wavelengths in the range of 200 nm to 750 nm. The precise wavelengths used were $\boldsymbol{w} = (200, 202.5, 205, ..., 597.5, 750)$. This yields a realisation of a 221-variate time series $\boldsymbol{x}_t$, summarized in a data matrix $\tilde{\boldsymbol{X}} \in \mathbb{R}^{n\times 221}$. With the help of the vector $\boldsymbol{w}$ containing the used wavelengths, the elements of $\tilde{\boldsymbol{X}}$, $\tilde{X}_{[t,i]}$ can be described as the absorption coefficient measured with wavelength $\boldsymbol{w}_{[i]}$ at time $t($ $t \in T$, $i = 1...221)$. Lag one in the time series means a time interval of 10 minutes between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$. The first measurement was taken on January 14th 2014 at 2:20 PM, the last measurement was on December 31st 2014 at 11:50 PM. This means in theory, the time

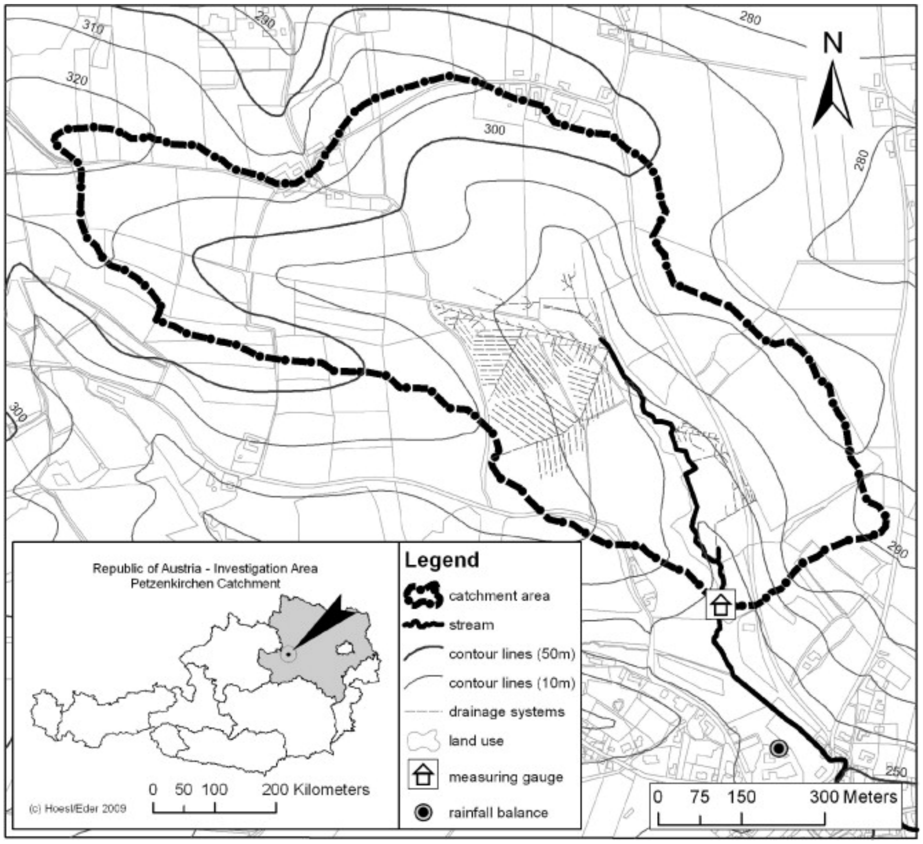

Figure 1: Map of the area near the stream. Reprinted with the permission from Eder and Hoesl, it first appear in their paper: Eder et al. [2010].

index t of the time series $\boldsymbol{x}_t$ runs through $t \in T = \{1, \ldots, 50614\}$. However, due to maintenance of the device and technical errors, the original data set consists of only 44296 observations. Furthermore the original data set has a column encoding the quality of a measurement. Valid measurements were encoded with the level *Ok*. The data set has 42785 valid measurements.

Keeping only observations marked with *Ok*, the data set still has 39 missing values. With these we proceeded as follows: If an observation had more than 3 missing values, we removed the entire observation from the data set. This was the case for one observation, the measurement from May 28th, 12:30 PM. For the other observations, having 3 missing values or less, we replaced the missing value with the mean of the absorption coefficients of the five nearest wavelengths. Missing values occurred at wavelengths 200, 202.5 and 205 nm. The missing value for wavelength 200 for example was replaced with the mean of the absorption coefficients of the wavelengths 202.5 to 212.5.

Values less or equal than 0 required some attention too. Looking at Eq. 71, a negative or 0 value of a means that the the output radiation I is bigger or equal than the input radiation I_0. That makes no sense, so these had to be treated as measurement errors as well. Also our methodology requires the data to have strictly positive values. In reconciliation with Matthias Pucher, we used only the variables from the wavelengths 200 to 600 nm for our analysis because for wavelengths 602.5 to 750 nm the absorption coefficient is very small which might skew the results. Also due to inaccuracy of the measurement device there are 70 values less or equal than zero in that wavelength range. In the wavelength range from 200 to 600 all values are strictly positive.

Having removed all variables above 600 nm and the one observation with 4 missing values we arrived at our cleaned time series data matrix $\boldsymbol{X}$, with 42784 rows and 161 columns .

6.4 Visualisation of the data

Figure 2 shows the absorption spectrum of the first observation. In the ultraviolet to visible (UV-Vis) part of the spectrum (about 200 nm to 700 nm) the absorption coefficients for DOM decrease approximately exponential with increasing frequency (see for example Massicotte and Markager [2016]). Figure 3 shows the absorption spectra of 2000 about equidistant observations, where clearly some outliers to this exponential trend can be seen. In fact, deviations from this exponential trend is caused by DOM in the stream. The thick black curve is the mean over all observations in the data set. Figure 4 shows the time series for the wavelength 200 nm. Gaps in the graph represent missing values in the data set for that time.

6.5 Dimension reduction via PCA

In our case the ilr-data matrix $\boldsymbol{X}^{ilr}$ turns out to contain very high linear correlations. The determinant of the estimated variance covariance matrix is zero. This suggests that we are in the situation of highly correlated compositional time series. Therefore, first PCA is performed on the ilr-data matrix and only the first few principal components are used as the observable time series in the BSS models.

Definition 32 (Ilr- and Clr- data matrix)
Let $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ be the data matrix representing realisations of the time series $\boldsymbol{x}_t$. Applying the ilr-tranformation from Def. 5 on the rows of the data matrix $\boldsymbol{X}_{[t,.]}$ yields the ilr-transformed data matrix $\boldsymbol{X}^{ilr}$

$$\boldsymbol{X}^{ilr}_{[t,.]} = ilr(\boldsymbol{X}_{[t,.]})$$
$$\boldsymbol{X}^{ilr} = [\boldsymbol{X}^{ilr}_{[1,.]}, \dots, \boldsymbol{X}^{ilr}_{[n,.]}]^\top.$$

In a similar fashion the clr- data matrix is obtained:

$$\boldsymbol{X}^{clr}_{[t,.]} = clr(\boldsymbol{X}_{[t,.]})$$
$$\boldsymbol{X}^{clr} = [\boldsymbol{X}^{clr}_{[1,.]}, \dots, \boldsymbol{X}^{clr}_{[n,.]}]^\top.$$

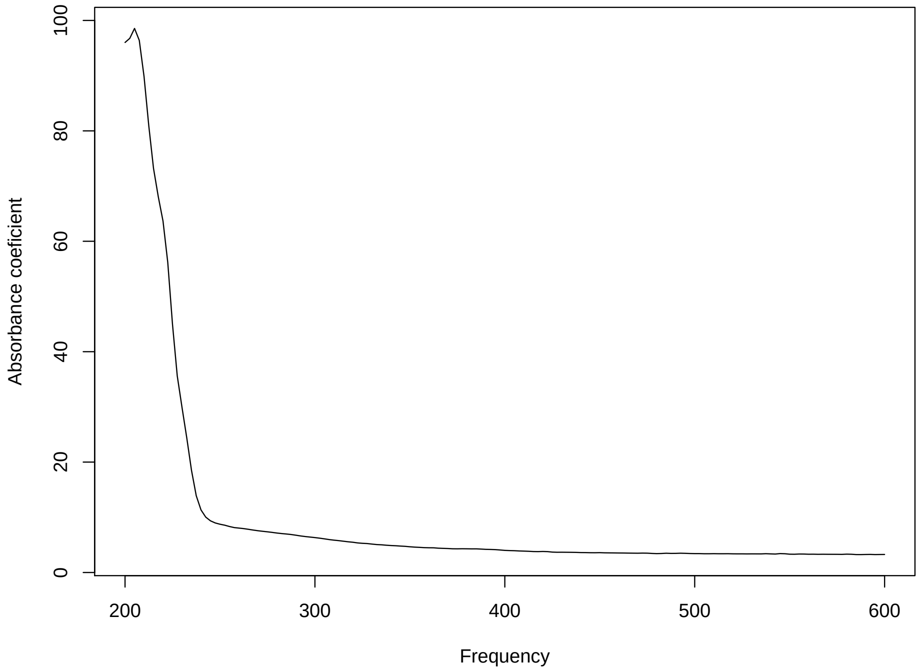

Figure 2: Absorption coefficients of the first observation.

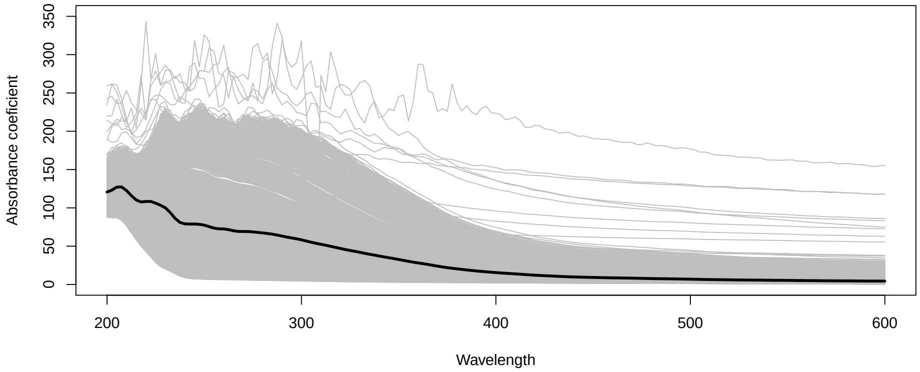

Figure 3: Absorption coefficients for 2000 observations, thick black curve represents mean of all observations.

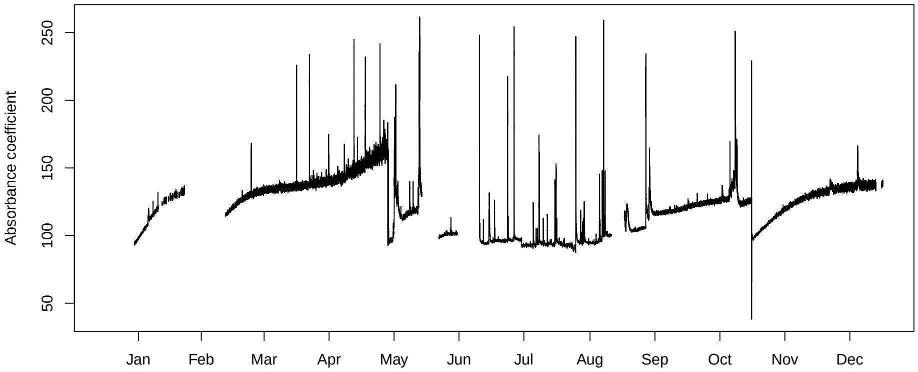

Figure 4: Time series of a for wavelength 200 nm.

We performed both a classical and a robust principal component analysis. In the classical PCA we used the classical estimator for the variance covariance matrix of the ilr- data matrix, see Def. 9. And in the robust one we used the robust minimum covariance determinant as the scatter estimator, see Def. 11.

The clr-loadings of the first four classically estimated principal components can be seen in Fig. 5. The clr-loadings of the first four robust estimated components are shown in Fig. 7. The loadings of these two different principal component analysis look very similar, indicating no significant outliers in the data. Looking at the clr-loadings of the first principal component indicates that the highest variability in the data lies in the relative value between the components from wavelengths 230 to 350 and 500 to 600. The signs of the loadings 230 to 350 being the same for instance means, that absorption coefficients of wavelengths 230 to 350 tend to be either all high or all low. The clr-loading of the second principal component shows that there is also some variability in the absorption coefficients of the wavelengths 200 to 220. Figure 6 shows the eigenvalues of the classical estimated covariance matrix, Fig. 8 shows the eigenvalues of the robust estimated covariance matrix. These plots show that quite a big dimension reduction can be done on the data. In the classical PCA the first four principal components explained 99.96 % of the variance, and in the robust PCA 99.91 % of the variance is explained by the the first four principal components. Therefore we only used the first four principal components as the observable time series in the blind source separation models.

6.6 Adjusting the estimators of $\boldsymbol{S}_\tau$ and $\boldsymbol{B}_\tau$ in the BSS models

One thing we have to adjust to apply BSS for CTS to our data are the estimators of the autocovariance matrix at lag τ, $\boldsymbol{S}_\tau$, and the fourth order cross moment matrix $\boldsymbol{B}_\tau$. Looking at the estimator of $\boldsymbol{S}_\tau$ in Def. 34 for instance, this estimator assumes that for every observation $\boldsymbol{X}_{[t,.]}$ in the data matrix, the observation τ rows down in the data matrix, $\boldsymbol{X}_{[t+\tau,.]}$, is indeed a measurement taken τ lags

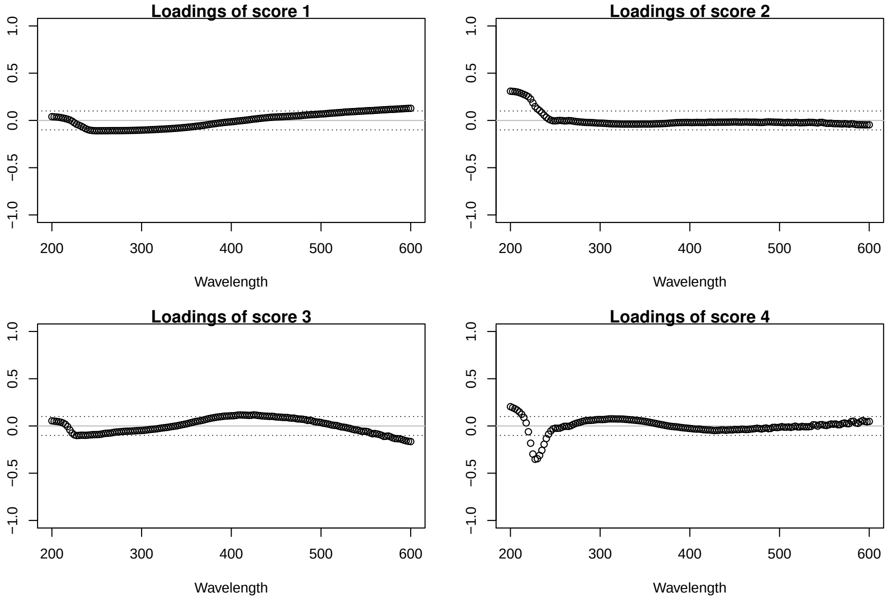

Figure 5: Loadings of the first four classically estimated scores.

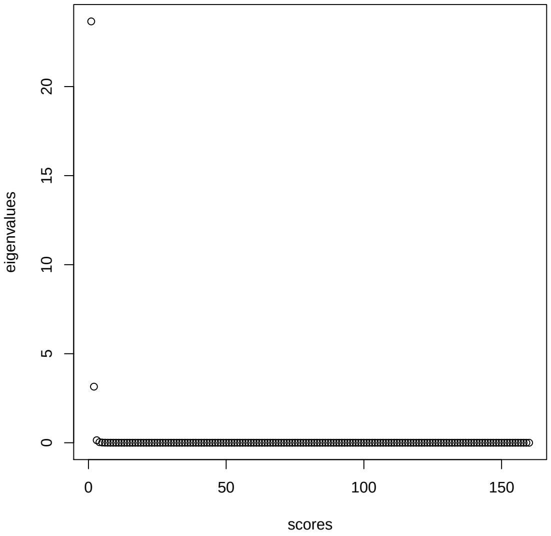

Figure 6: Eigenvalues of the classically estimated covariance matrix.

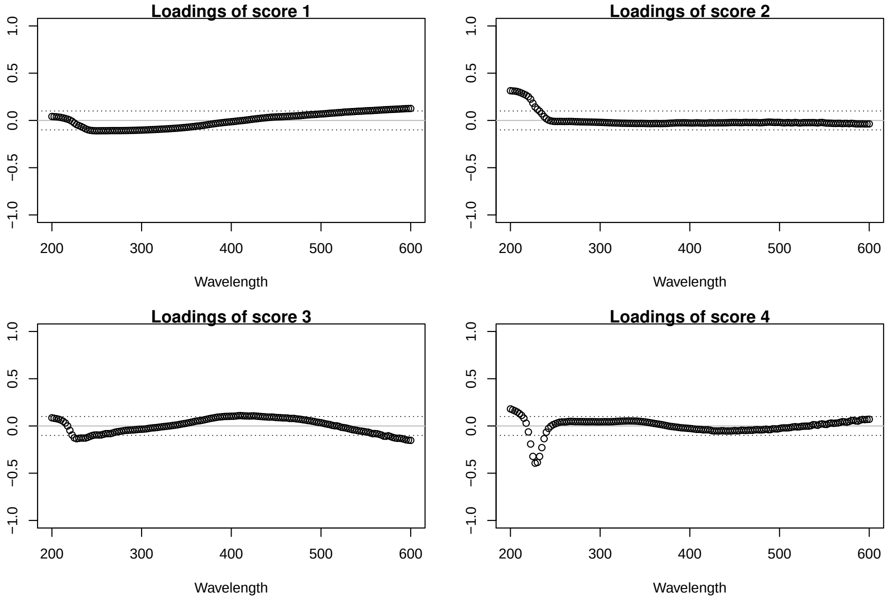

Figure 7: Loadings of the first four robust estimated scores.

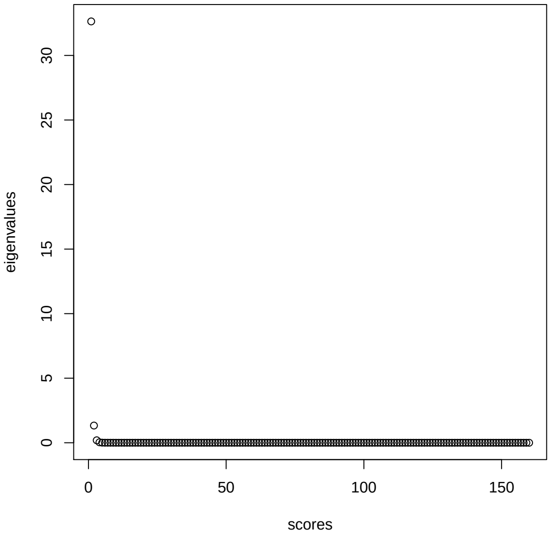

Figure 8: Eigenvalues of the MCD.

later. Due to missing values and measurement errors, this is not always the case in our data matrix. Therefore we have to make sure that the two rows $\boldsymbol{X}_{[t,.]}$ and $\boldsymbol{X}_{[t+\tau,.]}$ in the sum

$$\boldsymbol{A}_\tau = \frac{1}{n-\tau} \sum_{t=1}^{n-\tau} \boldsymbol{X}_{[t,.]} \boldsymbol{X}_{[t+\tau,.]}^\top \tag{72}$$

are indeed measurements with a time difference of lag τ. Each row $\boldsymbol{X}_{[t,.]}$ in the data matrix has a time stamp θ_t in the form of "14.01.2014 14:20" attached to it. We designed a function that takes two such time stamps as an input, and outputs the time difference in terms of lags between those two measurements.

Function 1 (Lag-function)
The lag-function f_l takes two time stamps θ_{t_1} and θ_{t_2} as an input and outputs the time difference in terms of the lag τ at which the corresponding observations are apart, keeping in mind that a lag of $\tau = 1$ means a time difference of ten minutes between two measurements:

$$f_l(\theta_{t_1}, \theta_{t_2}) = \tau. \tag{73}$$

For instance,

$$f_l(\text{“14.01.2014 14:20”}, \text{“14.01.2014 14:50”}) = 3. \tag{74}$$

Using this lag-function we calculated lag matrices $\boldsymbol{L}_\tau$, containing the information of all possible pairs of row-indices in the data matrix that are indeed measurements with a time difference of lag τ:

Definition 33 (Lag-matrix $\boldsymbol{L}_\tau$)
Let $\boldsymbol{L}_\tau \in \mathbb{R}^{n(\tau)\times 2}$ be the matrix whose rows $\boldsymbol{L}_{\tau[i,.]} \in \mathbb{R}^2$ contain all possible $n(\tau)$ pairs of rows in the data matrix $\boldsymbol{X}$ being a time difference of lag τ apart.

Using this lag matrix, the adjusted estimator of the auto-covariance matrix at lag τ, $\boldsymbol{S}_\tau$, can be calculated as

Definition 34 (Adjusted symmetrisized estimator of $\boldsymbol{S}_\tau$)

$$\boldsymbol{A}_\tau = \frac{1}{n(\tau)} \sum_{i=1}^{n(\tau)} \boldsymbol{X}_{[L_{\tau[i,1]},.]} \boldsymbol{X}_{[L_{\tau[i,2]},.]}^\top \tag{75}$$

$$\hat{\boldsymbol{S}_\tau} = \frac{1}{2}\left(\boldsymbol{A}_\tau + \boldsymbol{A}_\tau^\top\right). \tag{76}$$

The estimators for the fourth order cross moment matrix (Def. 25) and the spacial sign auto-covariance matrix (Def. 20) were adjusted in an analogue fashion.

6.7 Fitting the BSS models

Since both in the standard and the robust case the first four principal components explain around 99.9% of the variance, we use only the first four principal components as the observable time series in the BSS models. The robust principal components were used in combination with the robust SOBI-method. For all other BSS methods the classically obtained principal components were used. We apply the procedure, explained in subsection 5.1. This yields a clr-unmixing matrix $\tilde{\mathbf{\Lambda}} \in \mathbb{R}^{4\times161}$ and estimated sources $\tilde{\boldsymbol{z}}_t \in \mathbb{R}^4$ fulfilling

$$\tilde{\boldsymbol{z}}_t = \tilde{\mathbf{\Lambda}}(\boldsymbol{x}_t^{clr} - \hat{\boldsymbol{\mu}}_{clr}), \tag{77}$$

see Eq. 69. In this procedure, one has the choice which BSS method to apply on the scores. We applied the SOBI-method with the lag sets T_1 and T_2:

$$\begin{aligned} T_1 &= \{6, 12, \dots, 144\} \\ T_2 &= \{72, 144, \dots, 1008\}. \end{aligned} \tag{78}$$

A lag of 6 represents a difference of one hour between two measurements and a lag of 72 represents a difference of half a day between two measurements. Therefore T_1 represents a set of lags of 1 hour, 2 hours, ..., 24 hours. T_2 represents a set of lags of half a day, a full day, 36 hours and so on, up to one week.

Furthermore we applied the robust SOBI method and the gFOBI method, once for each lag set T_1 and T_2. We also applied the novel NSS-SOBI-gFOBI-method. This method requires two lag sets, lag set L_1 for the autocovariance matrices and lag set L_2 for the fourth order cross-moment matrices, see BSS method 6. We used T_1 for lag set L_1 and T_2 for lag set L_2 (Eq. 78). In this method also weights α_j have to be chosen. We fitted three models for this method: once with $\alpha_j \equiv 1$, and once each with the weights α_j described in Eq. 52 and 53. Additionally we fitted the models with the NSS-SOBI-gFOBI method with $L_1 = L_2 = T_1$, also one each with $\alpha_j \equiv 1$ and α_j from Eq. 52 and 53.

When comparing the estimated loadings of the different models, one has to take into account the possible permutations and the sign-changes of the sources. A sign-change of a source would correspond to mirroring the loadings around the x-axis. Figure 9 shows the loadings of the sources obtained by the SOBI method with lag set $T_1 = \{6, 12, ..., 144\}$, while the loadings obtained by the SOBI method with lag set $T_2 = \{72, 144, ..., 1008\}$ are shown in Fig. 11. Comparing these two figures, one can see that in both methods the estimated loadings for the sources 1 and 2 are almost identical. Also the loadings of source 3 from the method with lag-set T_1 look very similar to the loadings of source 4 in the method with lag-set T_2, which showcases that the sources are only obtainable up to permutations. We continue to compare the loadings of source 4 from the method with lag-set T_1 with the loadings from source 3 from the method with lag-set T_2. Interestingly, there is a small difference notable: In the method with lag set T_1 the loadings for wavelengths 200 to 300 are zero, while in the other method there are some influential negative loadings around wavelength 220.

Figures 13 and 15 show the loadings acquired by the robust SOBI-method with lag set $T_1 = \{6, 12, ..., 144\}$, and $T_2 = \{72, 144, ..., 1008\}$ respectively. In the robust case the loadings of the sources look exactly the same, when taking into account that the sources 3 and 4 switch places. That showcases that in this specific data set the robust method is a little less dependent on the chosen lag set. Overall the similarity between the loadings obtained by the robust method and the ones obtained by the classical method indicate few outliers in the data.

The loadings of the sources obtained by the gFOBI method with the lag-sets T_1 and T_2 respectively are displayed in Figures 17 and 19. Also in the gFOBI case the choice of the lag-set has no huge impact on the output for our concrete data set. The biggest difference can be seen between the loadings for source 2 when lag-set T_1 was used and the loadings for source 3, when lag set T_2 was used. In the output from lag-set T_2 the wavelengths around 220 are not as important for the source 3 as they are for source 2 in the output from lag-set T_1. On the other side, in the method with lag-set T_2 wavelengths from 590 to 600 nm seem to be important for source 3, while for the corresponding source 2 in the output with lag set T_1, the loadings for that wavelength range are close to zero indicating no influence there.

In Figures 21, 23 and 25 the loadings for the sources obtained from the NSS-SOBI-gFOBI-method with lag set $T_1 = \{6, 12, \ldots, 144\}$ for the auto-covariance matrices and lag set $T_2 = \{72, 144, \ldots, 1008\}$ for the fourth-order cross moment matrices are shown for each choice of weights α_j. Comparing the loadings from each choice of α_j shows some influence of the weights on the output, although also here a lot of similarity is present. Interestingly the loadings obtained by $\alpha_j \equiv 1$ and by

$$\begin{aligned} \alpha_j &= \frac{1}{p+2} \quad \text{if } \boldsymbol{V}_{T_i,j} \text{ is a fourth order cross-moment matrix} \\ \alpha_j &= 1 \quad \text{otherwise} \end{aligned} , \tag{79}$$

are almost identical, while the choice of α_j from

$$\begin{aligned} \alpha_j &= \frac{1}{\max|\boldsymbol{V}(T_i, j)|} \quad \text{if } \max|\boldsymbol{V}(T_i, j)| > 1 \\ \alpha_j &= 1 \quad \text{otherwise} \end{aligned} \tag{80}$$

yields different loadings for source 1. Note that the loadings with the choice of α_j from Eq. 80 have changed their sign compared to the loadings from the other choices of α_j. In the output from α_j from Eq. 80 the loadings for source 1 look more similar to the loadings for source 3. Source 1 seems to focus on the ratio between absorption coefficients in the range of 200 to 210 and absorption coefficients in the range from 210 to 220. Source 3 also takes absorption ratios from wavelengths around 300 and 400 into account, which is not the case for source 1.

Figures 27, 29 and 31 show the loadings acquired by the NSS-SOBI-gFOBI-method with the lag sets used for the auto-covariance matrices and the fourth order cross-moment matrices being identical ($L_1 = L_2 = \{6, 12, \ldots, 144\}$), for all three choices of weights respectively. Here a similar picture as in the case with different lag-sets can be seen. That indicates, that the choice of the lag set does not have a huge impact on the application to this data set.

Overall the outputs from all methods are very similar which is an indication for relevant latent sources containing features of all BSS models. Hence all methods were able to recover them. All methods show that the absorption coefficients in the wavelength range 200 to 300 are the most important ones for constructing the estimated sources.

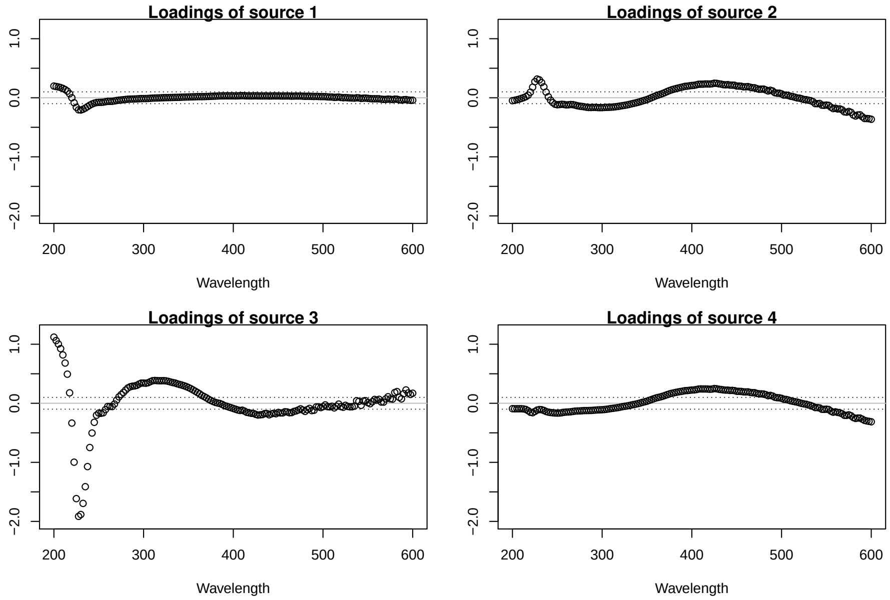

Figure 9: Loadings in clr-space of the sources from the classical SOBI with lag set $T_1 = \{6, 12, ..., 144\}$.

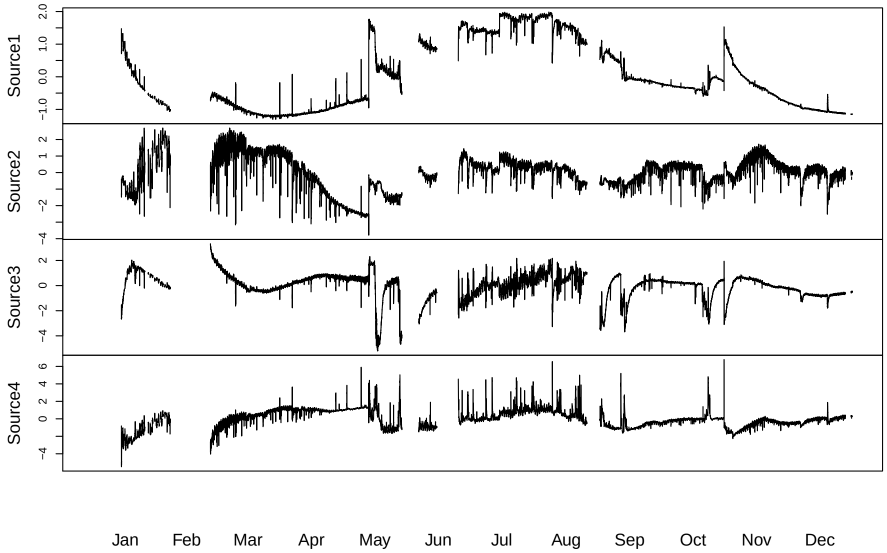

Figure 10: Sources of the classical SOBI model with lag set $T_1 = \{6, 12, ..., 144\}$.

6.8 Interpretation of loadings and sources

In this section we are going to show how the loadings in clr-space can be interpreted, though we leave a more detailed interpretation to the experts on the field. Both the loadings and the sources of the models can be used for an interpretation of these sources. It is remarkable that the loadings of the sources are very similar throughout the different models. Therefore we are just going to pick one output from a model and show how these sources can be interpreted. One thing that stands out, from the sources point of view, is that one source has a seasonal high. For instance, in all models of the NSS-SOBI-gFOBI-method, source 1 has higher values in the summer period than in the rest of the year. Throughout the rest of the year the source stays more or less constant.

We interpret the clr-loadings with the help of the paper from Li and Hur [2017]. This paper gives an overview on what has been done to detect DOM using absorption coefficients. Ratios of absorption coefficients have been used to characterize the quality of DOM, like molecular weight, humification degree or aromacity [Li and Hur, 2017]. For instance the absorption ratio $\frac{a_{250}}{a_{365}}$ has been used to characterize the molecular weight of DOM [Peuravuori and Pihlaja, 1997].

As discussed in the section about compositional data, the log-ratios lie within the linear space of the clr-transformed components. The log-ratio between the i-th and the j-th component can be obtained via the difference of the i-th and the j-th clr-coordinate, since

$$\begin{aligned} \log\left(\frac{x_i}{(\prod_{l=1}^{D} x_l)^{\frac{1}{D}}}\right) - \log\left(\frac{x_j}{(\prod_{l=1}^{D} x_l)^{\frac{1}{D}}}\right) = \\ \log\left(\frac{x_i(\prod_{l=1}^{D} x_l)^{\frac{1}{D}}}{x_j(\prod_{l=1}^{D} x_l)^{\frac{1}{D}}}\right) = \log\left(\frac{x_i}{x_j}\right) \end{aligned} \tag{81}$$

Using this fact, a symmetrical loading around 0 of the clr-loadings for the wavelengths 250 and 365 can be interpreted as a loading for molecular weight since

$$\alpha x_{21}^{clr} - \alpha x_{67}^{clr} = \alpha \log\left(\frac{a_{250}}{a_{365}}\right). \tag{82}$$

The numbers 21 and 67 in Eq. 82 are the indices in $\boldsymbol{w}$ for the wavelengths 250 and 365. Such a symmetric distribution can be seen in the loadings for source four in the BSS model with the NSS-SOBI-gFOBI-method with α_j from Eq. 53, see Fig. 33. Therefore the source 4 in that model can be interpreted as a source that characterizes the molecular weight of DOM in the stream. According to Peuravuori and Pihlaja [1997] the ratio $\log(\frac{a_{250}}{a_{365}})$ is negatively correlated with the molecular weight. In loading four of the model, the clr-loading for wavelength 365 is positive and the one of wavelength 250 is negative. That means a positive value of source 4 indicates DOM with high molecular weight in the stream and a negative value of that source means that the DOM in the

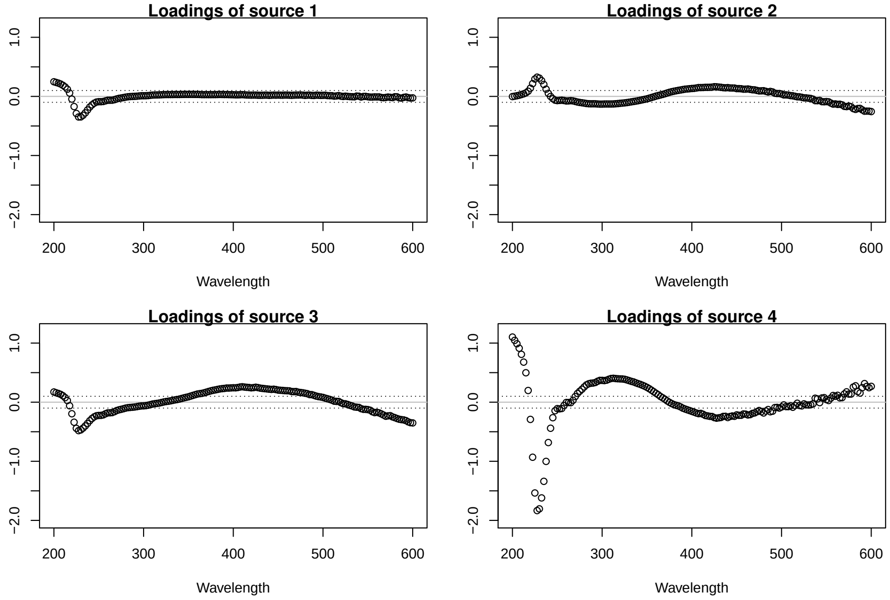

Figure 11: Loadings in clr-space of the sources from the classical SOBI with lag set $T_2 = \{72, 144, ..., 1008\}$.

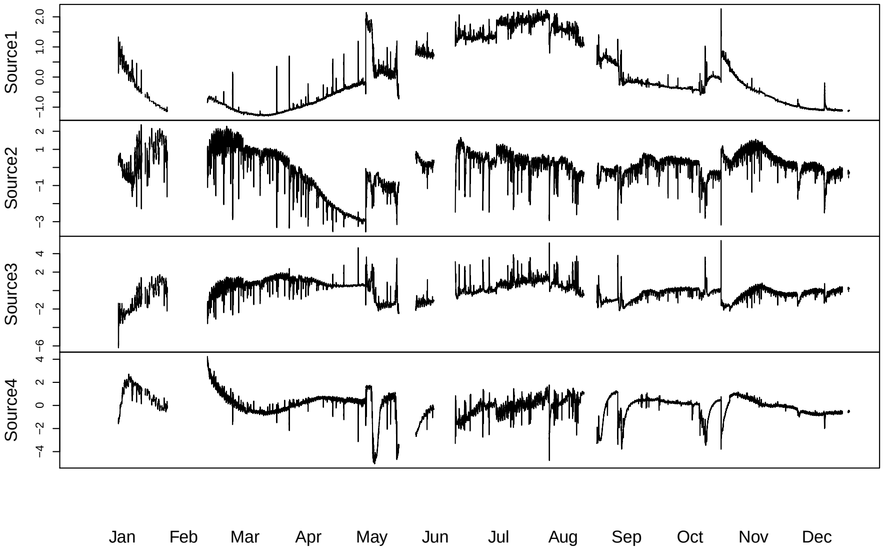

Figure 12: Sources of the classical SOBI model with lag set $T_2 = \{72, 144, ..., 1008\}$.

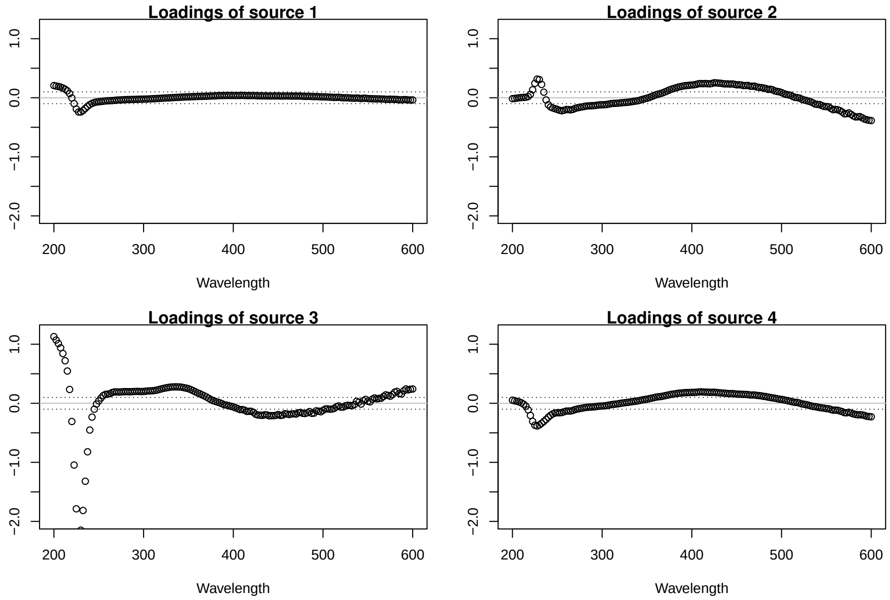

Figure 13: Loadings in clr-space of the sources from the robust SOBI with lag set $T_1 = \{6, 12, ..., 144\}$.

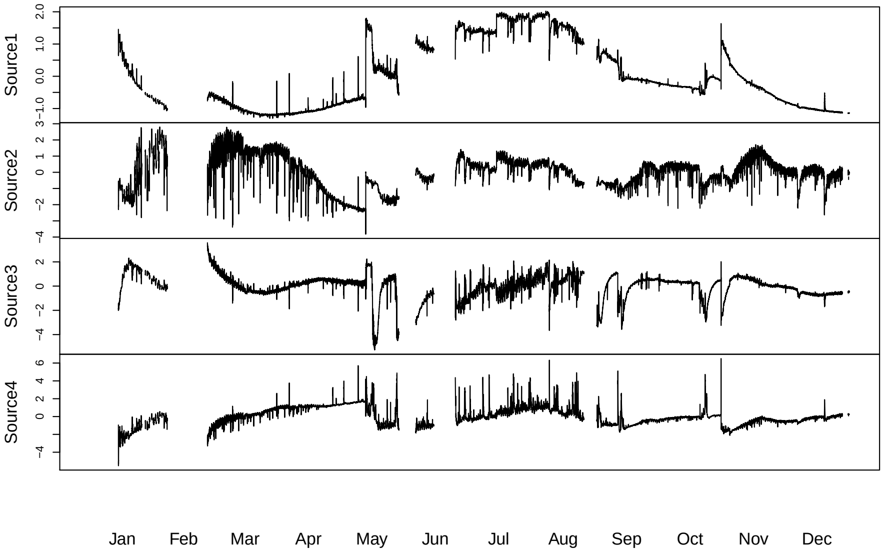

Figure 14: Sources of the robust SOBI model with lag set $T_1 = \{6, 12, ..., 144\}$.

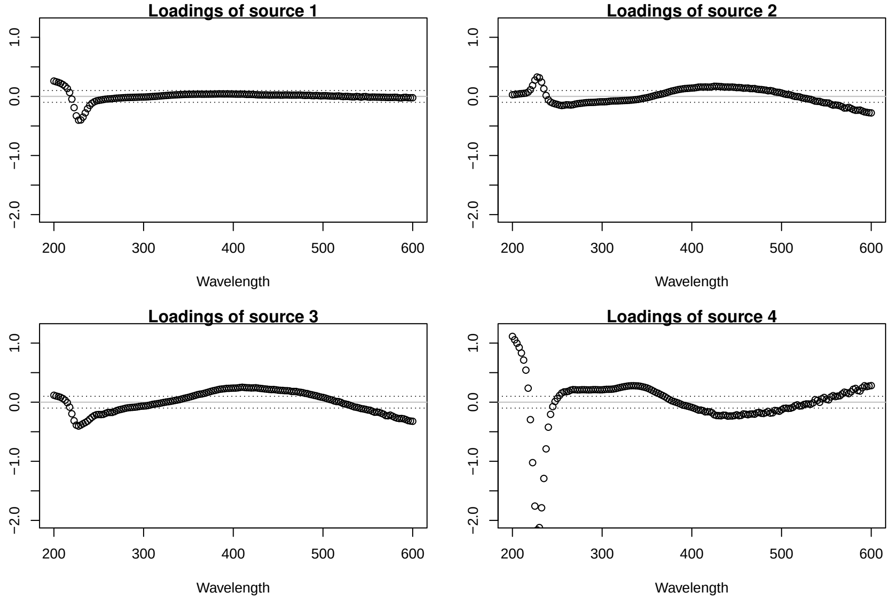

Figure 15: Loadings in clr-space of the sources from the robust SOBI with lag set $T_2 = \{72, 144, ..., 1008\}$.

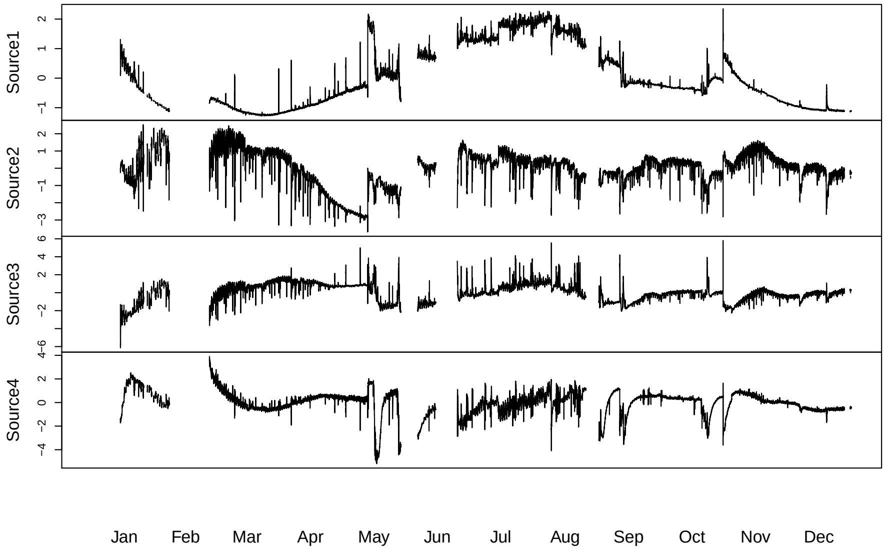

Figure 16: Sources of the robust SOBI model with lag set $T_2 = \{72, 144, ..., 1008\}$.

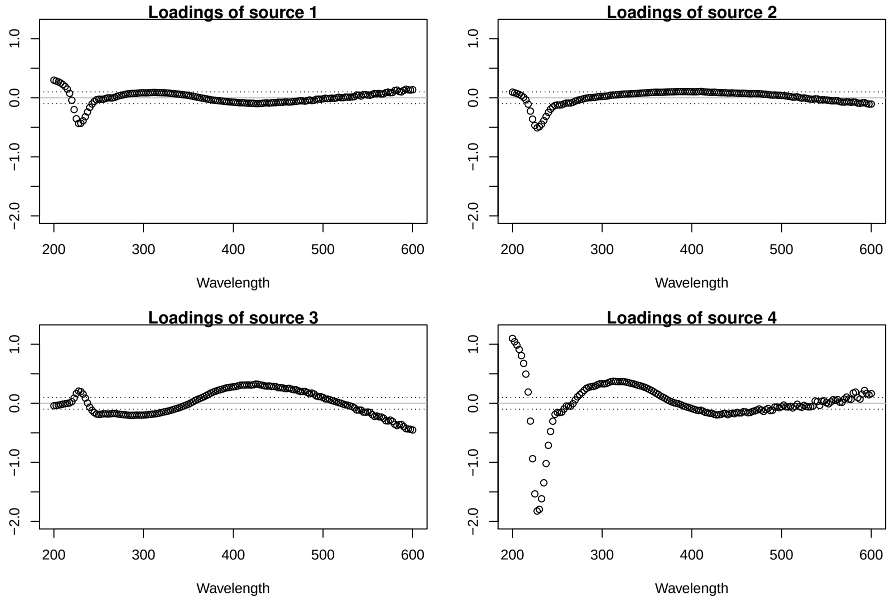

Figure 17: Loadings in clr-space of the sources from the gFOBI model with lag set $T_1 = \{6, 12, ..., 144\}$.

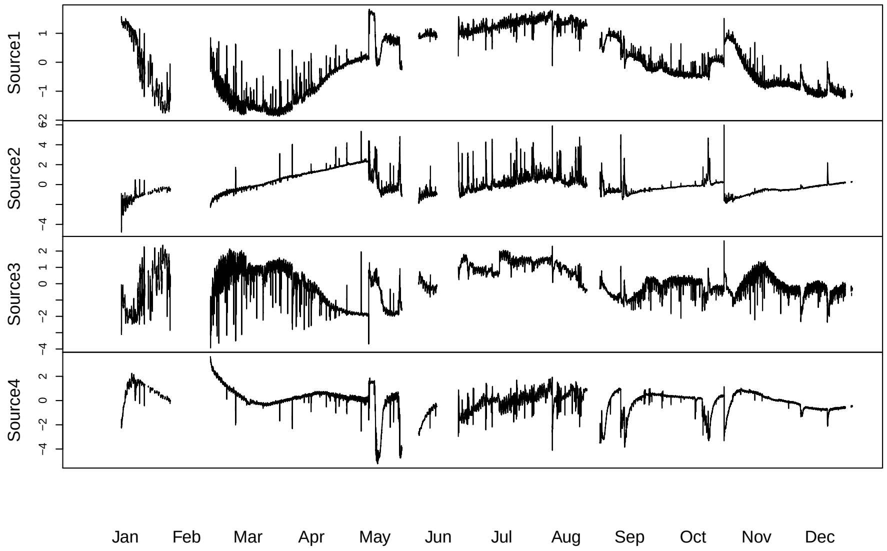

Figure 18: Sources of the gFOBI model with lag set $T_1 = \{6, 12, ..., 144\}$.

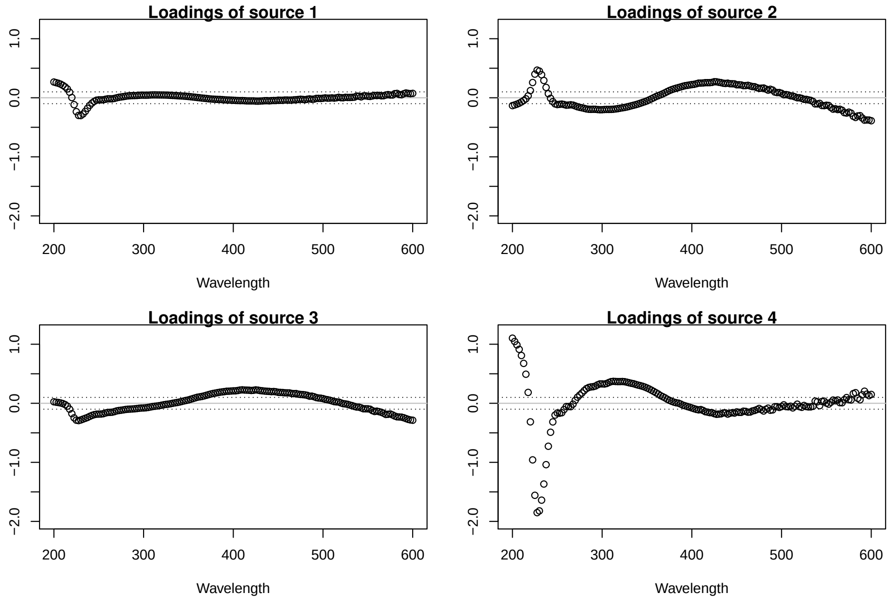

Figure 19: Loadings in clr-space of the sources from the gFOBI model with lag set $T_2 = \{72, 144, ..., 1008\}$.

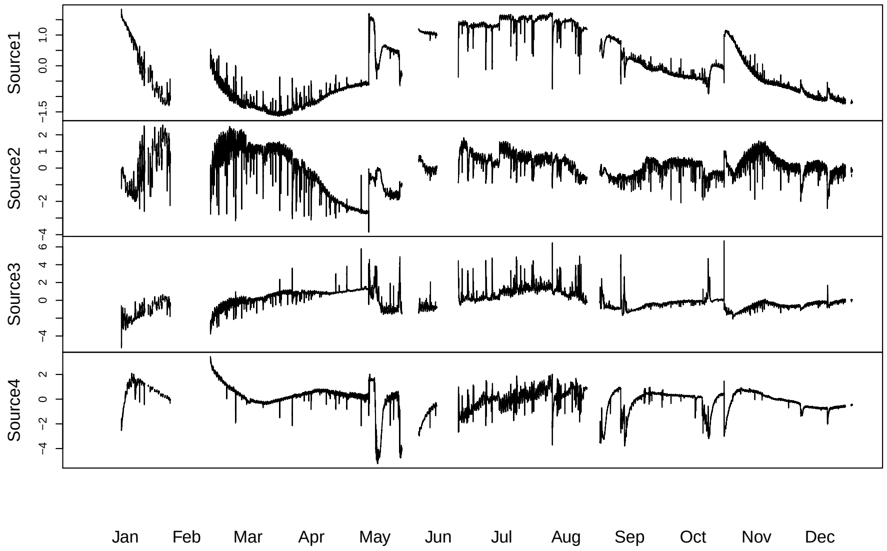

Figure 20: Sources of the gFOBI model with lag set $T_2 = \{72, 144, ..., 1008\}$.

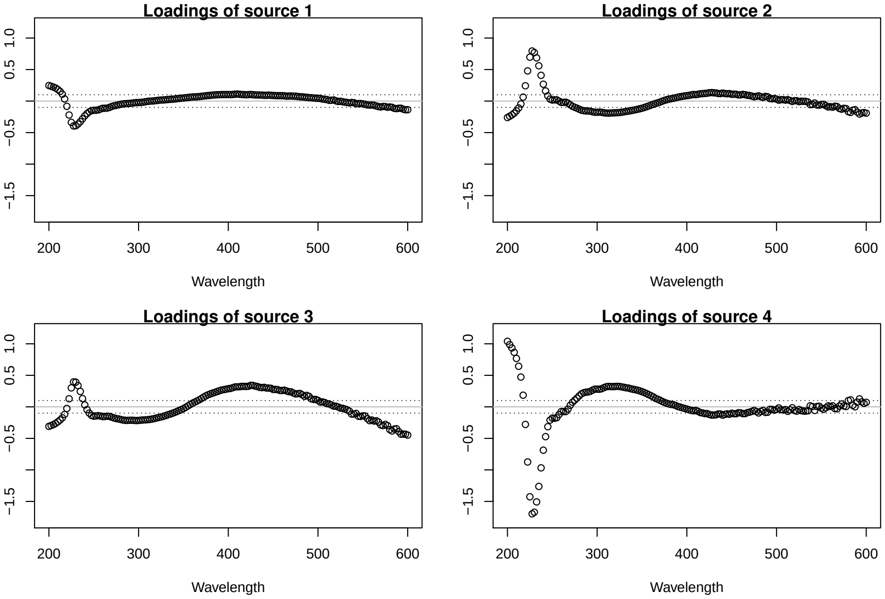

Figure 21: Loadings in clr-space of the sources from the NSS-SOBI-gFOBI-method with $\alpha_j \equiv 1$.

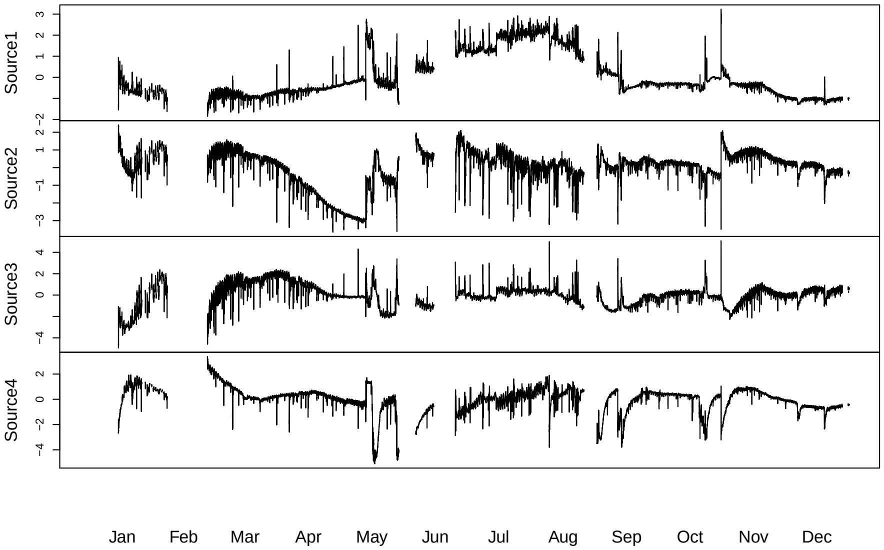

Figure 22: Sources of the NSS-SOBI-gFOBI-method with $\alpha_j \equiv 1$.

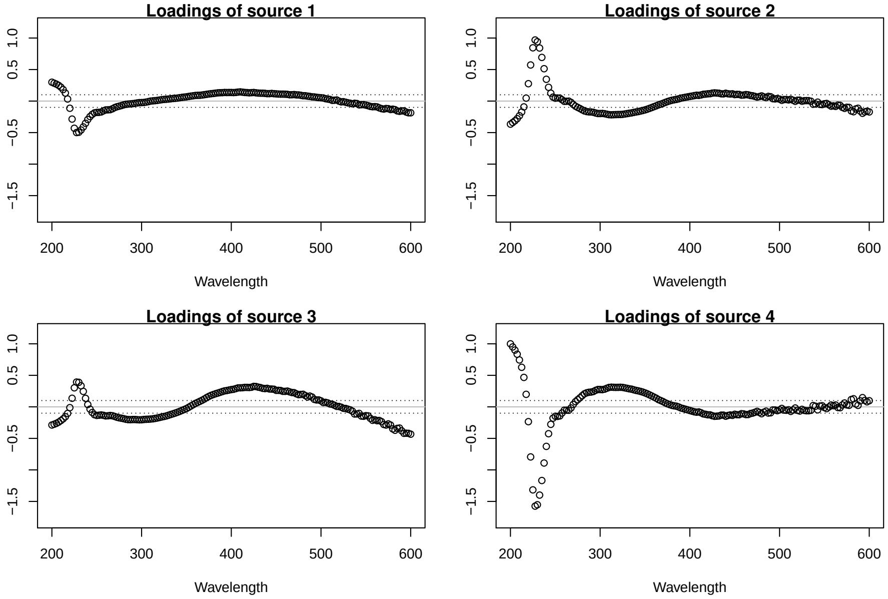

Figure 23: Loadings in clr-space of the sources from the NSS-SOBI-gFOBI-method with α_j from Eq. 52.

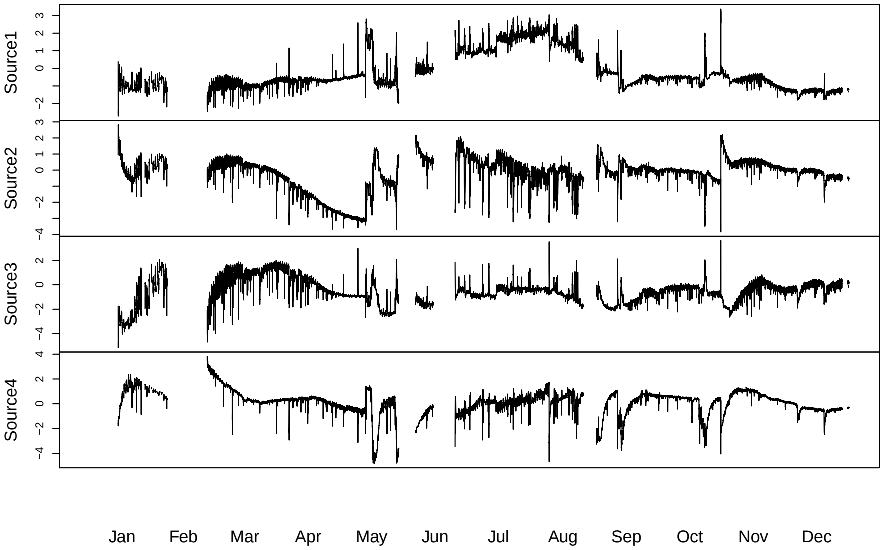

Figure 24: Sources of the NSS-SOBI-gFOBI-method with α_j from Eq. 52.

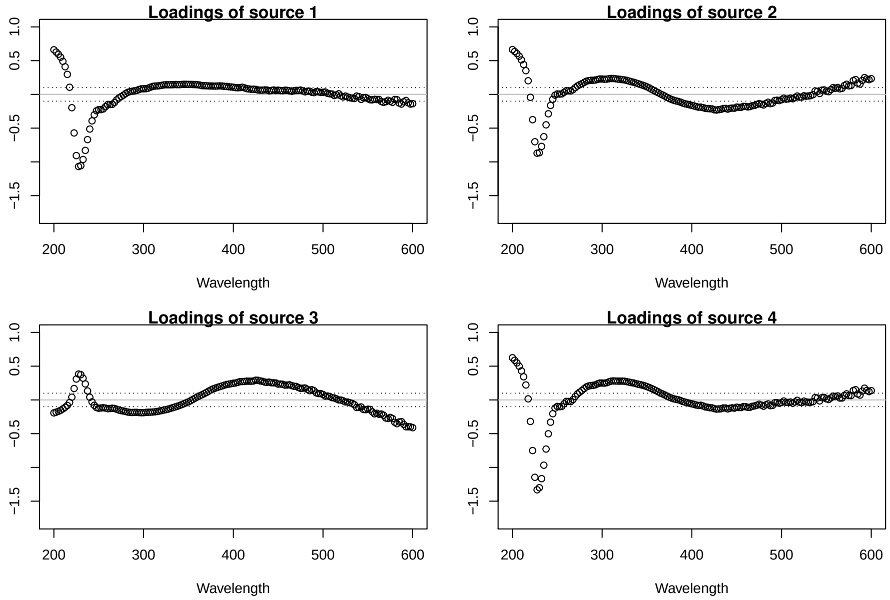

Figure 25: Loadings in clr-space of the sources from the NSS-SOBI-gFOBI-method with α_j from Eq. 53.

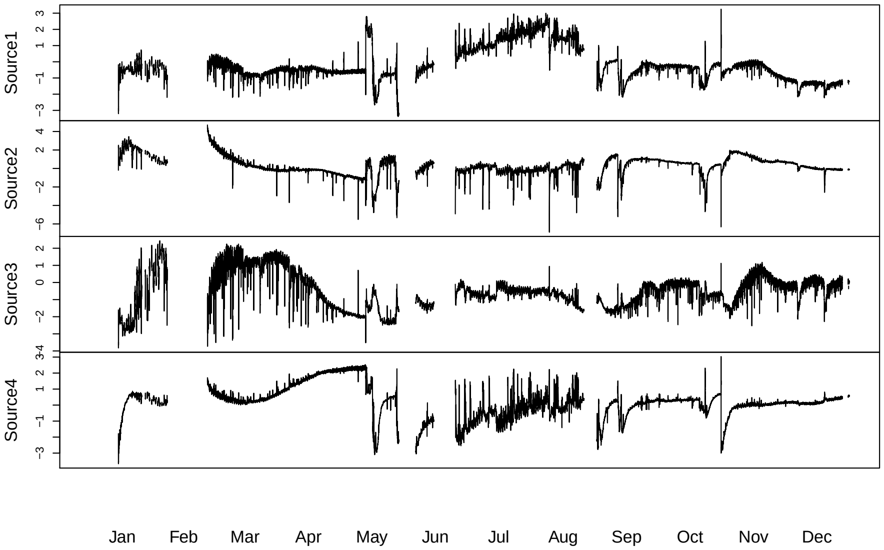

Figure 26: Sources of the NSS-SOBI-gFOBI-method with α_j from Eq. 53.

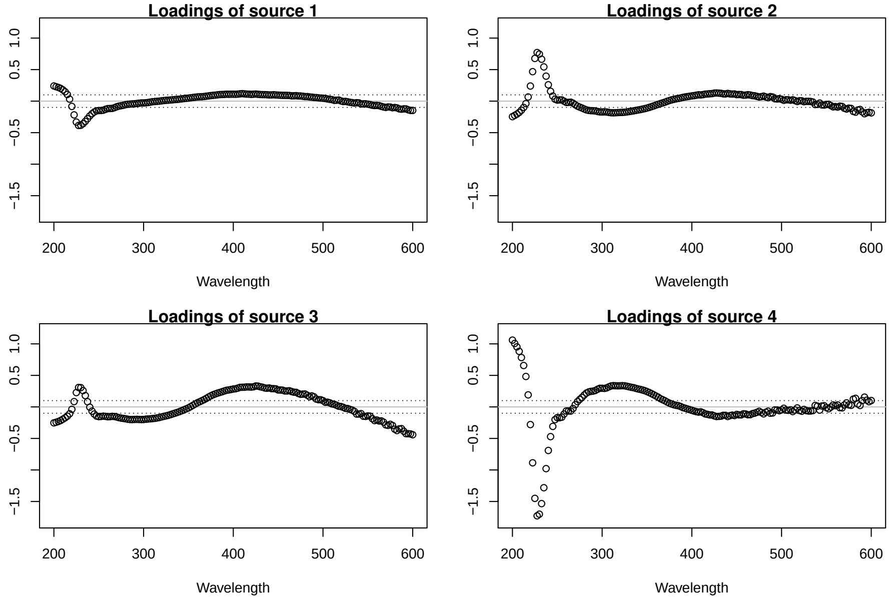

Figure 27: Loadings in clr-space of the sources from the NSS-SOBI-gFOBI-method with $L_1 = L_2 = \{6, 12, ..., 144\}$ and $\alpha_j \equiv 1$.

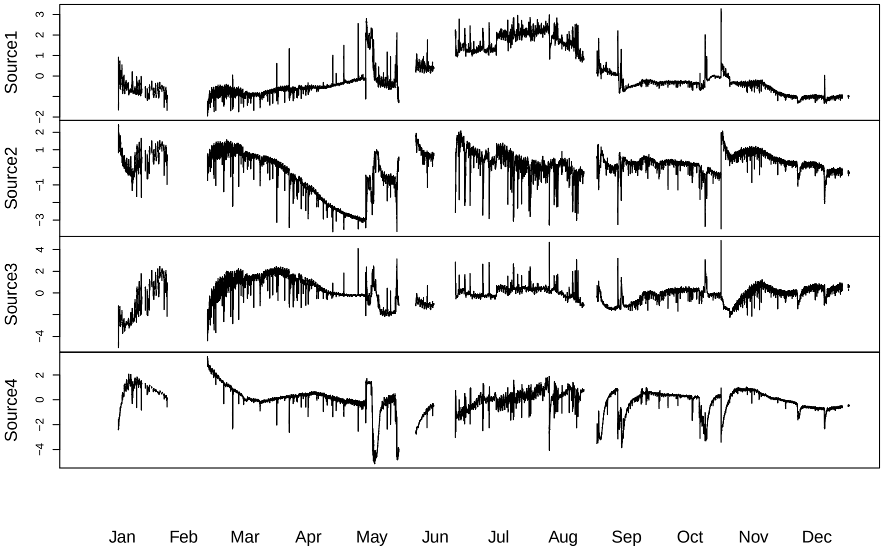

Figure 28: Sources of the NSS-SOBI-gFOBI-method with $L_1 = L_2 = \{6, 12, ..., 144\}$ and $\alpha_j \equiv 1$.

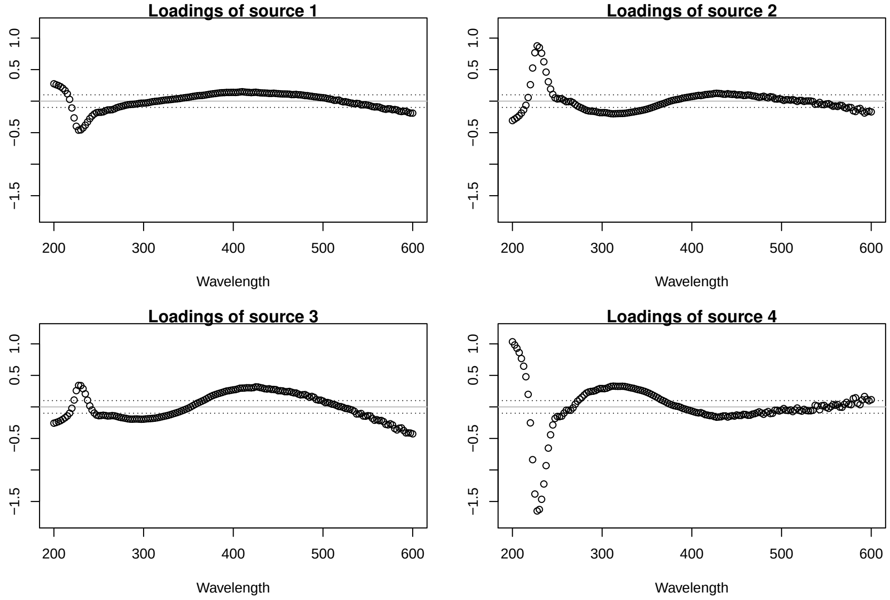

Figure 29: Loadings in clr-space of the sources from the NSS-SOBI-gFOBI-method with $L_1 = L_2 = \{6, 12, ..., 144\}$ and α_j from Eq. 52.

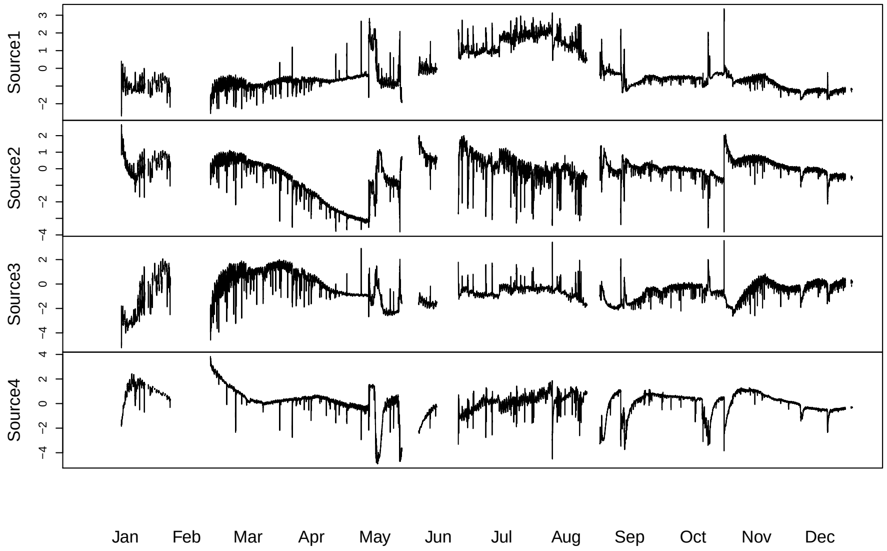

Figure 30: Sources of the NSS-SOBI-gFOBI-method with $L_1 = L_2 = \{6, 12, ..., 144\}$ and α_j from Eq. 52.

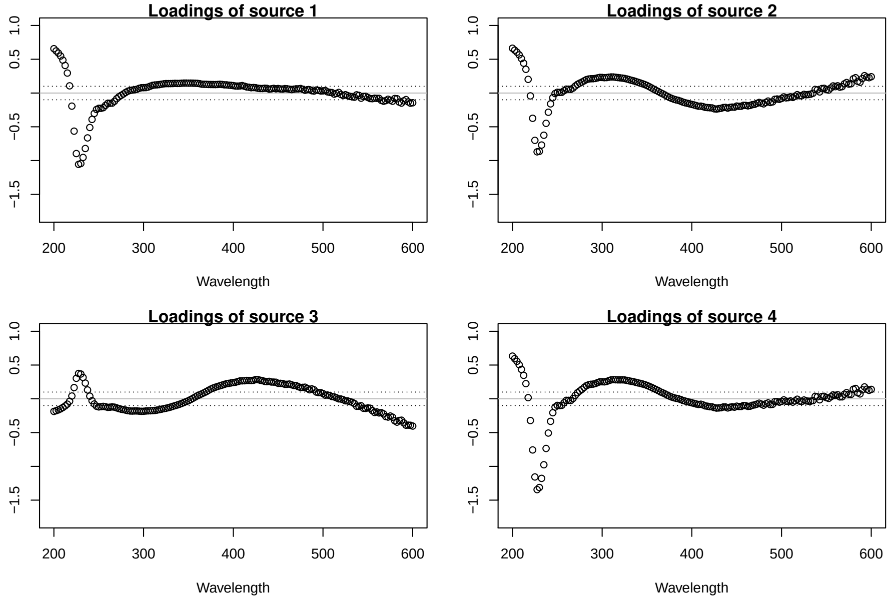

Figure 31: Loadings in clr-space of the sources from the NSS-SOBI-gFOBI-method with $L_1 = L_2 = \{6, 12, ..., 144\}$ and α_j from Eq. 53.

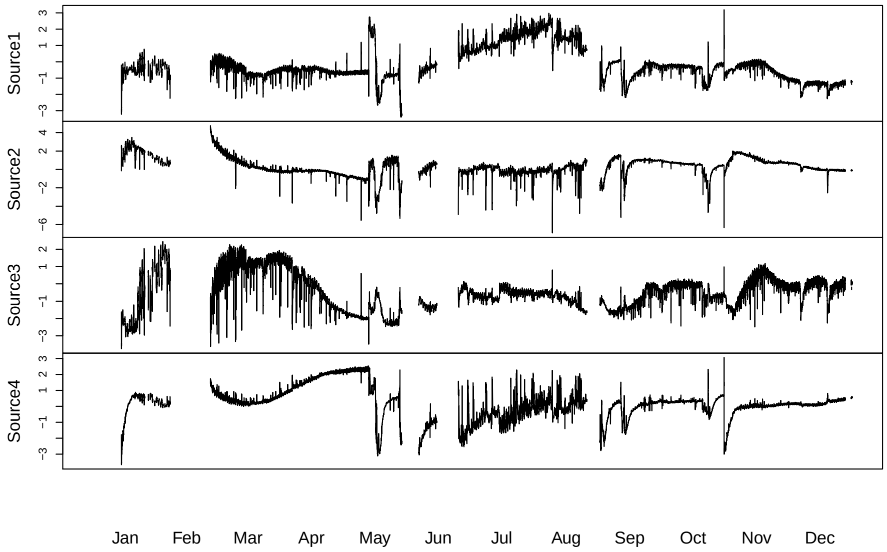

Figure 32: Sources of the NSS-SOBI-gFOBI-method with $L_1 = L_2 = \{6, 12, ..., 144\}$ and α_j from Eq. 53.

stream has rather little molecular weight. Looking at the path of source 4 in Fig. 25, in the spring period the stream tends to contain DOM of higher molecular weight. Also there are interesting peaks downwards in May, June, September and November indicating DOM with very little molecular weight in the stream for a very short time period.

Figure 33 shows the loadings of the four sources obtained by the NSS-SOBI-gFOBI-method with α_j from Eq. 53. The loadings corresponding to the wavelengths 250 and 365 have been highlighted and their six neighbouring loadings have been removed. The continuous line is at zero and the dashed lines are at the values of the loadings corresponding to the wavelengths 250 and 365. The corresponding loadings for source 2 are close to zero. The corresponding loadings for source 1 and 3 are not zero, but they are not symmetrical around zero. The loadings for source 4 are symmetrical around zero, which gives a loading to indicate the molecular weight.

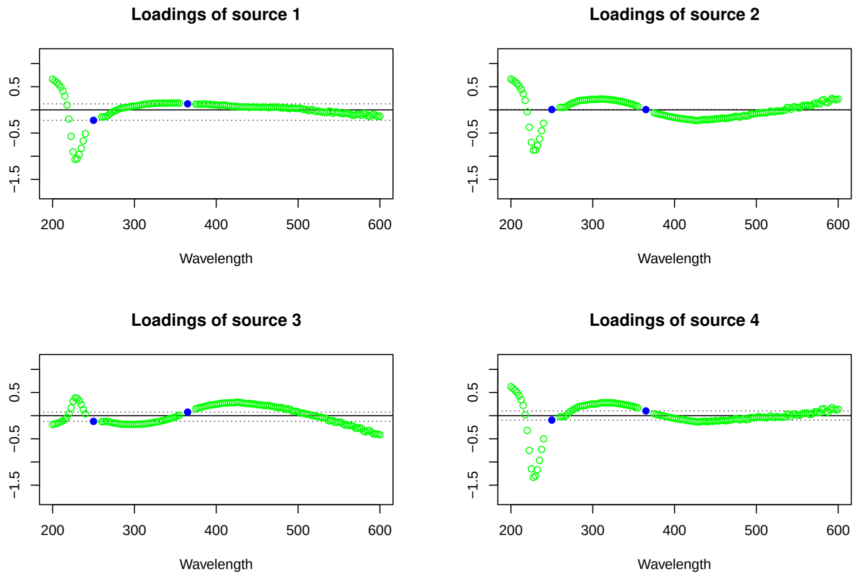

Figure 33: clr-laodings for wavelengths 250 and 365 in the model from the NSS-SOBI-gFOBI-method with α_j from 53.

7 Conclusion

Blind source separation is a popular modelling approach for time-series. In this thesis we showed how the modelling approach can be extended to compositional time series.

We therefore reviewed three different blind source separation models for time-series together with methods on how sources can be estimated in each model respectively. These methods include among others the novel method from Nordhausen et al. [2020]. We further discussed the two main log-ratio transformations for compositions: the clr-transformation, which is used for interpretability, and the ilr-transformation, which expresses the compositions within an orthonormal basis in the Euclidean vector space. Finally we showed how these transformations and the linear relationship between them can be used in the blind source separation framework in order to apply BSS on compositional data, and interpret the loadings in terms of the original compositions. Moreover we showed how principal component analysis can be added to make the important extension to highly correlated compositional time series. This extension proved to be very useful in the application of our new methodology on a real world data set. We showed how various BSS-methods can be applied to the absorption data from a small stream in Lower Austria. Additionally we sketched how the loadings and sources can be interpreted. Besides, we suggest the methodology of compositional data analysis, when absorption ratios are used to model DOM.

Since blind source separation has proven to be useful for modelling and interpreting multivariate time series, and many data sets are of compositional nature, we are confident there will be many more applications for our new method. An extension of our method could be to fit time-series models on the obtained sources. Additionally other blind source separation models and methods could be used for compositional time series.

8 Acknowledgements

I want to greatly thank my supervisor Prof. Nordhausen and Prof. Filzmoser for their incredible support during the creation of this thesis. They have been very quick to respond to my questions, and have always provided a lot of professional input in our numerous meetings.

Furthermore I would like to pay my special regards to Matthias Pucher from the University of Natural Resources and Life Sciences in Vienna. Not only has he provided the data set, but also given a lot of explanations and sophisticated input in various meetings.

Special thanks goes to my girlfriend Annika Held for proofreading my thesis. She has always motivated me very much during my studies. I also want to deeply thank my family, especially my parents, for facilitating my studies with their encouragement and assistance.

References

J. Aitchison. The statistical analysis of compositional data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(2):139–160, 1982.

J. Aitchison, C. Barceló-Vidal, J. A. Martín-Fernández, and V. Pawlowsky-Glahn. Logratio analysis and compositional distance. *Mathematical Geology*, 32(3):271–275, 2000.

A. Belouchrani and M. G. Amin. Blind source separation based on time-frequency signal representations. *IEEE Transactions on Signal Processing*, 46(11):2888–2897, 1998.

A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines. A blind source separation technique using second-order statistics. *IEEE Transactions on Signal Processing*, 45(2):434–444, 1997.

J. Bergman and B. Holmquist. Poll of polls: A compositional loess model. *Scandinavian Journal of Statistics*, 41(2):301–310, 2014.

T. M. Brunsdon and T. Smith. The time series analysis of compositional data. *Journal of Official Statistics*, 14(3):237–253, 1998.

S. Choi and A. Cichocki. Blind separation of nonstationary sources in noisy mixtures. *Electronics Letters*, 36(9):848–849, 2000.

P. Dawson, P. Downward, and T. C. Mills. Olympic news and attitudes towards the olympics: a compositional time-series analysis of how sentiment is affected by events. *Journal of Applied Statistics*, 41(6):1307–1314, 2014.

K. M. Docherty, K. C. Young, P. A. Maurice, and S. D. Bridgham. Dissolved organic matter concentration and quality influences upon structure and function of freshwater microbial communities. *Microbial Ecology*, 52(3):378–388, 2006.

A. Eder, P. Strauss, T. Krueger, and J. Quinton. Comparative calculation of suspended sediment loads with respect to hysteresis effects (in the Petzenkirchen catchment, Austria). *Journal of Hydrology*, 389(1-2):168–176, 2010.

J. J. Egozcue, V. Pawlowsky-Glahn, G. Mateu-Figueras, and C. Barcelo-Vidal. Isometric logratio transformations for compositional data analysis. *Mathematical Geology*, 35(3):279–300, 2003.

J. J. Egozcue, V. Pawlowsky-Glahn, M. Templ, and K. Hron. Independence in contingency tables using simplicial geometry. *Communications in Statistics-Theory and Methods*, 44(18):3978–3996, 2015.

K. Filzmoser, Peter Hron and M. Templ. *Applied compositional data analysis*. Springer, Cham, 2018.

E. Fišerová and K. Hron. On the interpretation of orthonormal coordinates for compositional data. *Mathematical Geosciences*, 43(4):455, 2011.

C. He and T. Teräsvirta. Fourth moment structure of the GARCH (p, q) process. *Econometric Theory*, 15(6):824–846, 1999.

C. Hu, F. E. Muller-Karger, and R. G. Zepp. Absorbance, absorption coefficient, and apparent quantum yield: A comment on common ambiguity in the use of these optical concepts. *Limnology and Oceanography*, 47(4):1261–1267, 2002.

M. Hubert and M. Debruyne. Minimum covariance determinant. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):36–43, 2010.

K. Ikeya and A. Watanabe. Direct expression of an index for the degree of humification of humic acids using organic carbon concentration. *Soil Science and Plant Nutrition*, 49(1):47–53, 2003.

P. Ilmonen, K. Nordhausen, H. Oja, and F. Theis. An affine equivariant robust second-order bss method. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 328–335. Springer, 2015.

P. Kynčlová, P. Filzmoser, and K. Hron. Modeling compositional time series with vector autoregressive models. *Journal of Forecasting*, 34(4):303–314, 2015.

P. Li and J. Hur. Utilization of UV-Vis spectroscopy and related data analyses for dissolved organic matter (DOM) studies: A review. *Critical Reviews in Environmental Science and Technology*, 47(3):131–154, 2017.

P. Massicotte and S. Markager. Using a Gaussian decomposition approach to model absorption spectra of chromophoric dissolved organic matter. *Marine Chemistry*, 180:24–32, 2016.

M. Matilainen, K. Nordhausen, and H. Oja. New independent component analysis tools for time series. *Statistics & Probability Letters*, 105:80–87, 2015.

M. Matilainen, J. Miettinen, K. Nordhausen, H. Oja, and S. Taskinen. On independent component analysis with stochastic volatility models. *Austrian Journal of Statistics*, 46:57–66, 2017.

M. Matilainen, C. Croux, J. Miettinen, K. Nordhausen, H. Oja, S. Taskinen, and J. Virta. tsbss: blind source separation and supervised dimension reduction for time series. *R package version 0.5.3*, 2019.

J. Miettinen, K. Illner, K. Nordhausen, H. Oja, S. Taskinen, and F. J. Theis. Separation of uncorrelated stationary time series using autocovariance matrices. *Journal of Time Series Analysis*, 37(3):337–354, 2014.

J. Miettinen, K. Nordhausen, and S. Taskinen. Blind source separation based on joint diagonalization in R: the packages JADE and BSSasymp. *Journal of Statistical Software*, 76:1–31, 2017.

J. Miettinen, M. Matilainen, K. Nordhausen, and S. Taskinen. Extracting conditionally heteroscedastic components using independent component analysis. *Journal of Time Series Analysis*, 41:293–311, 2018.

K. Nordhausen, G. Fischer, and P. , Filzmoser. Blind source separation for compositional time series. *to appear in Mathematical Geosciences*, 2020.

H. Oja, S. Sirkiä, and J. Eriksson. Scatter matrices and independent component analysis. *Austrian Journal of Statistics*, 35(2&3):175–189, 2006.

V. Pawlowsky-Glahn and J. J. Egozcue. Geometric approach to statistical analysis on the simplex. *Stochastic Environmental Research and Risk Assessment*, 15(5):384–398, 2001.

V. Pawlowsky-Glahn, J. J. Egozcue, and R. Tolosana-Delgado. *Modeling and analysis of compositional data*. Wiley, Chichester, 2015.

E. Perdue and J. Ritchie. Dissolved organic matter in freshwaters. *Treatise on Geochemistry*, 5:273–318, 2003.

J. Peuravuori and K. Pihlaja. Molecular size distribution and spectroscopic properties of aquatic humic substances. *Analytica Chimica Acta*, 337(2):133–149, 1997.

N. Shephard and T. G. Andersen. Stochastic volatility: origins and overview. In *Handbook of financial time series*, pages 233–254. Springer, 2009.

D. Silva and T. Smith. Modelling compositional time series from repeated surveys. *Survey Methodology*, 27(2):205–215, 2001.

P. Tichavsky and A. Yeredor. Fast approximate joint diagonalization incorporating weight matrices. *IEEE Transactions on Signal Processing*, 57(3): 878–891, 2009.

L. Tong, V. Soon, Y. Huang, and R. Liu. AMUSE: a new blind identification algorithm. In *IEEE International Symposium on Circuits and Systems*, pages 1784–1787. IEEE, 1990.

S.-J. You, Y. Yin, and H. E. Allen. Partitioning of organic matter in soils: effects of pH and water/soil ratio. *Science of the Total Environment*, 227 (2-3):155–160, 1999.

A R-Codes

A.1 Codes for fixing the issue with the lags

```
# Here we code functions for finding the true lag of two
# observations

#The following functions are used to calculate the true
#lag between two measurements. We need to extract the
#information of month, day, hour and minute from the time
#stamp in the form of "DD.MM.JJJ HH:MM"

Get.Day <- function(x){char <- substr(x,1,2)
  int <- strtoi(char,base = 10 ) #Conversion from string
  # to integer
  return(int)}

Get.Month <- function(x){char <- substr(x,4,5)
  int <- strtoi(char,base = 10 )
  return(int)}

Get.Hour <- function(x){char <- substr(x,12,13)
int <- strtoi(char,base = 10 )
return(int)}

Get.Minute <- function(x){char <- substr(x,15,16)
int <- strtoi(char,base = 10 )
return(int)}

Days.Of.Month<- function(mon){ #This function outputs the
#days of a month
  if(mon %in% c(1,3,5,7,8,10,12) ){return(31)}
  if(mon %in% c(4,6,9,11)){return(30)}
  if(mon == 2){return(28)}
}
Get.DayDifference <- function(day1,day2,month1,month2){
  if(month1 == month2)
    {return(day2-day1)}
  else{return(Days.Of.Month(month1)-day1+day2)}
}#This function only works if the difference between the
# months is utmost 1. If the difference is bigger than one,
# the days of the missed months are
#missed which is fixed with the next function
Get.Days.Of.Missed.Months <- function(month1,month2){
  daysofmissedmonths <- 0
  for(i in (month1+1):(month2-1) ){
```

```
  daysofmissedmonths <- daysofmissedmonths+Days.Of.Month(i)
  }
  return(daysofmissedmonths)
}
#This following function calculates the lag between two
#time stamps
Calculate.Lag <- function(date1,date2){
  month1 <- Get.Month(date1)
  day1 <- Get.Day(date1)
  hour1 <- Get.Hour(date1)
  minute1 <- Get.Minute(date1)

  month2 <- Get.Month(date2)
  day2 <- Get.Day(date2)
  hour2 <- Get.Hour(date2)
  minute2 <- Get.Minute(date2)

  daydiff <- Get.DayDifference(day1,day2,month1,month2)

  if(month2-month1 > 1){
    daydiff <- daydiff +
    Get.Days.Of.Missed.Months(month1,month2)
  }

  lag <- daydiff*144 + (hour2-hour1)*6 +
  (minute2 - minute1)/10
  #Here the true lag is calulated and output
  return(lag)
}

#The function Calculate.Lag is used to obtain a vector
#containing the true lag between two consecutive rows
#in the data matrix (scan15mm.2014)
nr <- nrow(scan15mm.2014)
lag.vector15mm.2014 <- rep(0,nr-1)


for(i in 1:(nr-1)){
 lag.vector15mm.2014[i]
  <-Calculate.Lag(rownames(scan15mm.2014)[i],
  rownames(scan15mm.2014)[i+1])
}

# A function for finding all pairs of  indices in the
#data matrix that are indeed measurements with a time
#difference of the desired lag
```

```
#Inputs: lag.vector (vector containing the true lags
# between consecutive
#observation, calculated in previous function)
#       lag (lag for which the indices want to be found)
get.vectors.with.true.lag.indices <- function(lag.vector,lag){
  nrow <- length(lag.vector)+1
  x1 <- rep(NaN,nrow-1)
  x2 <- rep(NaN,nrow-1)
  ne <- 1
  for(i in 1:(nrow-1)){
    for(j in 1:lag){
      if(sum(lag.vector[i:(min(i+j-1,nrow-1))]) == lag){
        x1[ne] <- i
        x2[ne] <- i+j
        ne <- ne +1
        break
      }
      if(sum(lag.vector[i:(min(i+j-1,nrow-1))]) > lag){
        break
      }
    }
  }

x1 <- x1[!is.na(x1)]
x2 <- x2[!is.na(x2)]

return(cbind(x1,x2))

}
```

A.2 Adjusting the SOBI-function

The following codes describe adjustments of the R-function *SOBI* in the R-package *JADE* [Miettinen et al., 2017], in order for it to be applicable to our concrete data set.

```
#The following function is an adjustment of the SOBI function
SOBI.dif.lags <- function (X, k = 12, method = "frjd",
eps = 1e-06, maxiter = 100,
lagvector = rep(1,nrow(X)-1))
#Here the lag-vector containing the true lags between two
#consecutive rows is an additional input
{
  if (length(k) == 1)
    k <- 1:k
  nk <- length(k)
```

```
method <- match.arg(method, c("rjd", "djd", "frjd"))
#Here between different joint diagonalisation methods can
#be chosen
MEAN <- colMeans(X)
COV <- cov(X)
EVD <- eigen(COV, symmetric = TRUE)
COV.sqrt.i <- EVD$vectors %*%
tcrossprod(diag(EVD$values^(-0.5)), EVD$vectors)
X.C <- sweep(X, 2, MEAN, "-")
Y <- tcrossprod(X.C, COV.sqrt.i) #Here the whitening of
#the data matrix happens
p <- ncol(X)
R <- array(0, dim = c(p, p, nk))
n <- nrow(X)
lag.matrix.array <- list()
for (i in 1:nk) {
  lag.matrix <-
  get.vectors.with.true.lag.indices(lagvector,k[i])
  #the vector k contains the chosen lags for the SOBI
  #method
  #Here our adjustment of the estimator of the
  #auto-covariance matrix is implemented. The rows of
  # the contains all pairs of observations in
  #the data matrix with a time difference of lag k[i]
  Yt <- Y[lag.matrix[,1],]
  Yti <- Y[lag.matrix[,2],]
  Ri <- crossprod(Yt, Yti)/nrow(Yt)
  R[, , i] <- (Ri + t(Ri))/2
  lag.matrix.array[[i]] <- lag.matrix
}
JDoutput <- switch(method, frjd = {
  #Here the joint diagonalisation is done
  frjd(R, eps = eps, maxiter = maxiter)
}, rjd = {
  rjd(R, eps = eps, maxiter = maxiter)
}, djd = {
  djd(R, eps = eps, maxiter = maxiter, ...)
})

JD <- JDoutput$V

W <- crossprod(JD, COV.sqrt.i)
#Calcualtion of the final unmixing matrix
W <- sweep(W, 1, sign(rowMeans(W)), "*")
S <- tcrossprod(X.C, W) #Calculation of the sources
ssq_ac <- rep(0,p)
```

```
#Here the sum of the squared auto-correlations for each
#latent source is calculated. This information could be
#used for dimension reduction, which was not
#needed in our case
  for(j in 1:p){
    for(i in 1:nk){
      ssq_ac[j] <- ssq_ac[j]+JDoutput$D[j,j,i]^2
    }
  }

  #
  ord <- order(ssq_ac, decreasing = TRUE)
  P <- matrix(0, p, p)
  for (j in 1:p) {
    P[j, ord[j]] <- 1
  }
  #Here the sources get ordered according to the
  #above calculated sum of squared auto-correlations,
  # usually used for dimension reduction,
  #but was not needed in our case

  S <- S[, ord]
  W <- P %*% W
  S <- ts(S, names = paste("Series", 1:p))
  RES <- list(W = W, k = k, method = method, S = S)
  class(RES) <- "bss"
  RES
}
```

The following code is an implementation of the additional robustification adjustment as described in BSS-method 3.

```
SOBI.dif.lags.rob <- function (X, k = 12, method = "frjd",
eps = 1e-06, maxiter = 100,
                           lagvector = rep(1,nrow(X)-1))
#Also here the lagvector is patched as an additional input
{
  if (length(k) == 1)
    k <- 1:k
  nk <- length(k)
  method <- match.arg(method, c("rjd", "djd", "frjd"))

  robustscaleandcenter <- robustbase::covMcd(X,cor=FALSE)
  MEAN <- robustscaleandcenter$center
  COV <- robustscaleandcenter$cov
```

```
EVD <- eigen(COV, symmetric = TRUE)
COV.sqrt.i <- EVD$vectors %*%
tcrossprod(diag(EVD$values^(-0.5)),
                                       EVD$vectors)
#Here the robust whitening is performed using the
#robust MCD for location and scatter
X.C <- sweep(X, 2, MEAN, "-")
Y <- tcrossprod(X.C, COV.sqrt.i)
p <- ncol(X)
R <- array(0, dim = c(p, p, nk))
n <- nrow(X)
lag.matrix.array <- list()
for (i in 1:nk) {
  lag.matrix <-
   get.vectors.with.true.lag.indices(lagvector,k[i])
  Yt <- Y[lag.matrix[,1],]
  Yti <- Y[lag.matrix[,2],]
  #Here the estimator of the spatial sign
  #auto-covriance matrix is
  #adjusted using the true lags
  for(i in 1:(nrow(Yt))){
    Yt[i,] <- Yt[i,]/norm_vec(Yt[i,])
    Yti[i,] <- Yti[i,]/norm_vec(Yti[i,])
  }
  #the function norm_vec outputs the
  #Euklidean norm of a vector
  Ri <- crossprod(Yt, Yti)/nrow(Yt)



  R[, , i] <- (Ri + t(Ri))/2
  lag.matrix.array[[i]] <- lag.matrix
}
#The rest of the function is the same as in the
#adjustment of the classical SOBI

JDoutput <- switch(method, frjd = {
  frjd(R, eps = eps, maxiter = maxiter)
}, rjd = {
  rjd(R, eps = eps, maxiter = maxiter)
}, djd = {
  djd(R, eps = eps, maxiter = maxiter, ...)
})

JD <- JDoutput$V
```

```
  W <- crossprod(JD, COV.sqrt.i)
  W <- sweep(W, 1, sign(rowMeans(W)), "*")
  S <- tcrossprod(X.C, W)
  ssq_ac <- rep(0,p)

  #
  for(j in 1:p){
    for(i in 1:nk){
      ssq_ac[j] <- ssq_ac[j]+JDoutput$D[j,j,i]^2
    }
  }


  #
  ord <- order(ssq_ac, decreasing = TRUE)
  P <- matrix(0, p, p)
  for (j in 1:p) {
    P[j, ord[j]] <- 1
  }
  S <- S[, ord]
  W <- P %*% W
  S <- ts(S, names = paste("Series", 1:p))
  RES <- list(W = W, k = k, method = method, S = S)
  class(RES) <- "bss"
  RES
}
```

A.3 Adjusting the gFOBI-function

The following code shows an adjustment of the R-function *gFOBI* from the package *tsBSS* [Matilainen et al., 2019].

```
gFOBI.dif.lags <- function (X, k = 0:12, eps = 1e-06,
maxiter = 100, method = "frjd",
na.action = na.fail, weight = NULL, ordered = FALSE,
acfk = NULL, original = TRUE, alpha = 0.05,lagvector)
{# like in the adjusted SOBI functions, additionally
#the vector containing the
  #true lags is patched
  nk <- length(k)
  method <- match.arg(method, c("rjd", "frjd"))
  MEAN <- colMeans(X)
  COV <- cov(X)
  EVD <- eigen(COV, symmetric = TRUE)
  COV.sqrt.i <- EVD$vectors %*%
  tcrossprod(diag(EVD$values^(-0.5)),  EVD$vectors)
```

```
#Here the whitening of the data matrix happens
X.C <- sweep(X, 2, MEAN, "-")
Y <- tcrossprod(X.C, COV.sqrt.i)
p <- ncol(X)
R <- array(0, dim = c(p, p, nk))
n <- nrow(X)
for (i in 1:nk) {

  lag.matrix <-
   get.vectors.with.true.lag.indices(lagvector,k[i])
  Yt <- Y[lag.matrix[,1],]
  Yti <- Y[lag.matrix[,2],]

  #Here the estimator of the fourth order-crossmoment-
  #matrix is calculated
  r <- sqrt(rowSums(Yt^2))
  Yu <- r * Yti
  Ri <- crossprod(Yu)/nrow(Yt)
  R[, , i] <- Ri

}
JD <- switch(method, frjd = {
#Here the joint diagonalisation is done
 frjd(R, eps = eps, maxiter = maxiter,
  na.action = na.action,
       weight = weight)$V
}, rjd = {
  rjd(R, eps = eps, maxiter = maxiter, na.action =
  na.action)$V
})
W <- crossprod(JD, COV.sqrt.i)
#Calculation of the final unmixing matrix
W <- diag(sign(rowMeans(W))) %*% W
S <- tcrossprod(X.C, W) #Calculation of the sources
if (ordered == TRUE) {
  if (is.null(acfk) == TRUE) {
    acfk <- k
  }
  ord <- ordf(S, acfk, p, W, alpha, ...)
  W <- ord$W
  if (original == TRUE) {
    S <- ord$S
  }
  else {
    S <- ord$RS
    Sraw <- ord$S
```

```
      Sraw <- ts(Sraw, names = paste("Series", 1:p))
      if (is.ts(X))
        attr(Sraw, "tsp") <- attr(X, "tsp")
    }
  }
  S <- ts(S, names = paste("Series", 1:p))
  RES <- list(W = W, k = k, S = S, MU = MEAN)
  if (ordered == TRUE) {
    if (original == FALSE) {
      RES$Sraw <- Sraw
    }
    RES$fits <- ord$fits
    RES$armaeff <- ord$armaeff
    RES$linTS <- ord$linTS
    RES$linP <- ord$linP
    RES$volTS <- ord$volTS
    RES$volP <- ord$volP
  }
  class(RES) <- c("bssvol", "bss")
  RES
}
```

A.4 Code for the NSS-SOBI-gFOBI-method

This subsection contains the code for the NSS-SOBI-gFOBI-method from Nordhausen et al. [2020], also adjusted for our data-set.

```
MS.x <-
  function (X, Tau = 0, lagvector)
  #This function returns the estimated auto-covariance matrix
  # for a chosen lag using the lagvector containing the
  #information about the true lags
  {

    lag.matrix <-
    get.vectors.with.true.lag.indices(lagvector,Tau)

    n <- nrow(X)

    Xt <- X[lag.matrix[,1],]
    Xti <- X[lag.matrix[,2],]


    Ri <- crossprod(Xt, Xti)/nrow(Xt)
    return((Ri + t(Ri))/2)
  }
```

```
MF.x <-
  function (X, Tau = 0,lagvector)
  {
    #This function returns the estimated
    #fourth-order-cross-moment-matrix for a chosen lag
    #using the lagvector containing the information about
    #the true lags
    lag.matrix <-
    get.vectors.with.true.lag.indices(lagvector,Tau)

    n <- nrow(X)

    Xt <- X[lag.matrix[,1],]
    Xti <- X[lag.matrix[,2],]

    r <- sqrt(rowSums(Xt^2))
    Xu <- r * Xti
    Ri <- crossprod(Xu)/nrow(Xt)
    return(Ri)
  }

SM <- function(M)
{ # This function is used to implement the weights
# with the maximum element in the joint
#diagonalisation
  maxM <- max(abs(M))
  if (maxM > 1) M <- M/maxM
  return(M)
}

BSSmix <- function(X, tauS, tauF, K = 12, n.cuts = NULL,
eps = 1e-06,  maxiter = 100, lagvector)
{#tauS is the lag-set for the auto-covariance matrices
  #tauF is the lag-set for the fourth-order cross-moment
  #matrices
  #per default the time-period is divided into 12 equal
  #parts also the lagvector is used for adjusting the
  #estimators
  n <- nrow(X)
  MEAN <- colMeans(X)
  COV <- cov(X)
  EVD.COV <- eigen(COV, symmetric = TRUE)
  COV.sqrt.inv <- EVD.COV$vectors %*%
  tcrossprod(diag(sqrt(1/EVD.COV$values)),
                    EVD.COV$vectors)
```

```
X.C <- sweep(X, 2, MEAN, "-")
Y <- tcrossprod(X.C, COV.sqrt.inv)
 # Here the whitening of the data matrix is done
p <- ncol(X)
if (is.null(n.cuts))
  n.cuts <- ceiling(seq(1, n, length = K + 1))
else K <- length(n.cuts) - 1
N.cuts <- n.cuts + c(rep(0, K), 1)
LS <- length(tauS)
LF <- length(tauF)
R <- array(0, dim = c(p, p, (LS + LF) * K))
R1 <- R
R2 <- R
NAMES <- character((LS + LF) * K)
ii <- 1
for (i in 1:K) {
#Here the data matrix is divided into 12 equaly
#sized parts
  Y.i <- Y[N.cuts[i]:(N.cuts[i + 1] - 1), ]
  lagvector.i <- lagvector[N.cuts[i]:(N.cuts[i+1]-2)]
  for (j in 1:LS) {
    R[, , ii] <- MS.x(Y.i, Tau = tauS[j],lagvector.i)
    R1[, , ii] <- R[, , ii]
    R2[, , ii] <- SM(R[, , ii])
    NAMES[ii] <- paste0("ACOV_lag_",tauS[j],"_int_",i)
    ii <- ii + 1
  }
  for (k in 1:LF) {
    R[, , ii] <- MF.x(Y.i, Tau = tauF[k],lagvector.i)
    # Here the different weights are implemented
    R1[, , ii] <- R[, , ii]/(p+2)
    R2[, , ii] <- SM(R[, , ii])
    NAMES[ii] <- paste0("FCOV_lag_",tauF[k],"_int_",i)
    ii <- ii + 1
  }
}
JD <- frjd(R, eps = eps, maxiter = maxiter)
#Here the joint diagonalisation is done
#for each of the weights
JD1 <- frjd(R1, eps = eps, maxiter = maxiter)
JD2 <- frjd(R2, eps = eps, maxiter = maxiter)
D <- t(apply(JD$D,3,diag))
D1 <- t(apply(JD1$D,3,diag))
D2 <- t(apply(JD2$D,3,diag))
colnames(D) <- paste0("p",1:p)
rownames(D) <- NAMES
```

```
  colnames(D1) <- paste0("p",1:p)
  rownames(D1) <- NAMES
  colnames(D2) <- paste0("p",1:p)
  rownames(D2) <- NAMES
  W <- crossprod(JD$V, COV.sqrt.inv)
  #Calculation of the final unmixing matrix
  #for each of the weights
  W1 <- crossprod(JD1$V, COV.sqrt.inv)
  W2 <- crossprod(JD2$V, COV.sqrt.inv)
  S <- tcrossprod(X.C, W)
  #Calculation of the sources for each of the weights
  S <- ts(S, names = paste("Series", 1:p))
  RES <- list(W = W, D=D, tauS =tauS, tauF=tauF,
  n.cut = n.cuts, K = K, S = S,
   W1 = W1, D1=D1,W2 = W2, D2=D2)
  class(RES) <- "bss"
  RES
}
```