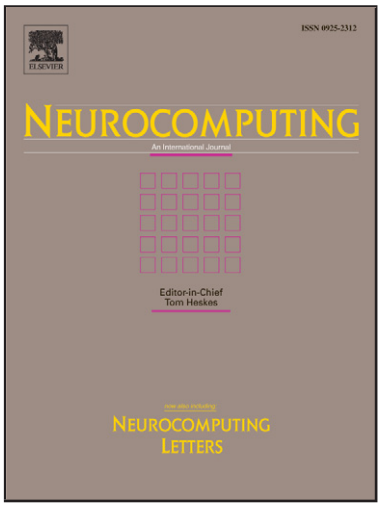
Author's Accepted Manuscript



www.elsevier.com/locate/neucom

Hybridizing sparse component analysis with genetic algorithms for microarray analysis

K. Stadlthanner, F.J. Theis, E.W. Lang, A.M. Tomé, C.G. Puntonet, J.M. Górriz

PII: S0925-2312(08)00105-7
DOI: doi:10.1016/j.neucom.2007.09.017
Reference: NEUCOM 11019

To appear in: *Neurocomputing*

Cite this article as: K. Stadlthanner, F.J. Theis, E.W. Lang, A.M. Tomé, C.G. Puntonet and J.M. Górriz, Hybridizing sparse component analysis with genetic algorithms for microarray analysis, *Neurocomputing* (2008), doi:10.1016/j.neucom.2007.09.017

Hybridizing Sparse Component Analysis with Genetic Algorithms for Microarray Analysis

K. Stadlthanner, F. J. Theis, E. W. Lang

Institute of Biophysics, Computational Intelligence Group
University of Regensburg, 93040 Regensburg, Germany
email: elmar.lang@biologie.uni-regensburg.de

A. M. Tomé

Dept. de Electrónica e Telecomunicações/IEETA
Universidade de Aveiro, 3810 Aveiro, Portugal
email: ana@ieeta.pt

C. G. Puntonet, J. M. Górriz

Dept. Arqitectura y Tecnología de Computadores
Universidad de Granada, 18371 Granada, Spain
email: carlos@atc.ugr.es

Abstract

Nonnegative Matrix Factorization (NMF) has proven to be a useful tool for the analysis of nonnegative multivariate data. However, it is known not to lead to unique results when applied to Blind Source Separation (BSS) problems. In this paper we present an extension of NMF capable of solving the BSS problem when the underlying sources are sufficiently sparse. In contrast to most well-established BSS methods, the devised algorithm is capable of solving the BSS problem in cases where the underlying sources are not independent or uncorrelated. As the proposed fitness function is discontinuous and possesses many local minima, we use a genetic algorithm for its minimization. Finally, we apply the devised algorithm to real world microarray data.

Key words: sparse nonnegative matrix factorization, blind source separation, gene microarray analysis

1 Introduction

Environmental stimuli cause the induction or repression of genes in living cells via a corresponding up- or down-regulation of the amount of messenger RNA (mRNA). Different experimental conditions may thus result in different characteristic expression patterns. The expression of each gene thereby relies on the specific processing of a number of regulatory inputs. Recently, high throughput methods like microarrays have become available and allow to measure whole genome wide gene expression profiles [2]. Intelligent data analysis tools are needed to unveil the information hidden in those microarray data sets [32].

Besides traditional supervised approaches, matrix factorization techniques like principal component analysis (PCA), independent component analysis (ICA), sparse component analysis (SCA) [17], [7] or non-negative matrix factorization (NMF) [20] might be useful to go beyond simple clustering and decompose such data sets into component profiles which might be indicative of underlying biological processes [21], [34], [5]. Without any hypothesis, such unsupervised approaches might be able to discover novel biological mechanisms and reveal genetic regulatory networks in large data sets when little *a priori* knowledge is available.

Unsupervised analysis methods for microarray data analysis can be divided into three categories:

Clustering approaches
They group genes with similar behavior under similar experimental conditions making it possible to analyze data within each group separately. It is supposed that genes within a cluster are functionally related. In general no attempt is made to model the underlying biology. Note that clusters are disjoint but genes may participate in several biological processes.

Model-based approaches
They generate a model explaining the interactions among the biological entities participating in gene regulatory networks. Parameters of the model are trained on expression data sets. With complex models not enough data may be available to estimate the parameters. Also algorithms are often of prohibitive complexity and computational load.

Matrix decomposition methods
They decompose any given data matrix into a product of two matrices with desired properties. The latter are imposed as constraints on the matrix de-

composition procedure. These techniques can equivalently be regarded as expanding the given data vectors into a basis of desired property. Famous among such projection methods are the following:

- **PCA** projects data into a space spanned by mutually orthogonal principal components (PCs). Each principal component captures the maximum information, i.e. variance, that is not already present in the previous components. PCA can be used for data compression as well and it is the optimal dimension-reduction technique in the sum-of-squared-error sense. Dimension reduction of expression data can be used for visualization, filtering of noise or simplifying subsequent computations. With microarray data, the principal components are called *eigenarrays*.
- **ICA** decomposes the data into statistically independent components (ICs). A common application of ICA is in blind source separation (BSS) problems, for example of EEG [27], [14], MEG [41] and fMRI [42], [19] data. ICA can also be used to reduce noise [40], [35], [12] or artifacts [36, 37] if generated from independent sources. Usually a linear superposition of the underlying unknown source signals is assumed but nonlinear ICA algorithms also exist. With microarray data, ICA extracts *expression modes*, the ICs, each of which represents a linear influence of a hidden cellular variable. Each retrieved IC is considered to be indicative of a putative biological process, which can be characterized by the functional annotations of genes that are predominant within the independent component. Each component thus defines corresponding groups of induced and repressed genes. Samples and genes can be visualized by projecting them to particular expression modes or to their influences, respectively. A projection to expression modes helps to highlight particular biological functions, to reduce noise, and to compress the data in a biologically sensible way.
- **NMF** has been suggested by [20] to provide a decomposition of images into parts which are amenable to a more intuitive interpretation. It decomposes a given data matrix into a product of two strictly non-negative matrices. When applied to microarrays this constraint seems natural as the raw fluorescence intensities measured can have only non-negative values, of course. Thus this technique alleviates some of the problems which arise with PCA and ICA both of which yield negative entries in their component expression profiles. The latter have no obvious interpretation. In most applications so far this fact is gently ignored and the negative entries are turned positive by simply considering only the absolute values of the entries to the component profiles. In the following we give a more detailed account of the NMF method.

2 Matrix Factorization and Blind Source Separation

In the field of modern data analysis mathematical transforms of the observed data are often used to unveil hidden causes. Especially in situations where different observations of the same process are available matrix factorization techniques have proven useful [22]. Thereby, the $M \times T$ observation matrix $\mathbf{X}$ is decomposed into a $M \times N$ matrix $\mathbf{W}$ and a $N \times T$ matrix $\mathbf{H}$

$$\mathbf{X} = \mathbf{WH}. \tag{1}$$

Here, it is assumed that M observations, consisting of T samples, constitute the rows of $\mathbf{X}$ and that $M \leq N$. Obviously, the decomposition is highly non-unique and can only be solved uniquely if additional conditions constraining the row vectors of $\mathbf{H}$ or the column vectors of $\mathbf{W}$ are given.

One application of matrix factorization is linear blind source separation (BSS), where the observed signals $\mathbf{X}$ can be considered a weighted superposition of N underlying source signals. If the source signals form the rows of the $N \times T$ matrix $\mathbf{S}$, then each element a_{ij} of the mixing matrix $\mathbf{A}$ represents the weight with which the j-th source signal contributes to the i-th observed signal. Thus the matrix $\mathbf{X}$ of observed signals can be decomposed as

$$\mathbf{X} = \mathbf{AS}. \tag{2}$$

In BSS now, given only the matrix $\mathbf{X}$, a matrix factorization as in (1) is sought such that $\mathbf{A}$ and $\mathbf{S}$ are essentially equal to $\mathbf{W}$ and $\mathbf{H}$ for any given constraints, i.e. they are identical up to inherent scaling and permutation indeterminacies.

Note that in the sequel we will confine ourselves to the symmetric BSS problem where the number of source signals to be recovered equals the number of observed signals, i.e. we will present an algorithm for the case $M = N$.

3 Sparse Nonnegative Blind Source Separation

3.1 Nonnegative matrix factorization

A natural constraint to many real world problems is reflected in nonnegative matrix factorization (NMF) where the source matrix $\mathbf{S}$, the mixing matrix $\mathbf{A}$ as well as the observation matrix $\mathbf{X}$ are assumed to be strictly nonnegative. Although NMF has been shown to yield quite intuitive data decompositions

in the field of image and text analysis [20], it cannot solve the BSS problem uniquely up to scaling and permutation indeterminacies, hence additional constraints are needed.

With microarrays a naturally arising additional constraint seems to be the assumption that the source signals are sparsely represented, i.e. the signal vectors $\mathbf{s}_n$, which form the rows of the source signal matrix $\mathbf{S}$, have many close-to-zero entries. Such sparse coding concepts have since long been discussed in the vision community [33], [29], [18], [23] and have already been exploited successfully in NMF based image analysis methods [15] as well as in other BSS algorithms [24]. Considering gene expression profiles, it may be expected that only a small number of genes are highly up- or down-regulated within a single process if underlying source signals should be indicative of ongoing biological processes in cells.

An extension to standard NMF is thus presented in this paper. It is only assumed that the row vectors of $\mathbf{S}$ contain several close-to-zero components, while their exact sparseness need not be known. The proposed algorithm, called sparse NMF (sNMF), tries to find a nonnegative matrix factorization of the data matrix, for which the source matrix $\mathbf{S}$ has the largest number of close-to-zero components. As will be shown in several simulations, this approach is capable of solving the BSS problem uniquely up to the usual scaling and permutation indeterminacies inherent in BSS.

The basic idea is to estimate the original source matrix $\mathbf{S}$ and mixing matrix $\mathbf{A}$, respectively, by determining two estimates $\hat{\mathbf{A}}, \hat{\mathbf{S}}$, of the nonnegative matrices $\mathbf{A}$ and $\mathbf{S}$ such that

(1) $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$ are both nonnegative
(2) the rows of the matrix $\hat{\mathbf{S}}$ are as sparsely encoded as possible, i.e. contain as many close-to-zero components as possible
(3) the reconstruction error $F(\hat{\mathbf{A}}, \hat{\mathbf{S}}) = ||\mathbf{X} - \hat{\mathbf{A}}\hat{\mathbf{S}}||^2$ of the estimated observations is as small as possible.

Fig. 1 illustrates this approach. Fig. 1 a) shows the scatter plot of two random nonnegative sparse sources which constitute the rows of the matrix $\mathbf{S}$. Obviously, all points in the scatterplot lie in the first quadrant of the coordinate system or, in other words, reside within a conic hull with cone lines defined by those columns of $\mathbf{S}$ which contain at least one null entry. By left multiplying $\mathbf{S}$ by a nonnegative mixing matrix $\mathbf{A}$, the mixture matrix $\mathbf{X}$ is obtained whose scatterplot is depicted in Fig. 1 b). The points in this scatterplot are now contained in a conic hull whose cone lines are defined by the images of those points which were lying on the axis of the coordinate system in Fig. 1 a).

The mixture matrix $\mathbf{X}$ can now be decomposed into two nonnegative matrices $\mathbf{W}$ and $\mathbf{H}$ by means of NMF. However, this decomposition is not unique, i.e.

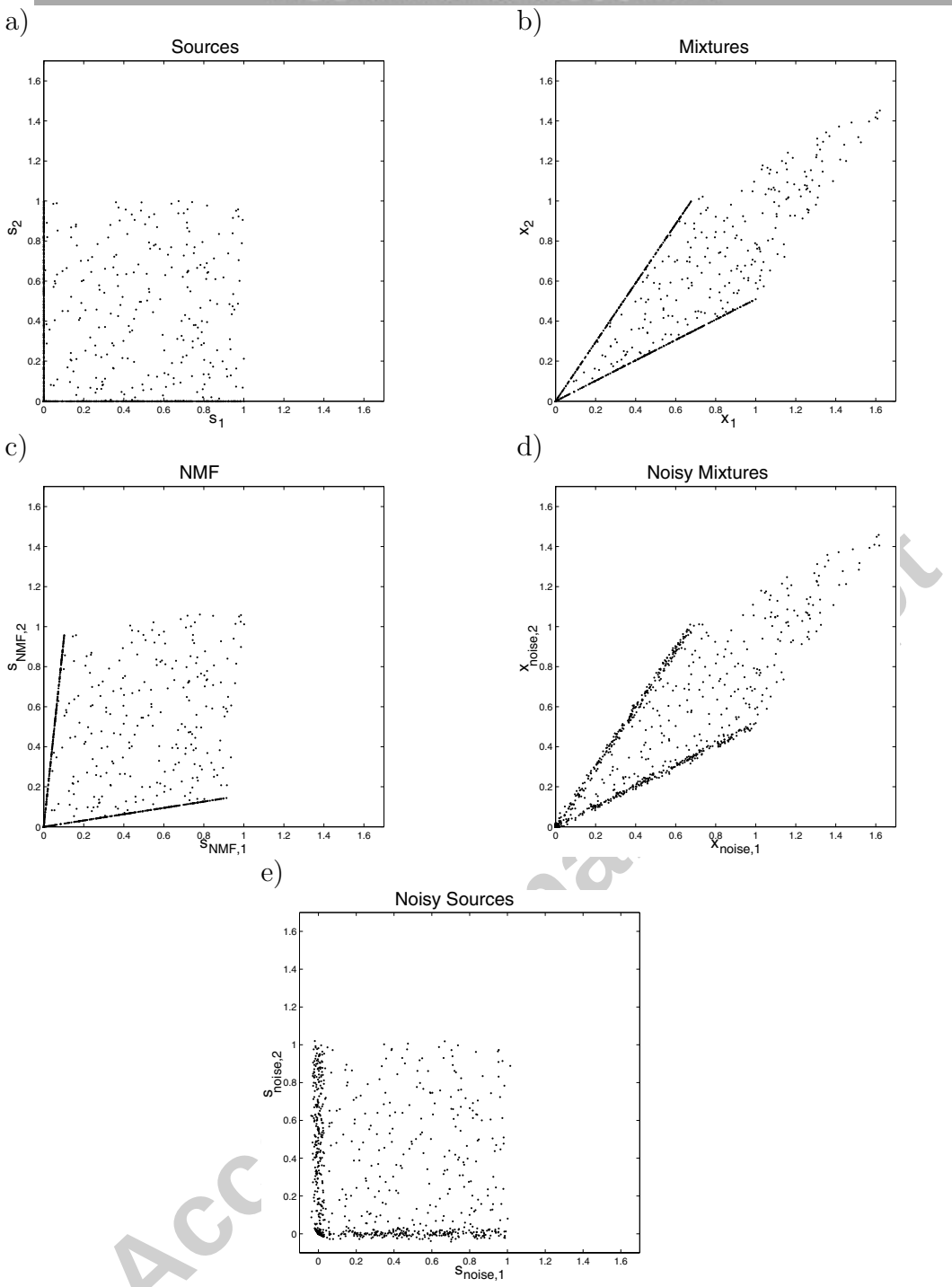


Fig. 1. Illustration of NMF and sNMF. a) Scatterplot of two random nonnegative sources used to constitute the rows of the matrix $\mathbf{S}$. b) Scatterplot of the mixtures $\mathbf{X}$ obtained by multiplying $\mathbf{S}$ by a nonnegative 2×2 mixing matrix $\mathbf{A}$. Note that all data points remain in the first quadrant. c) One of the infinitely many possible NMF factorizations of $\mathbf{X}$. Requiring only nonnegativity of $\mathbf{A}$ and $\mathbf{S}$ is insufficient to solve the BSS problem uniquely up to scaling and permutation indeterminacies. d) Scatterplot of noisy mixtures $\mathbf{X}_{noise}$ which were generated by adding random Gaussian noise to $\mathbf{X}$. e) Scatterplot of noisy sources $\mathbf{S}_{noise} = \mathbf{A}^{-1}\mathbf{X}_{noise}$. The Gaussian noise added to $\mathbf{X}$ leads to negative entries in $\mathbf{S}_{noise}$ even if the original mixing matrix can be recovered and generally exacerbates the detection of nil-entries.

apart from the original mixing and source matrix an infinite number of matrix pairs $\mathbf{W}$ and $\mathbf{H}$ exists. They are also nonnegative and perfectly factorize $\mathbf{X}$ but which do not equal the original source and mixing matrix due to scaling and permutation indeterminacies. As illustrated in Fig. 1 c) the conic hulls of such matrices $\mathbf{H}$ also reside in the first quadrant but their cone lines do not coincide with the axes of the coordinate system.

Hence, our approach to recover the original mixing and source matrix (except for scaling and permutation indeterminacies) is to select among all possible nonnegative factorizations of $\mathbf{X}$, the one for which the cone lines of $\mathbf{H}$ coincide with the axes of the coordinate system or for which $\mathbf{H}$ contains as many null entries as possible.

However, data from real life experiments is always corrupted by noise. Hence, only an approximate factorization of $\mathbf{X}$ is usually feasible. Furthermore, additional noise also blurs the null entries in the sources (see Fig. 1d) and e)) such that they may even contain small negative elements. To cope with this problem, close-to-zero elements have to be set to zero when the sparseness of the estimated sources is determined.

To solve the sNMF problem algorithmically, we propose to estimate two nonnegative matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$ which minimize the following cost function $E(\hat{\mathbf{A}}, \hat{\mathbf{S}})$

$$E(\hat{\mathbf{A}}, \hat{\mathbf{S}}) = \frac{1}{2M}||n(\mathbf{X}) - n(\hat{\mathbf{A}}\hat{\mathbf{S}})||^2 - \frac{\lambda}{M}\sum_{m=1}^{M}\sigma_\tau(\hat{\mathbf{s}}_m), \tag{3}$$

where the function $n(..)$ is used to normalize the row vectors of $\mathbf{X}$ and $\hat{\mathbf{A}}\hat{\mathbf{S}}$, i.e.

$$n(\mathbf{X}) = \hat{\mathbf{X}} \quad \text{such that} \quad \hat{x}_{ij} = \frac{x_{ij}}{\sqrt{\sum_{k=1}^{T} x_{ik}^2}} \tag{4}$$

Thus it holds that

$$0 \leq \frac{1}{2M}\|n(\mathbf{X}) - n(\hat{\mathbf{A}}\hat{\mathbf{S}})\|^2 \leq 1 \tag{5}$$

In these equations σ_τ denotes an appropriate sparseness measure given below, $\lambda > 0$ represents a Lagrange weighting factor, and $\hat{\mathbf{s}}_m$ denotes the m-th row of the matrix $\hat{\mathbf{S}}$.

Recall that we assume $\mathbf{A}$ to be a full rank square matrix throughout this

paper. Hence, $\hat{\mathbf{S}}$ can be determined straightforwardly as

$$\hat{\mathbf{S}} = \hat{\mathbf{A}}^{-1}\mathbf{X} \tag{6}$$

once a good estimate $\hat{\mathbf{A}}$ of the original mixing matrix $\mathbf{A}$ has been obtained. Eventually, this means that the above optimization problem depends only on $\hat{\mathbf{A}}$.

3.2 Sparseness Measure

For the sparse nonnegative BSS problem at hand, we define the sparseness σ_τ of a row vector $\mathbf{s}$ as the ratio of the number of its zero elements and the total number of components. However, measurements in real world experiments are always corrupted by noise. Thus small nonzero entries of a measurement vector should be set to zero as well. Hence, a nonnegative threshold is used to define the minimum value which avoids any component of $\mathbf{s}_n$ being zero. This threshold is defined as a fraction $\tau \in [0, 1]$ of the maximal component s_n^{max} of $\mathbf{s}_n$. Formally this leads to the following sparseness measure σ_τ:

$$0 \leq \sigma_\tau(\mathbf{s}_m) = \frac{\text{number of elements of } \mathbf{s}_m \leq \tau \cdot \mathbf{s}_m^{max}}{\text{number of elements of } \mathbf{s}_m} \leq 1, \tag{7}$$

where $\mathbf{s}_m^{max}$ is the maximum value of $\mathbf{s}_m$ and $\tau \in [0, 1]$. Thus, the regularizing sparseness term, i.e. the second summand in the above given cost function, has values also in the range $[0, 1]$ only. The Lagrange parameter λ is used to balance the factorization of the data matrix with the sparseness requirement. As both terms in the cost function are normalized, results should be robust against varying sizes of $\mathbf{X}$ and $\mathbf{S}$ as soon as an appropriate λ has been obtained.

In the literature various measures $\sigma(\mathbf{s})$ have been proposed to estimate the sparseness of a signal without counting its nil entries explicitly. These measures are usually based on L_n norm considerations and are computationally less demanding than the sparseness measure in (7).

A prominent example of such a measure is the normalized ratio of the L_1 and L_2 norm of an T-dimensional vector $\mathbf{s}$ [15] according to

$$\sigma(\mathbf{s}) = \frac{\sqrt{T} - \sum_{i=1}^{T} |s_i| / \sqrt{\sum_{i=1}^{T} s_i^2}}{\sqrt{T} - 1}, \tag{8}$$

where s_i is the i-th component of $\mathbf{s}$. As will be shown in the following, however, such estimates of the sparseness of a signal are inadequate for the BSS task

at hand. To illustrate this point, consider the nonnegative BSS model

$$\mathbf{X} = \mathbf{AS} \qquad \mathbf{X}, \mathbf{S} \in \mathbb{R}^{M \times T}, \mathbf{A} \in \mathbb{R}^{M \times M} \tag{9}$$

in which $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{X}$ are all nonnegative and the rows of $\mathbf{S}$ contain many close-to-zero components. In sNMF, given only $\mathbf{X}$, the matrices $\mathbf{A}$ and $\mathbf{S}$ need to be recovered by searching for two nonnegative matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$ such that $\mathbf{X} = \hat{\mathbf{A}}\mathbf{D}^{-1}\mathbf{P}^{-1}\mathbf{P}\mathbf{D}\hat{\mathbf{S}}$ and the rows of $\hat{\mathbf{S}}$ are encoded as sparse as possible. Thereby the matrices $\mathbf{P}$ and $\mathbf{D}$ represent a permutation matrix and a diagonal scaling matrix, respectively. Obviously, such an approach can only succeed if among all the possible nonnegative factorizations of $\mathbf{X}$, i.e.

$$\{(\hat{\mathbf{A}}, \hat{\mathbf{S}}), \hat{\mathbf{A}} \in \mathbb{R}^{M \times M},\ \hat{\mathbf{S}} \in \mathbb{R}^{M \times T},\ \hat{\mathbf{A}}, \hat{\mathbf{S}} \text{ nonnegative} \mid \hat{\mathbf{A}}\,\hat{\mathbf{S}} = \mathbf{X}\}, \tag{10}$$

no pair of matrices $(\hat{\mathcal{A}}, \hat{\mathcal{S}}) \neq (\mathbf{A}, \mathbf{S})$ exists with the rows of $\hat{\mathcal{S}}$ being more sparse than the rows of $\mathbf{S}$.

But this seems to happen occasionally if the sparseness of the matrix $\hat{\mathbf{S}}$ is only estimated by the sparseness measure σ instead of being determined precisely by means of σ_τ. To provide an illustrative example, two nonnegative random sources $\mathbf{s}_1$ and $\mathbf{s}_2$ consisting of 1000 data points were generated with 90 % of the components of the first source signal and 80 % of the components of the second source signal being equal to zero. These two sources were normalized and then used to constitute the rows of the source matrix $\mathbf{S}$ (cf. Fig. 2). The matrix of observations $\mathbf{X}$ was obtained by mixing the sources with the following mixing matrix

$$\mathbf{A} = \begin{bmatrix} 5 & 1 \\ 6 & 1 \end{bmatrix} \tag{11}$$

Note that the first source signal is dominating in both mixtures according to

$$\begin{aligned} x_{1,i} &= a_{11}s_{1,i} + a_{12}s_{2,i} \\ x_{2,i} &= a_{21}s_{1,i} + a_{22}s_{2,i} \end{aligned} \tag{12}$$

Thus the following alternative factorization of the observation matrix $\mathbf{X}$ is feasible. First, the original mixing matrix $\mathbf{A}$ can be replaced by the pseudo

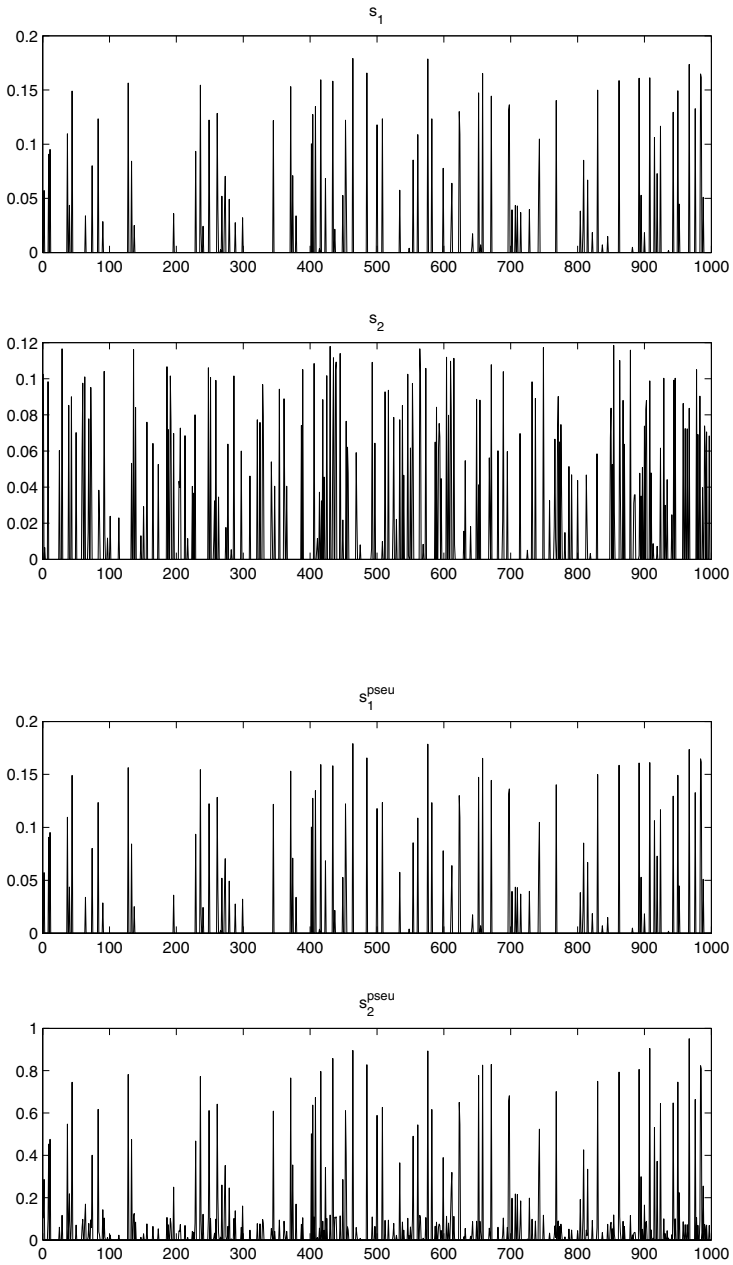


Fig. 2. Top: The original source signals $\mathbf{s}_1$ and $\mathbf{s}_2$. Bottom: The pseudo source signals $\mathbf{s}_1^{pseu}$ and $\mathbf{s}_2^{pseu}$. Even if the number of zero elements in $\mathbf{s}_2^{pseu}$ is lower than in the original source $\mathbf{s}_2$, the sparseness measure σ assigns to it a higher value than to the original source.

mixing matrix

$$\mathbf{A}^{pseu} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}. \tag{13}$$

	$\sigma(\mathbf{s})$	$\sigma_\tau(\mathbf{s})$
$\mathbf{s}_1$	0.76	0.90
$\mathbf{s}_2$	**0.62**	**0.80**
$\mathbf{s}_1^{pseu}$	0.76	0.90
$\mathbf{s}_2^{pseu}$	**0.69**	**0.72**

Table 1
The sparsenesses σ and σ_τ ($\tau = 0$) of the original and the pseudo sources. Note that contradicting the fact that the number of zero elements of the pseudo source signal $\mathbf{s}_2^{pseu}$ is lower than that of the original source signal $\mathbf{s}_2$, the sparseness reaches a higher value for the second pseudo source than for the second original source.

Correspondingly, the original source matrix $\mathbf{S}$ then has to be replaced by the matrix $\mathbf{S}^{pseu}$, the row vectors of which constitute the following pseudo source signals (see Fig. 2)

$$\begin{aligned} \mathbf{s}_1^{pseu} &= \mathbf{s}_1, \\ \mathbf{s}_2^{pseu} &= 5\mathbf{s}_1 + \mathbf{s}_2. \end{aligned} \tag{14}$$

Obviously, these matrices also factorize $\mathbf{X}$, i.e. $\mathbf{X} = \mathbf{A}^{pseu}\mathbf{S}^{pseu}$ still holds. But despite the fact that the number of zero elements in the second pseudo source signal $\mathbf{s}_2^{pseu}$ is about 8% lower than that of the original source signal $\mathbf{s}_2$, its sparseness estimate $\sigma(\mathbf{s}_2^{pseu})$ is higher than that of the original source $\sigma(\mathbf{s}_2)$ (cf. Tab. 1).

Hence, a smaller value of the cost function is obtained with the matrices $\mathbf{A}$ and $\mathbf{S}$ than with the matrices $\mathbf{A}^{pseu}$ and $\mathbf{S}^{pseu}$ if the sparseness measure σ_τ is used in (7). Accordingly, the matrices $\mathbf{A}$ and $\mathbf{S}$ would be recovered correctly up to the usual scaling and pertmutation indeterminacies inherent in the BSS model.

In contrast, the higher sparseness estimate $\sigma(\mathbf{s}_2^{pseu})$ of the second pseudo source signal compared to the original source signal $\mathbf{s}_2$ leads to a smaller value of the cost function with the matrices $\mathbf{A}^{pseu}$ and $\mathbf{S}^{pseu}$ than with the matrices $\mathbf{A}$ and $\mathbf{S}$. Accordingly, a sparse BSS algorithm which only estimates the sparseness by means of σ would fail to recover the original source signal matrix $\mathbf{S}$ and mixing matrix $\mathbf{A}$, respectively, except for scaling and pertmutation indeterminacies. Note that the same conclusion holds with other L_n - norm based sparsity criteria.

4 Genetic Algorithm based Optimization

4.1 Fitness function

As mentioned above, the cost or fitness function defined in Eq. (3) is discontinuous such that it cannot by optimized by techniques based on gradient descent. Furthermore, it possesses many local minima which suggests to use a Genetic Algorithm (GA) [28], [6] for its minimization.

GAs are stochastic global search and optimization methods inspired by natural biological evolution. The core of a GA is a population of potential solutions, named individuals, to a given optimization problem as well as a set of operators borrowed from natural genetics. At each generation of a GA, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain. Their reproduction is guided by the genetically motivated operators. This process leads to the evolution of individuals within the population which better solve the optimization problem than the individuals from which they were created. Finally, this process should lead to an optimal solution of the optimization problem even if many suboptimal solutions exist, i.e. if the target function to be optimized has many local minima.

For the minimization of the fitness function in Eq. (3) the M^2 elements of the solution matrix $\hat{\mathbf{A}}$ have to be determined. Taking advantage of the scaling indeterminacy inherent in the linear mixture model (2) we may assume that the columns of the original mixing matrix $\mathbf{A}$ are normalized such that the diagonal elements of $\mathbf{A}$ are $a_{ii} = 1 \; \forall \; i = 1, \ldots, M$. Accordingly, the N_{ind} individuals used in the GA to minimize (3) consists $M^2 - M$ parameters. However, these parameters (also called genes in the context of GAs) were not stored directly as real valued numbers but were transformed to binary strings of length b using Gray coding [11]. Hence, each individual consisted of $b(M^2 - M)$ binary numbers.

As the original mixing matrix is known to have only nonnegative entries, it seems self-evident to confine the genes to be nonnegative, too. However, we allow the genes to be negative throughout the optimization procedure as we have observed in our experiments that otherwise the GA often fails to find the global minimum of the fitness function.

In every generation of the GA, the fitness of each individual for the optimization task has to be computed in order to determine the number of offsprings that will be allowed to produce. These function values are not used directly as fitness values as otherwise the fittest individuals often produce too many offsprings such that the needed diversity in the population is lost and the

algorithm converges prematurely to a suboptimal solution. Hence, we use a linear scaling procedure to transform fitness function values to fitness values.

In order to compute the fitness function values for every individual, a matrix $\hat{\mathbf{A}}_-$ is generated with its off elements consisting of the genes as stored in the individual and with unit diagonal elements. As during the evaluation of the target function these matrices have to be inverted, the matrices which are singular to machine precision are replaced by nonsingular nonnegative random matrices (also with ones on their diagonal). Accordingly, the corresponding individuals in the population are adjusted.

Next, the matrices $\hat{\mathbf{A}} = \{a_{mn}\}_{1\le m\le M, 1\le n\le M}$ and $\hat{\mathbf{S}} = \{s_{mt}\}_{1\le m\le M, 1\le t\le T}$ are needed in order to evaluate the fitness function (3). For this purpose, the inverse $\hat{\mathbf{W}}_-$ of $\hat{\mathbf{A}}_-$ is computed and the matrices $\hat{\mathbf{S}}$ and $\hat{\mathbf{A}}$ are then obtained by setting the negative elements of the matrices $\hat{\mathbf{S}}_- = \{s_{mt,-}\}_{1\le m\le M, 1\le t\le T}$, $\hat{\mathbf{S}}_- = \hat{\mathbf{W}}_-\mathbf{X}$ and $\hat{\mathbf{A}}_- = \{a_{mn,-}\}_{1\le m\le M, 1\le n\le M}$, respectively, to zero:

$$\begin{aligned} \hat{a}_{mn} &= H(\hat{a}_{mn,-})a_{mn,-} \\ \hat{s}_{mt} &= H(\hat{s}_{mt,-})s_{mt,-} \end{aligned} \tag{15}$$

Here, H denotes the Heaviside step function [1]

$$H(x) = \begin{cases} 0 & x < 0 \\ 1 & x \ge 0 \end{cases}. \tag{16}$$

After inserting the matrices $\hat{\mathbf{S}}$ and $\hat{\mathbf{A}}$ into (3) the resulting fitness function value is assigned to the corresponding individual. The individuals are then arranged in an ascending order according to their fitness function values and their fitness values $F(p^{(i)})$, $i = 1, \ldots, N_{ind}$, are determined by

$$F(p^{(i)}) = 2 - \mu + 2(\mu - 1)\frac{p^{(i)} - 1}{N_{ind} - 1}, \tag{17}$$

where $p^{(i)}$ is the position of individual i in the ordered population. The scalar parameter μ, which is usually chosen to be between 1.1 and 2.0, denotes the selective pressure towards the fittest individuals.

Stochastic Universal Sampling (SUS) [28] is used to determine the absolute number of offsprings an individual may produce. Thereby, an arc R_i of length $F(p^{(i)})$ is assigned to the i-th individual, $i = 1, \ldots, N_{ind}$, on a circle of circumference $C = \sum_{i=1}^{N_{ind}} F(x^{(i)})$. Starting from a randomly selected position, $2N_{off}$ marker points are allocated on the circle, whereas the distance between two consecutive marker points is $C/2N_{off}$ and N_{off} is the total number of offsprings to be created. The i-th individual may then produce as many offsprings as there are marker points in its corresponding arc R_i on the circle.

The offsprings are created in a two step procedure. In the first step, two individuals, which are eligible for reproduction according to the SUS criterion, are chosen at random and are used to create a new individual by uniform crossover, i.e. each bit of the new individual is created by copying, each time with a probability of 50 %, the corresponding bit of the first or the second parent individual.

In the second step, called mutation, the actual offsprings are obtained by flipping with a low probability p_{mut} each bit of the new individuals. The role of mutation is often seen as providing a guarantee that the probability of searching any given parameter set will never be zero and acting as a safety net to recover good genetic material that may be lost through the action of selection and crossover.

The last step of the GA applied to each generation is the replacement of the parent individuals by their offsprings. We use an elitist reinsertion scheme meaning that a fixed number $N_{elitist}$ of the fittest individuals is deterministically allowed to propagate through successive generations. Hence, only the $N_{ind} - N_{elitist}$ less fittest parent individuals are replaced by their fittest offsprings which ensures that the best solution found so far remains in the population.

In order to keep the algorithm from converging prematurely we make use of the concept of multiple populations. Thereby, a number N_{pop} of populations, each consisting of $N_{subsize}$ individuals, are evolving independently in parallel and are only allowed to exchange their fittest individuals after every T_{mig}-th generation. Hence, as long as not all populations have converged to the same solution they will regain some diversity after every T_{mig}-th iteration step. We use the complete net structure scheme for the exchange of individuals which means that every population is exchanging a fraction μ_{mig} of its fittest individuals with all other populations.

4.3 Parallelization

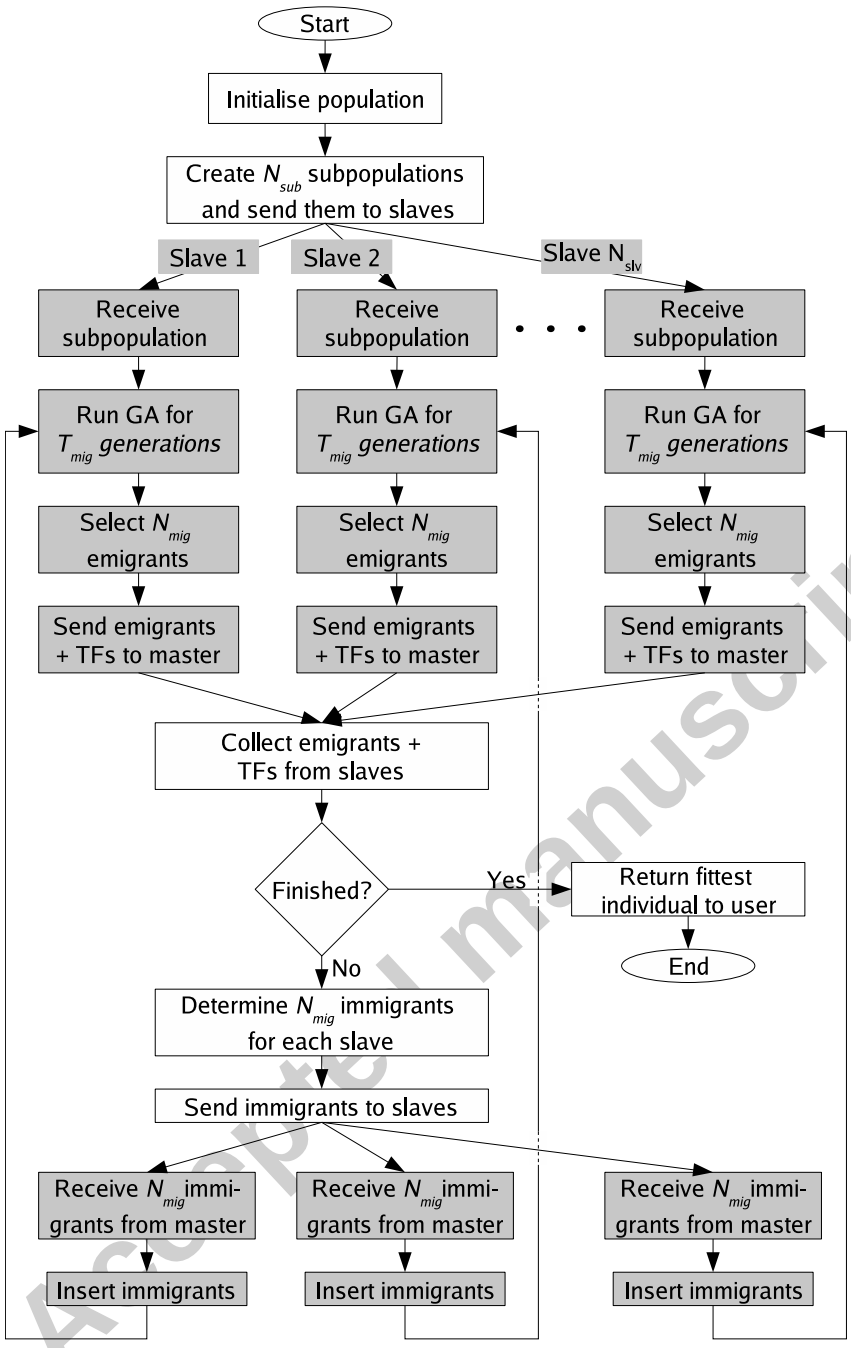


Fig. 3. Flowchart of the regional parallelization scheme used in the GA of sNMF. The actions of the individual slave processes are highlighted in gray, those of the master process in white.

A general problem of the proposed sNMF-GA algorithm is its tendency to converge prematurely to suboptimal solutions. This problem can be overcome to a large extent, however, if the regional parallelization model is used, i.e. if instead of one large population several smaller subpopulations are used; it

only exchanges individuals from time to time.

Another reason to parallelize the GA in sNMF-GA is that the evaluation of the fitness function may become very time consuming. The problem arises as in every evaluation of the fitness function the sparseness of the provided matrix $\hat{\mathbf{S}}$ has to be determined. This is done by comparing each entry of $\hat{\mathbf{S}}$ with a small, user defined threshold (cf. (7)), i.e. if $\hat{\mathbf{S}}$ is an $M \times T$ matrix, MT "if" statements have to be evaluated. As modern microarray chips are capable of detecting the expression of 50000 genes in parallel several hundreds of thousands of "if" comparisons need to be carried out per individual in real life applications.

Hence, the implementation of the sNMF-GA algorithm is designed such that it can be run on several computers in parallel. As the implementation of the sNMF-GA algorithm is fully written in the C programming language, the routines of the MPICH2 [16] implementation of the Message Passing Interface standard [43] for the communication between computers can be used advantageously.

As mentioned above, the regional scheme (see Fig. 3) is used for the parallelization of the GA. In order to determine the emigrants of each subpopulation, stochastic universal sampling is used for ranking, i.e. fitter emigrants have a higher chance to participate in the migration process than poorly performing ones. For the migration process the complete net structure scheme is used and the immigrants are reinserted into the individual subpopulation by replacing randomly selected individuals.

As the migrants contain only $M^2 - M$ (Gray encoded) numbers corresponding to the number of off-elements of the mixing matrix, the time needed for the data transfer between the master and the slave processes is negligible compared with the time the individual slaves need for the T_{mig} generations of their GAs. Hence, the execution time of the GA can be reduced by a factor close to $1/N_{nod}$ if a cluster consisting of N_{nod} nodes is available.

4.4 Algorithm repetitions

Despite the use of the mutation operator and multiple populations, the algorithm failed in many experiments to recover the source and mixing matrix after its first run. In order to keep the computational load of the algorithm bearable, this problem could not be overcome by simply increasing the number N_{ind} of individuals and N_{pop} of populations to arbitrarily large values. But satisfying results could still be obtained by applying the algorithm repeatedly. This approach reflects recent efforts towards more robust NMF algorithms using multilayer techniques [8].

The algorithm is normally provided with the observation matrix $\mathbf{X}$ in its first run which is then decomposed into first estimates of the source matrix $\hat{\mathbf{S}}^{(1)}$ and the mixing matrix $\hat{\mathbf{A}}^{(1)}$, i.e. $\mathbf{X} \approx \hat{\mathbf{A}}^{(1)}\hat{\mathbf{S}}^{(1)}$. In order to make use of the suboptimal results already achieved, during the next run the matrix $\hat{\mathbf{S}}^{(1)}$ is provided to the algorithm instead of the matrix $\mathbf{X}$. The matrix $\hat{\mathbf{S}}^{(1)}$ is then factorized into the matrices $\hat{\mathbf{A}}^{(2)}$ and $\hat{\mathbf{S}}^{(2)}$, which means that the matrix $\mathbf{X}$ can now be factorized as $\mathbf{X} \approx \hat{\mathbf{A}}^{(1)}\hat{\mathbf{A}}^{(2)}\hat{\mathbf{S}}^{(2)}$. This procedure is repeated N_{rep} times until the newly determined mixing matrix $\mathbf{A}^{(N_{rep})}$ differs only marginally from the identity matrix. With this procedure the final estimates of the mixing matrix $\hat{\mathbf{A}}$ and of the source matrix $\hat{\mathbf{S}}$ are determined as

$$\hat{\mathbf{A}} = \prod_{j=1}^{N_{rep}} \hat{\mathbf{A}}^{(j)} \tag{18}$$

and

$$\hat{\mathbf{S}} = \hat{\mathbf{S}}^{(N_{rep})}, \tag{19}$$

respectively, as the matrix $\mathbf{X}$ can be factorized as $\mathbf{X} = \prod_{j=1}^{N_{rep}} \hat{\mathbf{A}}^{(j)}\hat{\mathbf{S}}^{(N_{rep})}$.

5 sNMF Simulation Studies using Toy Data

5.1 Robustness to noise

The data obtained from microarray experiments is often corrupted by noise. Hence, the sensitivity to noise of the proposed sNMF algorithm is investigated in this section. Furthermore, the results are compared with those obtained by fastICA as this algorithm is most often used when microarray data are analyzed by means of BSS.

For the investigation three nonnegative random sources $\mathbf{s}_n$ with sparsenesses $\sigma_{\tau=0}$ of 0.9, 0.8 and 0.7, respectively were generated and were used to constitute the source matrix $\mathbf{S}$. In order to generate a matrix $\mathbf{X}$ of observations, this matrix of source signals was multiplied by the following mixing matrix

$$\mathbf{A} = \begin{bmatrix} 0.253490 & 0.706450 & 0.518140 \\ 0.644120 & 0.442550 & 0.227250 \\ 0.074437 & 0.267830 & 0.878450 \end{bmatrix} \tag{20}$$

Additionally, the matrix $\mathbf{N} \in \mathbf{R}^{3\times 1000}$ was created whose elements were randomly selected from a normal distribution. By means of this matrix 4 noisy versions $\mathbf{X}^{\{i\}}_{noise,-}$ of $\mathbf{X}$ were generated as follows

$$\mathbf{X}^{\{i\}}_{noise,-} = \mathbf{X} + \eta^{\{i\}}\mathbf{N}, \quad i = 1, \ldots, 4 \tag{21}$$

whereas the weighting factors $\eta^{\{i\}}$ were given by

$$\eta^{\{i\}} = 0.01i. \tag{22}$$

As the data obtained from microarray experiments are strictly nonnegative, the matrices $\mathbf{X}^{\{i\}}_{noise}$ were generated from the matrices $\mathbf{X}^{\{i\}}_{noise,-}$ by multiplying the negative entries of $\mathbf{X}^{\{i\}}_{noise,-}$ by -1, (see Fig. 4 and Fig. 5). The average signal to noise ratios ($\overline{SNR}^{\{i\}}$) between the rows of the noisy matrices $\mathbf{X}^{\{i\}}_{noise}$ and the original matrix $\mathbf{X}$ of observations are collected in Tab. 2.

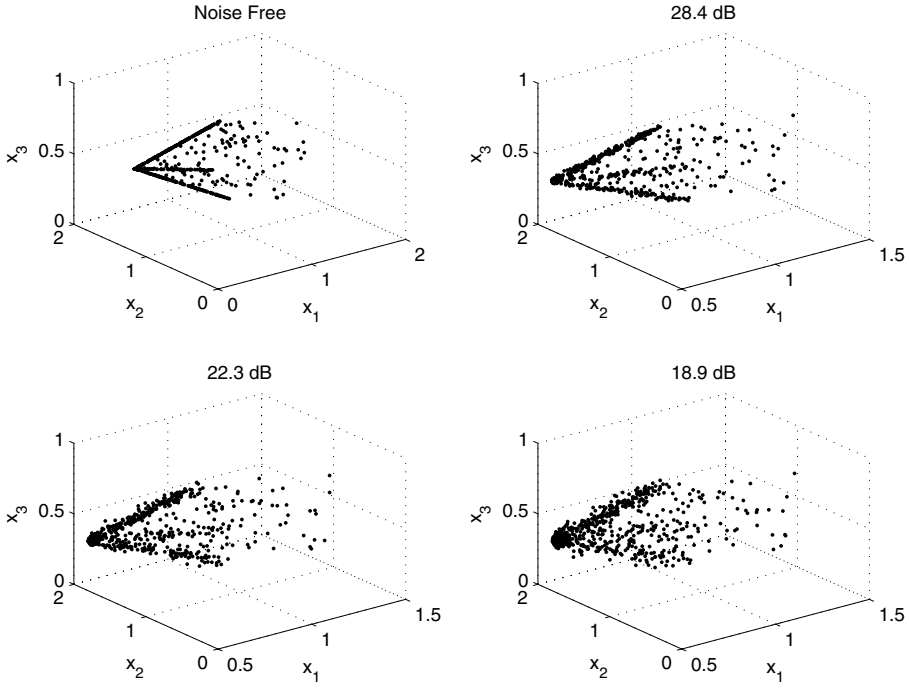


Fig. 4. The scatterplots of the original observation matrix (upper left) as well as of the noisy matrices $\mathbf{X}^{\{i\}}_{noise}$, $i = 1, 2, 3$, for which sNMF can recover the sources.

These observations $\mathbf{X}^{\{i\}}_{noise}$ were fed into the sNMF algorithm whereas the parameters as shown in Tab. 3 were used. The majority of these parameters are universal in the sense that they need not be adapted when the sNMF algorithm is applied to different data sets. An exception is the number of subpopulations N_{pop} which depends on the number of elements M^2 of the mixing matrix $\mathbf{A}$

Table 2
The $\overline{SNR}^{\{i\}}$ between the matrix $\mathbf{X}$ and the matrices $\mathbf{X}_{noise}^{\{i\}}$.

$\overline{SNR}^{\{1\}}$	$\overline{SNR}^{\{2\}}$	$\overline{SNR}^{\{3\}}$	$\overline{SNR}^{\{4\}}$
28.4 dB	22.3 dB	18.9 dB	16.2dB

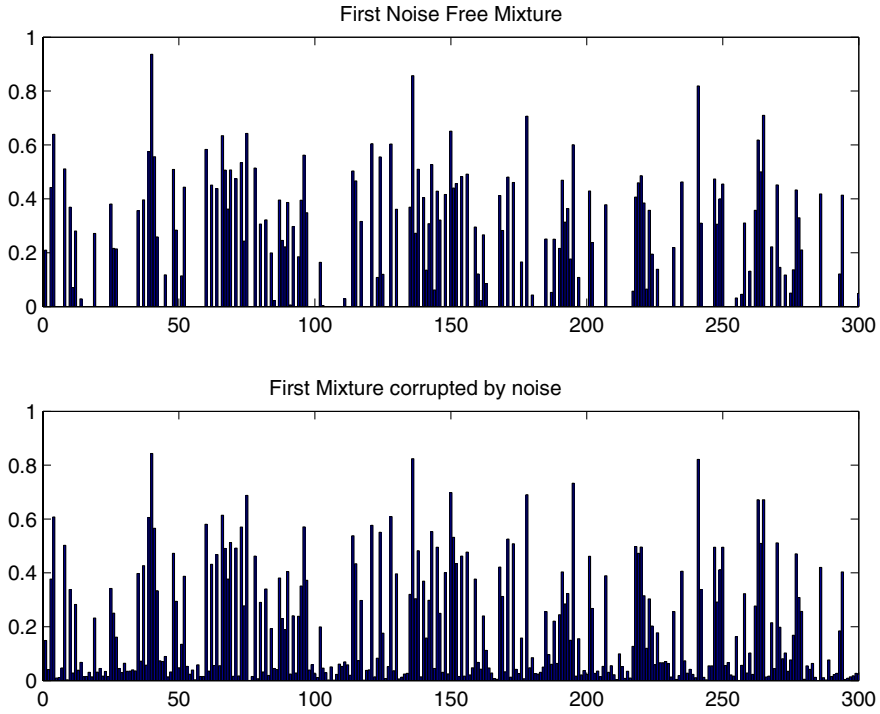


Fig. 5. The first row of the matrix $\mathbf{X}$ (top) and $\mathbf{X}_{noise}^{\{3\}}$ (bottom).

to be recovered. As a rule of thumb we suggest the following choice of N_{pop}

$$N_{pop}(M) = \begin{cases} 1 & \text{for } M = 2 \\ \lceil \frac{8}{9} M^2 \rceil & \text{for } M > 2 \end{cases} \tag{23}$$

whereas $\lceil x \rceil$ denotes the ceiling of x. Here the factor 8/9 originates from our observation that if $\mathbf{A}$ contains $M^2 = 9$ elements at least 8 subpopulations are needed to find the global minimum of the target function (3) reliably. Furthermore, both the number of migrations N_{mig} and the number of repetitions N_{rep} need to be adjusted according to the data set to be analyzed. Here, N_{mig} should be chosen sufficiently large such at the end of a single sNMF run all subpopulations are dominated by the same fittest individual. On the other hand N_{rep} should be set such that at least during the last three runs of the sNMF algorithm no new best solution is found.

Table 3
Parameters used by sNMF when applied to data sets $\mathbf{X}_{noise}^{\{i\}}$, $i = 1, \ldots, 4$

Parameter	Value
weighting factor λ	0.01
Parameter defining noise threshold τ	cf. Tab. 4
Number of bits in Gray coding b	60
Number of individuals of overall population N_{ind}	400
Number of subpopulations N_{pop}	8
Number of individuals per subpopulation $N_{subsize}$	50
Selection pressure μ	1.1
Number of offsprings N_{off}	50
Mutation probability p_{mut}	0.001
Generations between migrations T_{mig}	100
Total number of migrations N_{mig}	20
Fraction of individuals (per subpop.) allowed to migrate μ_{mig}	0.2
Number of elitist individuals $N_{elitist}$	1
Number of sNMF repetitions N_{rep}	5

Table 4
The values $\tau^{\{i\}}$ as used when the observations $\mathbf{X}_{noise}^{\{i\}}$ were analyzed by sNMF.

$\tau^{\{1\}}$	$\tau^{\{2\}}$	$\tau^{\{3\}}$	$\tau^{\{4\}}$
0.08	0.19	0.2	0.33

Eventually, the threshold τ in the sparseness measure σ_τ has to be chosen. For the simulation at hand different values $\tau^{\{i\}}$, $i = 1, 2, \ldots, 4$, were used depending on the noise level with which the data sets $\mathbf{X}^{\{i\}}$ were corrupted. These $\tau^{\{i\}}$'s were determined as follows: first, the matrices $\hat{\mathbf{S}}^{\{i\}}$, $i = 1, \ldots, 4$, were computed by multiplying the inverse of the original mixing matrix $\mathbf{A}$ by the matrices $\mathbf{X}_{noise}^{\{i\}}$, i.e.

$$\hat{\mathbf{S}}^{\{i\}} = \mathbf{A}^{-1}\mathbf{X}_{noise}^{\{i\}} \tag{24}$$

Next, the difference $\Delta\mathbf{S}^{\{i\}}$ between the original source matrix and $\hat{\mathbf{S}}^{\{i\}}$ was computed:

$$\Delta\mathbf{S}^{\{i\}} = \hat{\mathbf{S}}^{\{i\}} - \mathbf{S}. \tag{25}$$

This matrix $\Delta\mathbf{S}^{\{i\}}$ contains the noise which appears in the sources if the noisy mixture $\mathbf{X}^{\{i\}}_{noise}$ is fed into sNMF instead of the original mixture $\mathbf{X}$. Furthermore, the maximum element $\Delta s^{\{i\}}_{max}$ of $\Delta\mathbf{S}^{\{i\}}$ was determined. If this element was found in the q-th row of $\Delta\mathbf{S}^{\{i\}}$ the parameter $\tau^{\{i\}}$ was set to

$$\tau^{\{i\}} = \frac{\Delta s^{\{i\}}_{max}}{\hat{s}^{\{i\}}_{q,max}} \tag{26}$$

where $\hat{s}^{\{i\}}_{q,max}$ is the maximum value of the q-th row of $\hat{\mathbf{S}}^{\{i\}}$. Thus, $\tau^{\{i\}}$ should be sufficiently large such that the entries in $\hat{\mathbf{S}}^{\{i\}}$ which correspond to null entries in $\mathbf{S}$ are correctly detected by the sparseness measure $\sigma_{\tau^{\{i\}}}$.

After the $\tau^{\{i\}}$'s were determined the matrices $\mathbf{X}^{\{i\}}_{noise}$ were fed into sNMF and the source matrix $\hat{\mathbf{S}}^{\{i\}}$ as well as the mixing matrix $\hat{\mathbf{A}}^{\{i\}}$ were estimated. The quality of these estimates was evaluated using the following criteria:

(1) The cross talking error (CTE) between the original and the estimated mixing matrix was determined (cf. Fig. 7) according to [38]

$$CTE = \sum_{i=1}^{n}\left(\sum_{j=1}^{n}\frac{\|p_{ij}\|}{max_k\|p_{ik}\|} - 1\right) + \sum_{j=1}^{n}\left(\sum_{i=1}^{n}\frac{\|p_{ij}\|}{max_k\|p_{kj}\|} - 1\right) \tag{27}$$

where $\mathbf{P} = (p_{ij}) = \hat{\mathbf{A}}^{-1}\mathbf{A}$ and $\hat{\mathbf{A}}$ the calculated estimate of $\mathbf{A}$.

(2) The correlation coefficients CC_1 between the rows of the matrix $\hat{\mathbf{S}}^{\{i\}}$ and the original source matrix $\mathbf{S}$ were computed (cf. second row in Tab. 5 and Fig. 7).

(3) In sNMF, elements of the k-th row, $k = 1, 2, 3$, of the estimated source matrix $\hat{\mathbf{S}}^{\{i\}}$ which are smaller than $\tau^{\{i\}}s^{\{i\}}_{k,max}$ are treated as null elements (here $s^{\{i\}}_{k,max}$ denotes the maximum value of the k-th row of the matrix $\hat{\mathbf{S}}^{\{i\}}$). Hence, these elements were set to zero in $\hat{\mathbf{S}}^{\{i\}}$ and the correlation coefficients CC_2 between the so-obtained and the original source matrix $\mathbf{S}$ were determined (cf. third row in Tab. 5 and Fig. 7).

(4) For comparison, the data sets $\mathbf{X}^{\{i\}}_{noise}$ were also analyzed by fastICA. As before the results were evaluated by computing the cross-talk error ($CTE_{fastICA}$) between the original and the estimated mixing matrix. Furthermore, the correlation coefficients ($CC_{fastICA}$) between the original and the estimated source matrix have been determined (last row of Tab. 5, see also Fig. 7).

As can be seen in Fig. 7 the original mixing matrix $\mathbf{A}$ was well recovered by sNMF if the observation matrix was corrupted by low levels of noise (e.g. for data sets $X^{\{1\}}_{noise}$ and $X^{\{2\}}_{noise}$). For higher levels, however, $\mathbf{A}$ could hardly be recovered. This is in contrast to fastICA, which even recovered $\mathbf{A}$ well if

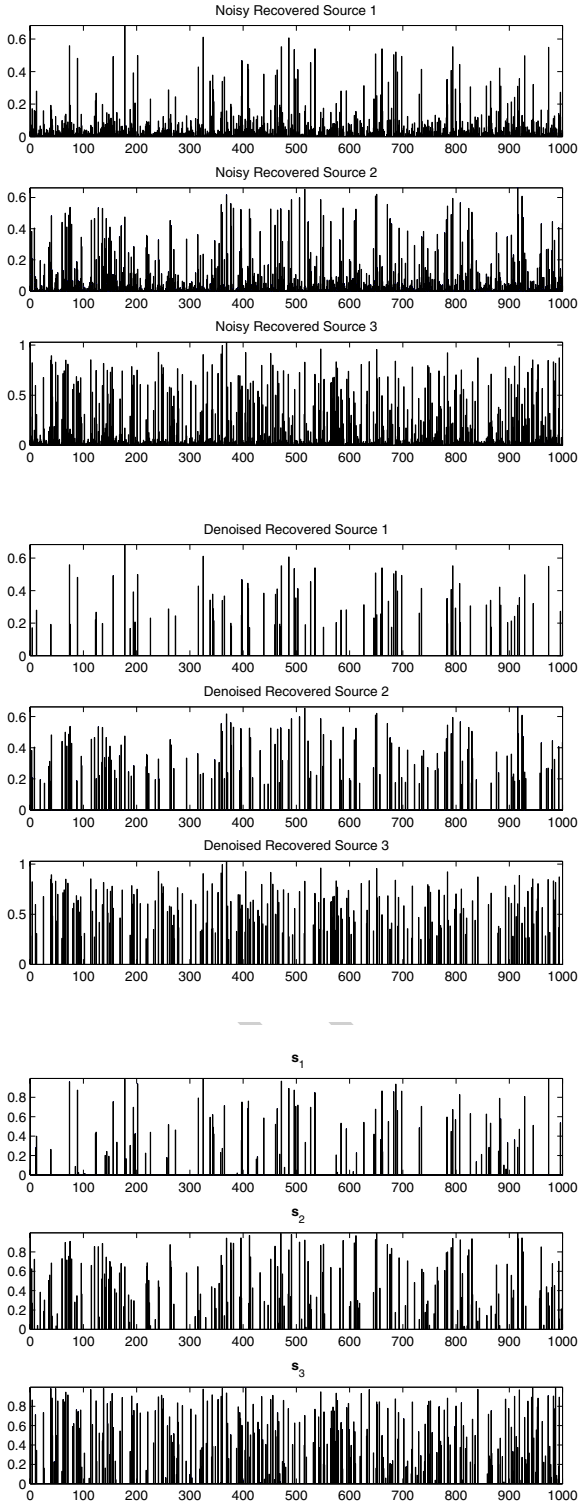


Fig. 6. Top: the noisy recovered sources $\hat{\mathbf{S}}^{\{3\}}$ as used in the computation of CC_1. Middle: the denoised sources $\hat{\mathbf{S}}^{\{3\}}$ as used in the computation of CC_2. Bottom: the original sources.

Table 5
Correlation coefficients CC_1, CC_2 and $CC_{fastICA}$ between the original and the estimated sources.

$X^{\{1\}}_{noise}$

	$\mathbf{s}_1$	$\mathbf{s}_2$	$\mathbf{s}_3$
CC_1	1.00	1.00	1.00
CC_2	1.00	1.00	1.00
$CC_{fastICA}$	1.00	1.00	1.00

$X^{\{2\}}_{noise}$

	$\mathbf{s}_1$	$\mathbf{s}_2$	$\mathbf{s}_3$
CC_1	0.95	0.98	0.99
CC_2	0.98	0.99	0.99
$CC_{fastICA}$	0.96	0.98	0.99

$X^{\{3\}}_{noise}$

	$\mathbf{s}_1$	$\mathbf{s}_2$	$\mathbf{s}_3$
CC_1	0.91	0.95	0.98
CC_2	0.95	0.97	0.98
$CC_{fastICA}$	0.96	0.94	0.99

$X^{\{4\}}_{noise}$

	$\mathbf{s}_1$	$\mathbf{s}_2$	$\mathbf{s}_3$
CC_1	0.75	0.87	0.97
CC_2	0.78	0.90	0.97
$CC_{fastICA}$	0.89	0.98	0.93

higher levels of noise were present.

Still, sNMF lead to better estimates of the recovered sources than fastICA if it was applied to the data sets $\mathbf{X}^{\{i\}}_{noise}$, $i = 1, 2, 3$ (see third row of Tab. 5 and Fig. 6). At first glance, this may seem surprising as the recovery of the sources by both sNMF and fastICA is solely based on the estimated mixing matrix. Hence, a poorer estimation of $\mathbf{A}$ by sNMF should also lead to poorer recoveries of the sources. However, recall that in sNMF any negative entries of the estimated sources are deliberately set to zero (cf. (15)). Furthermore note, that such negative entries only appear if the original mixing matrix $\hat{\mathbf{A}}$ was not recovered perfectly since the original sources are nonnegative. Thus, setting the negative entries in $\hat{\mathbf{S}}_-$ to zero compensates (at least to some extent) the deficiencies of the estimation of $\mathbf{A}$ and leads to the comparatively better recovered sources.

For data set $X^{\{4\}}_{noise}$ both sNMF and fastICA have problems in recovering the source $\mathbf{s}_1$.

In summary these simulations show that sNMF is capable of recovering the original sources and to some extent also the original mixing matrices if the observations are corrupted by noise of moderate level. For such noise levels, sNMF better recovers the sources than the fastICA. This is impressive as sNMF is partly based on counting null entries which easily get blurred in the presence of noise.

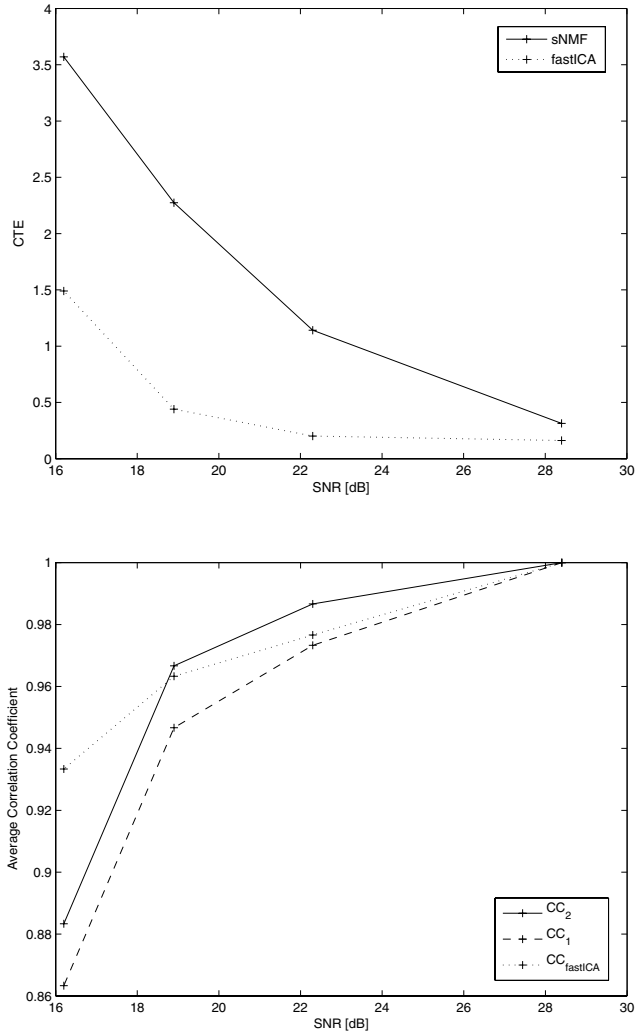


Fig. 7. Top: CTEs between original and estimated mixing matrix for the different noise levels of the observations. Bottom: the average correlation coefficients between the original and the recovered sources for both sNMF and fastICA. For SNRs larger than 18 dB sNMF leads to better results than fastICA. For smaller SNRs fastICA is superior but also fails to recover source $\mathbf{s}_1$ (cf. Tab. 5).

5.2 Recovery of correlated sources

In this section it is shown that the proposed method is capable of solving the BSS problem even if the underlying sources are correlated. This case is particularly interesting as the overwhelming majority of BSS algorithms only work if the underlying sources are uncorrelated or even statistically independent.

For the simulation, three sources $\mathbf{s}_i$, $i = 1, 2, 3$ were generated as follows. The first and the second source were generated as nonnegative random vectors where 90% and 80%, respectively, of the elements were randomly set to zero.

The third source was generated from the second source by adding a linear function, i.e.

$$\mathbf{s}_3(n) = \mathbf{s}_2(n) + 0.001(n-1), \tag{28}$$

where $\mathbf{s}_i(n)$ denotes the n-th element of the source $\mathbf{s}_i$ and $n = 1, \ldots, 1000$.

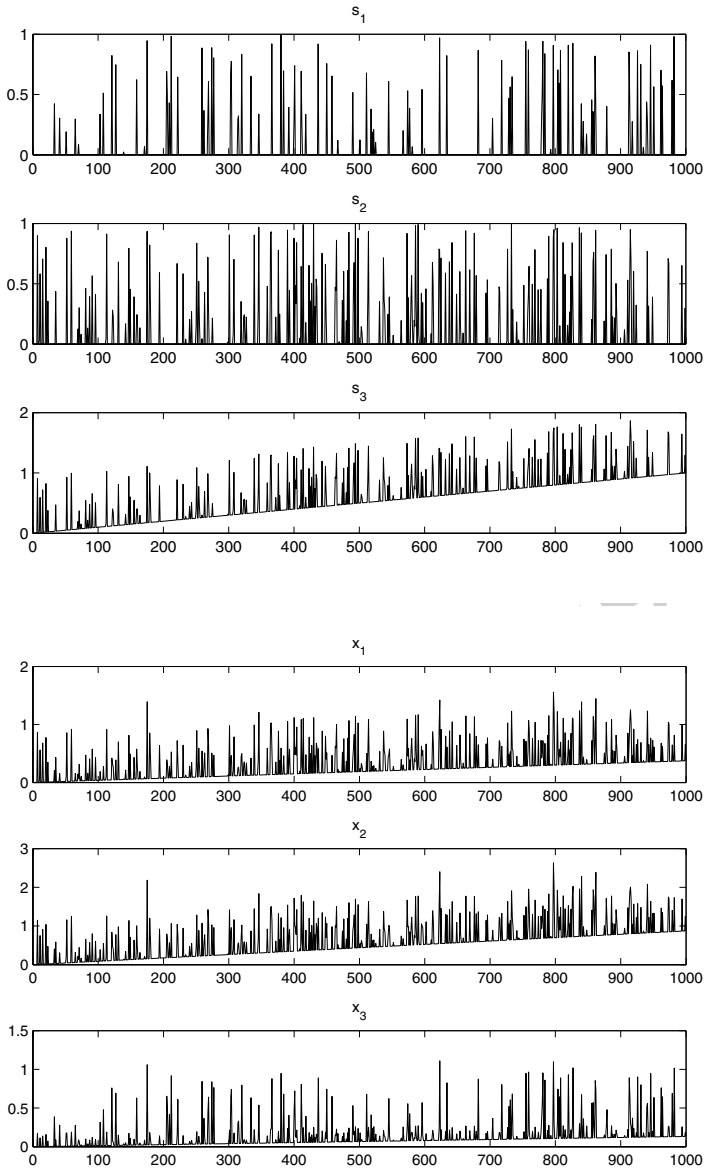


Fig. 8. Top: The original sources $\mathbf{s}_i$. Note that $\mathbf{s}_3$ was obtained from $\mathbf{s}_2$ by adding a linear function. Bottom: The rows $\mathbf{x}_i$ of the mixture matrix $\mathbf{X}$ as provided to the algorithms.

This procedure lead to a non-vanishing correlation coefficient of $c = 0.65$ between the second and the third source, while the sources $\mathbf{s}_1$ and $\mathbf{s}_2$ as well as $\mathbf{s}_1$ and $\mathbf{s}_3$ were uncorrelated. As before, these sources were used to constitute the source matrix $\mathbf{S}$.

The observation matrix $\mathbf{X}$ was generated by multiplying the source matrix $\mathbf{S}$ with the following mixing matrix

$$\mathbf{A} = \begin{bmatrix} 0.4554 & 0.5833 & 0.3739 \\ 0.8916 & 0.3988 & 0.8736 \\ 0.9042 & 0.0604 & 0.1326 \end{bmatrix}. \tag{29}$$

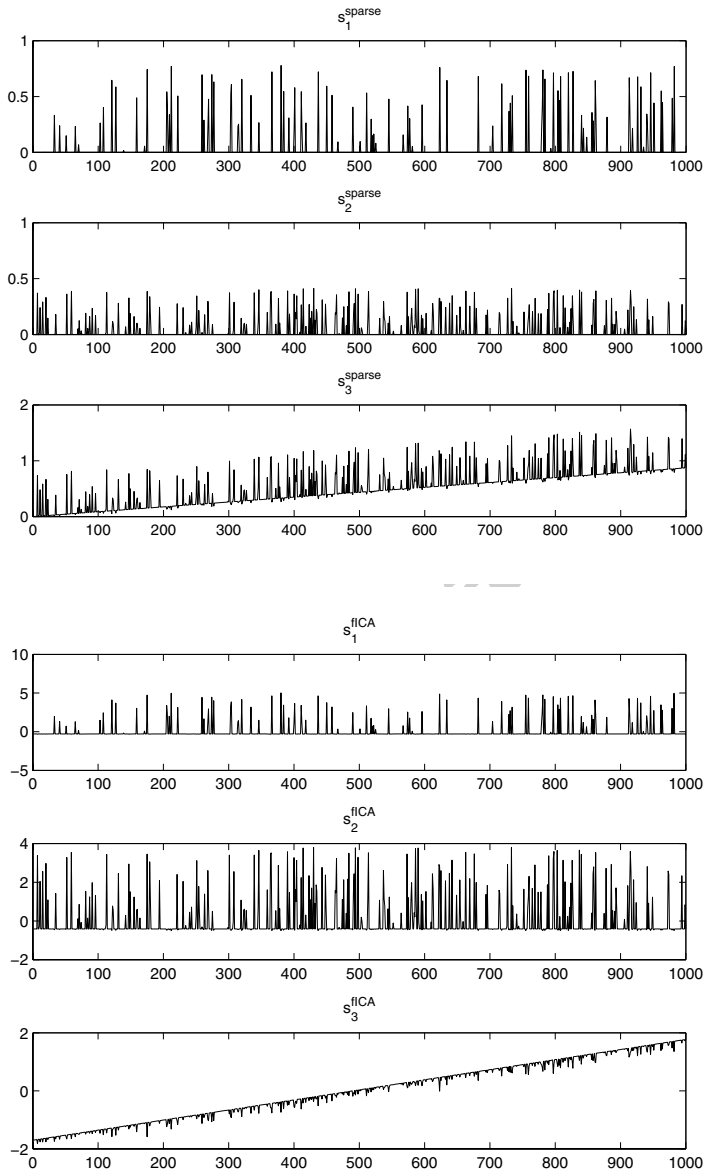


Fig. 9. Top: The estimates $\mathbf{s}_i^{sparse}$ of the sources as obtained by sNMF. Bottom: The estimates $\mathbf{s}_i^{fICA}$ of the sources as obtained by the fastICA algorithm. Note that fastICA fails to recover the third source.

Table 6
Results obtained by sNMF and fastICA algorithm. Displayed are the correlation coefficients c_i between the i-th original source and its corresponding estimate as well as the cross-talking error (CTE) between the estimated and the original mixing matrix.

	c_1	c_2	c_3	CTE
sNMF	1.00	1.00	1.00	0.39
fastICA	1.00	1.00	0.75	5.19

With sNMF the mixing matrix as as well as the sources were recovered almost perfectly as can be seen in Tab. 6 and Fig. 9.

In contrast, such a perfect recovery seems to be impossible by ICA based BSS. To show this, fastICA was used again to recover the sources $\mathbf{s}_i$, $i = 1, 2, 3$, and the mixing matrix $\mathbf{A}$. This algorithm also succeeded in recovering the sources $\mathbf{s}_1$ and $\mathbf{s}_2$ almost perfectly, but it failed to recover the third source $\mathbf{s}_3$ (cf. Fig. 9). Accordingly, the cross-talking error between the estimated and the original mixing matrix is more than five times higher than the CTE achieved with the sparse nonnegative BSS approach (cf. Tab. 6).

To see how easily fastICA fails if the underlying sources are correlated a second simulation was carried out. For this simulation three nonnegative random sources with sparsenesses $\sigma_{\tau=0}$ of 0.9, 0.8 and 0.7, respectively, were generated. In order to achieve correlation a certain number of randomly chosen elements of the second source were replaced by the corresponding elements of the first source. By means of this procedure nine variants $\mathbf{s}_2^{\{i\}}$, $i = 1, \ldots, 9$, of the second source were generated which had correlation coefficients with the first source of

$$CC^{\{i\}}(\mathbf{s}_1, \mathbf{s}_2^{\{i\}}) = 0.1 \cdot (i-1), \quad i = 1, \ldots, 9. \tag{30}$$

Each of these source matrices was multiplied by the mixing matrix

$$\mathbf{A} = \begin{bmatrix} 0.2535 & 0.7064 & 0.5181 \\ 0.6441 & 0.4426 & 0.2273 \\ 0.0744 & 0.2678 & 0.8785 \end{bmatrix} \tag{31}$$

such that nine observation matrices $\mathbf{X}^{\{i\}}$ were created.

As shown in Fig. 10 fastICA fails to recover the source and the mixing matrix satisfactorily if the correlation coefficient between the first and the second original source is larger than 0.3. For correlation coefficients larger than 0.7 the second source is not estimated at all, i.e. all three estimated sources are higher correlated with the first and the third original source than with the

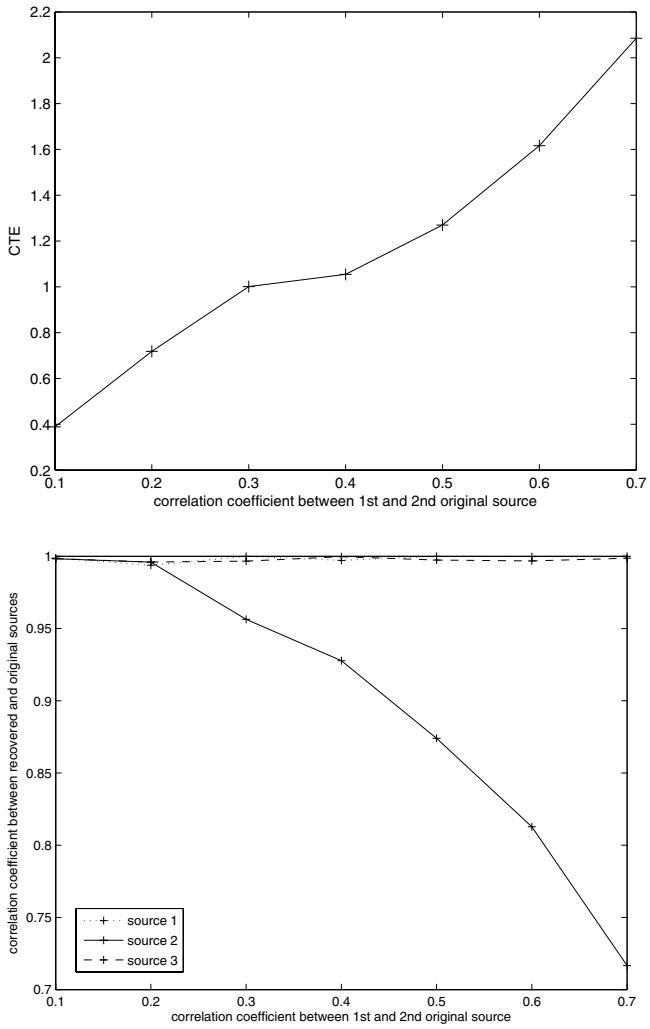


Fig. 10. Results obtained from fastICA. Top: CTEs between the original mixing matrix and the estimated mixing matrices. Bottom: correlation coefficients between the original sources and the estimated sources. fastICA leads to unsatisfying recoveries of the sources and the mixing matrix (i.e. correlation coefficient between 2nd original and recovered source < 0.95, $CTE > 1$) if the correlation coefficient between the first and the second original source is larger than 0.3.

second original source.

In contrast, sNMF successfully recovered the source matrices $\mathbf{S}^{\{i\}}$ and the mixing matrix $\mathbf{A}$ regardless of the correlation between the first and the second original source. The correlation coefficients between the original and the estimated sources were always close to 1.00 and the CTE between the original and estimated mixing matrix was about 0.04.

Hence, sNMF is capable of solving BSS problems in cases where other well established BSS methods like fastICA *principally* fail.

In the literature a multitude of different algorithms can be found which try to solve the BSS problem by exploring the geometry of the space of observations [10]. Most of these methods are based on uncorrelatedness or independence assumptions on the sources [30] and often require that the probability function p of the sources is symmetrical, i.e. $p(y) = p(-y)$ for y in $\mathbb{R}$ [31], [39]. Obviously, the latter condition can never be met with NMF resulting in a failure of these algorithms in this context. Furthermore, as shown in Sec. 5.2 sNMF can also solve the BSS problem if the underlying sources are correlated.

An intuitive approach to solve the nonnegative BSS problem geometrically is the following. First, note that as $\mathbf{A}$ and $\mathbf{S}$ are nonnegative all observations lie within a cone in the scatterplot of $\mathbf{X}$ whose cone lines are defined by the columns of $\mathbf{A}$. If furthermore $\mathbf{S}$ contains M columns $q_1, \ldots, q_M$ consisting of the unit vectors (or scalar multiples of them) spanning $\mathbb{R}^M$ then the corresponding columns of $\mathbf{X}$ lie on the cone lines, or, in other words, contain scalar multiples of the columns of $\mathbf{A}$ (see Fig. 11 top).

Hence, $\mathbf{A}$ can be recovered if the columns $q_1, \ldots, q_M$ of $\mathbf{X}$ can be detected. This detection is achieved by projecting $\mathbf{X}$ onto the standard simplex where the columns $q_1, \ldots, q_M$ of $\mathbf{X}$ constitute the edges of the resulting polytope. These edges can be determined automatically by computing the convex hull of projected data by means of, e.g., the QHull algorithm [3] [44] (see Fig. 11 bottom).

Once the estimate $\hat{\mathbf{A}}$ of the mixing matrix has been determined the corresponding source matrix $\hat{\mathbf{S}}$ can easily be computed as

$$\hat{\mathbf{S}} = \hat{\mathbf{A}}^{-1}\mathbf{X}. \tag{32}$$

However, problems arise when the data are corrupted by noise as in this case the convex hull is defined by more than just M edges and the original edges are blurred. To cope with this problem, Gruber et al. suggest to use Grassmann clustering [13] to determine the correct convex hull. In this approach, several subsets of the observations are formed at random and for each of these sets the convex hull is computed. The obtained facets of these convex hulls are then clustered in a projective space by an algorithm similar to k-means clustering [26]. Finally, the intersection points of the resulting centeroids are determined and are then used to estimate the mixing matrix $\mathbf{A}$.

Obviously, further difficulties appear if the columns of the source matrix $\mathbf{S}$ do not contain the M unit vectors or any vectors close to them. In such a case, the columns of $\mathbf{X}$ do not fully occupy the cone defined by $\mathbf{A}$ in the

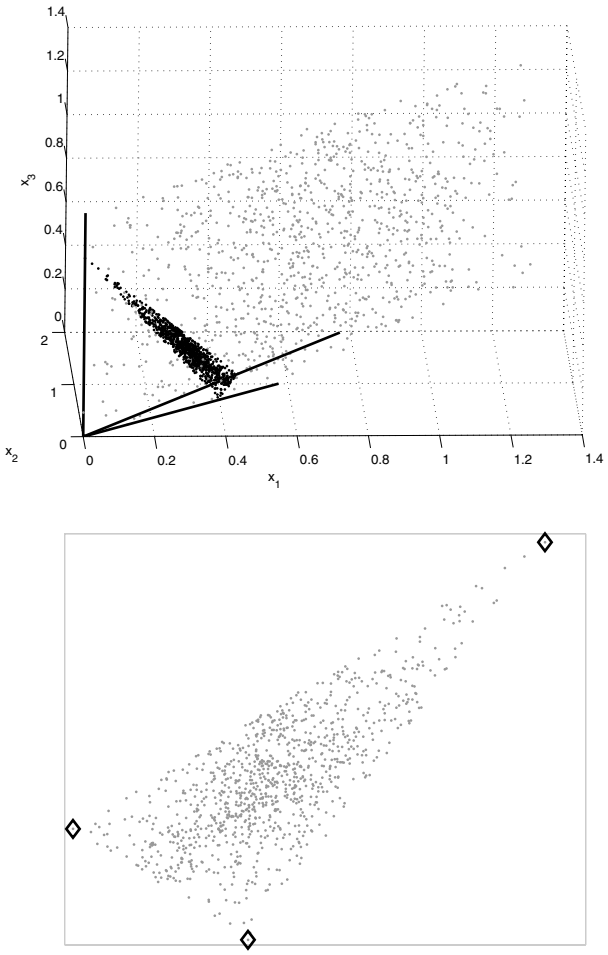


Fig. 11. Illustration of geometric NMF. Three random nonnegative sources, each consisting of 1000 datapoints, were generated and mixed by a random nonnegative 3×3 mixing matrix. The source matrix contained scalar multitudes of the 3 unit vectors of $\mathbb{R}^3$. Top: scatter plot of the mixtures (gray dots). The black lines indicate the cone spanned by the columns of the mixing matrix $\mathbf{A}$. The black points are the mixtures projected onto the standard simplex. Bottom: the observations $\mathbf{X}$ projected onto the standard simplex (i.e. the black dots of the top figure seen under a different angle). The edges of the obtained polytope (encircled by black diamonds) are those points of $\mathbf{X}$ which contain scalar multitudes of the columns of the mixing matrix $\mathbf{A}$. They can easily be detected by convex hull algorithms like Qhull [4, 44].

scatter plot of the observations. In particular, the areas close to the edges of the cone are not filled with any data points such that the polytope, which is observed after projection onto the standard simplex, does no longer contain the edges related with the columns of the mixing matrix $\mathbf{A}$ (points G, H and I in Fig. 12). Accordingly, convex hull algorithms like Qhull can no longer be used to recover the mixing matrix $\mathbf{A}$. Grassmann clustering based NMF may help in such scenarios, however, reasonable results can only be expected if the polytope on the simplex does at least to some extent adumbrate the edges

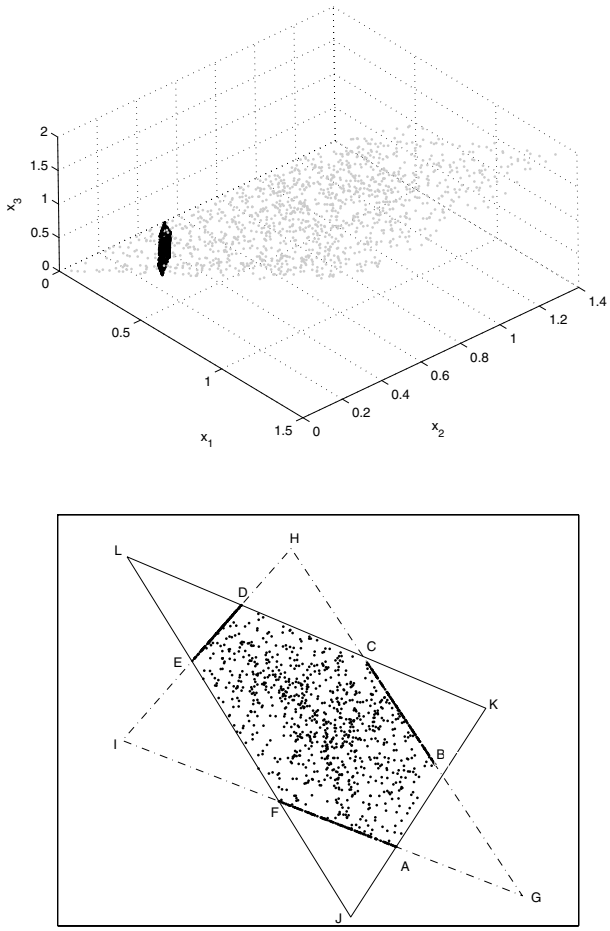


Fig. 12. Scatterplot and polytope obtained if $\mathbf{S}$ does not contain M unit vectors spanning $\mathbb{R}^M$ and if all nonzero elements of $\mathbf{S}$ are larger than 0.65. Top: scatterplot of the observations $\mathbf{X}$ (gray dots) and their projections onto the standard simplex (black dots). Bottom: blow-up of the datapoints projected onto the standard simplex (i.e. the black points in the figure at the top). The triangular GHI corresponds to the projection of the cone defined by the columns of $\mathbf{A}$ onto the standard simplex. The edges of this triangle are empty as no nonzero datapoints smaller than 0.65 appeared in the columns of $\mathbf{S}$ and only maximally one element per column was zero. Hence, all datapoints reside within the polytope $ABCDEF$ and the data could not only be confined by the cone defined by the columns of the original mixing matrix $\mathbf{A}$, but also by another matrix with columns related with the points JKL. Grassmann clustering based NMF spuriously detects the latter matrix while sNMF correctly identifies the original mixing matrix $\mathbf{A}$.

related with the columns of $\mathbf{A}$ (i.e. the points G, H and I in Fig. 12).

In other cases, the sparsity of the sources can be exploited to recover the mixing matrix $\mathbf{A}$. To demonstrate this, three nonnegative random sources consisting of 1290 data points were generated which had sparsenesses of 0.107

0.109 and 0.114, respectively. The sources were used to constitute the rows of the source matrix $\mathbf{S}$. The sources were created such that in each column of $\mathbf{S}$ only one null entry appeared. Furthermore, all nonvanishing elements of $\mathbf{S}$ were in the range $[0.65, 1.00]$. These sources were mixed by the random nonnegative mixing matrix

$$\mathbf{A} = \begin{bmatrix} 0.6109 & 0.7904 & 0.1368 \\ 0.4038 & 0.1487 & 0.8514 \\ 0.8587 & 0.5075 & 0.7062 \end{bmatrix}. \tag{33}$$

The scatterplot of the resulting mixtures $\mathbf{X}$ as well as the polytope obtained after projecting onto the standard simplex are depicted in Fig. 12.

As can be seen, the edges G, H and I corresponding to the columns of $\mathbf{A}$ are clearly cut off and all projected data points reside within the polytope $ABCDEF$. Inspection of this polytope without any knowledge about the original source and mixing matrix allows two conclusions: first, the points G, H, and I correspond to the original mixing matrix but the edges of the triangle GHI do not contain any data points because of the limitted range of the sources. Second, the points J, K, L also seem to be related with the columns of the original mixing matrix and the edges of the triangle JKL are again cut off because of the particular structure of the sources.

As the line segments $[DE]$, $[CB]$ and $[FA]$, which lie on the triangle GHI are shorter than the line segments $[AB]$, $[CD]$ and $[EF]$ on JKL, Grassmann clustering erroneously identifies the points J, K and L as those points which belong to the original source matrix.

In contrast, sNMF manages to recover the original mixing matrix by taking advantage of the sparseness of the sources. Again, all parameters are chosen as given in Tab. 3 except the Lagrange parameter which has to be set to $\lambda = 0.0002$. Most notably, both matrices, the one defined by the points J, K, and L as well as the other defined by the points G, H and I, lead to a nonnegative factorization of $\mathbf{X}$, but if the points G, H and I are used to reconstruct $\mathbf{A}$ the corresponding sources are more sparse. Hence, sNMF leads to a perfect recovery of the sources and the mixing matrix while the estimates obtained by Grassmann clustering hardly resemble the original data (see Tab. 7)

Table 7
Correlation coefficients C_i between the $i-th$ estimated and the corresponding original source as well as the CTE between the original and the estimated mixing matrix for both Grassmann clustering based NMF and sNMF. While sNMF perfectly recovers the sources and the mixing matrix Grassmann clustering based NMF fails to solve the given BSS problem.

	C_1	C_2	C_3	CTE
Grassmann NMF	0.70	-0.40	0.71	8.24
sNMF	1.00	1.00	1.00	0.62

6 sNMF Simulation Studies using Microarray Data

So far the performance of sNMF has only been investigated by means of artificial data. In contrast, this section deals with the sNMF analysis of microarray data. As before, the results will be compared with those achieved by fastICA.

The data sets to be analyzed were recorded during an investigation of *Pseudo-Xanthoma Elasticum* (PXE), an inherited connective tissue disorder characterized by progressive calcification and fragmentation of elastic fibers in the skin, the retina, and the cardiovascular system.

During the investigations $M = 8$ microarray experiments have been carried out. In the first and the second experiment the PXE fibroblasts were incubated in Bovine Serum Albumin (BSA) whereas the incubation time was three hours in the first and 24 hours in the second experiment. In the third experiment the PXE fibroblasts were incubated for three hours in an environment with a high concentration of the Transcription Growth Factor beta and in the fourth experiment the cells were incubated for 24 hours in an environment which was rich in Interleukin 1 beta. The same experiments were then repeated with a control group of normal fibroblasts.

In order to detect the the expression of the individual genes an Affymetrix HG-U133 plus 2.0 microarray chip was used. This *in-situ* chip is capable of detecting 54675 genes in parallel and makes use of the probe pair strategy. Hence, each measured expression value was accompanied by a detection call. If in all experiments the detection call of a particular gene was "absent" the gene was removed from all data sets. After this procedure only 10530 genes remained (see Fig. 13 and Fig. 14) in each data set.

These data sets were used to constitute the 8×10530 observation matrix $\mathbf{X}$ which was then decomposed into the matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$ by the proposed sNMF algorithm. For the genetic algorithm the number of sub-populations was increased to $N_{pop} = 56$ (thus the overall number of individuals N_{ind} increased to 2800), the total number of migrations to $N_{mig} = 25$ and the number of

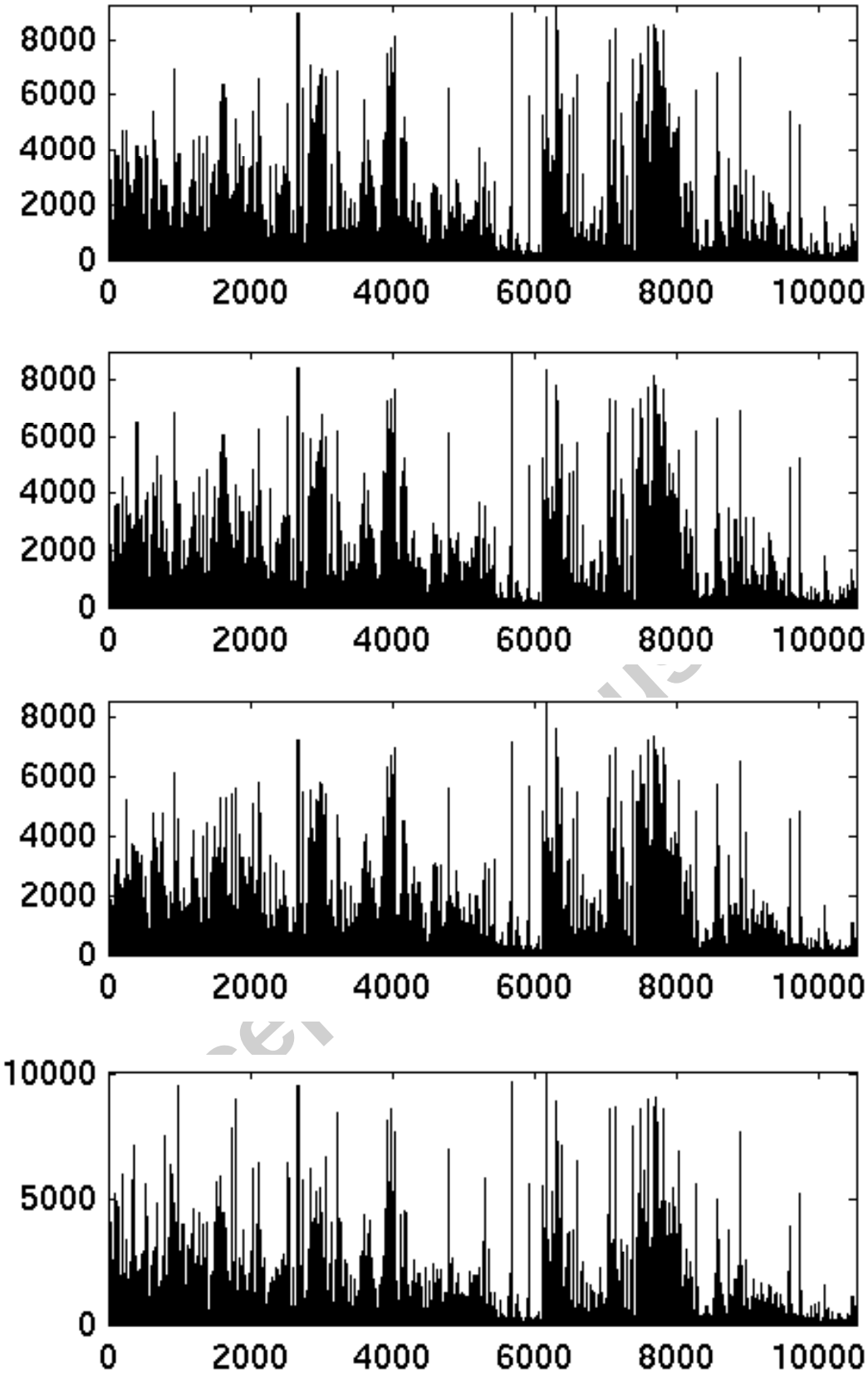


Fig. 13. The first four of the eight PXE data sets. Abscissa: genes. Ordinate: measured mRNA levels in arbitrary units.

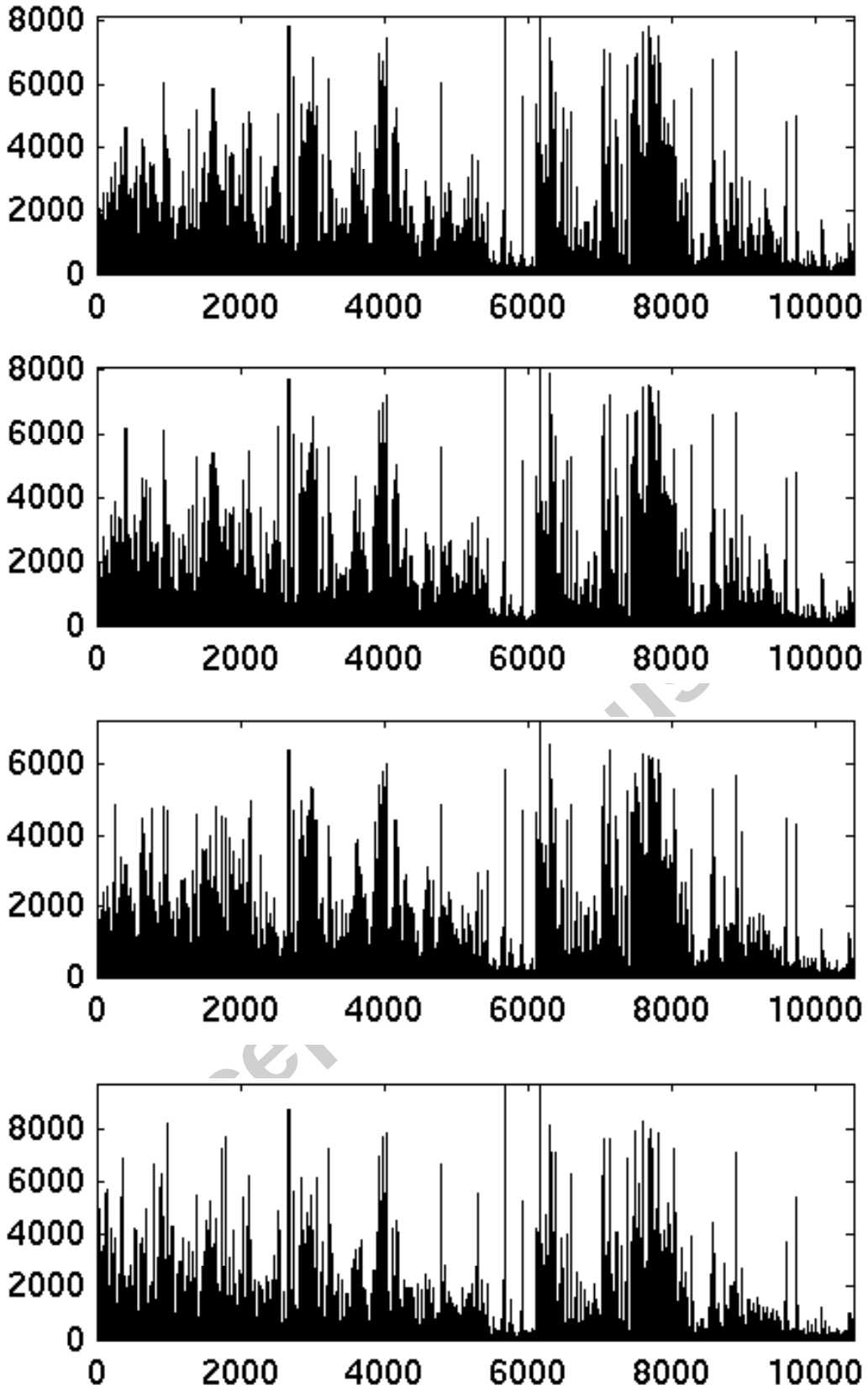


Fig. 14. The second four of the eight PXE data sets. Abscissa: genes. Ordinate: measured mRNA levels in arbitrary units.

algorithm repetitions to $N_{rep} = 8$. This number of repetitions seemed to be sufficient as it was observed that after the 5th of the $N_{rep} = 8$ repetitions of the algorithm the resulting matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$ did not change any further. For the sparseness measure σ_τ several values τ have been tried and it turned out that the best results were obtained if this parameter was set to 0.30. The values of the remaining parameters of the GA remained as listed in Tab. 3 above. Note that as the overall number of individuals was large ($N_{ind} = 2800$) the algorithm was run in parallel on a cluster of 28 computers and took 5 hours of computation time.

After the 8 repetitions the following estimate of the mixing matrix $\hat{\mathbf{A}}$ was obtained:

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1.9879 & 1.5171 & 1.6557 & 6.2249 & 0.8153 & 0.5777 & 2.1567 & 0.4715 \\ 2.0143 & 1.7504 & 0.9985 & 4.1203 & 1.0451 & 0.6593 & 1.5290 & 0.9634 \\ 3.5696 & 0.5377 & 1.0782 & 3.4697 & 0.2301 & 0.1243 & 1.0611 & 0.2914 \\ 1.9490 & 1.4348 & 0.8186 & 4.0773 & 0.5377 & 0.4795 & 0.9305 & 0.4145 \\ 0.7278 & 0.6451 & 0.5221 & 2.4486 & 1.3107 & 0.3698 & 0.4669 & 0.1856 \\ 0.9855 & 0.9942 & 0.6677 & 2.9247 & 1.1428 & 1.1557 & 0.6835 & 0.4150 \\ 2.5548 & 0.9079 & 2.1846 & 4.2192 & 1.0214 & 0.2973 & 1.3055 & 0.6379 \\ 1.9814 & 1.0337 & 0.8580 & 3.7670 & 0.5493 & 0.7602 & 0.6348 & 1.2538 \end{bmatrix} \tag{34}$$

Table 8
Sparsenesses $\sigma_{\tau=0}$ of the estimated sources.

Source	1	2	3	4	5	6	7	8
$\sigma_{\tau=0}$	0.996	0.971	0.983	0.995	0.984	0.954	0.989	0.985

The corresponding sources are depicted in Fig. 15 and Fig. 16. As the value of τ used by sNMF is rather large the obtained sources are very sparse as can be seen in Tab. 6. Hence, the sNMF algorithm uses only the expressions of a small portion of genes to reconstruct the mixing matrix $\mathbf{X}$. Here, it is assumed that these genes are the most important representatives of their corresponding cellular processes. This coincides with the notion often found in genetics which says that the most highly expressed genes are the most typical for a specific cellular process [21]. Thus, the high value of τ was not only used to deal with noise, but it was also used deliberately to obtain sources with only a few nonzero entries, i.e. only few active genes.

After the sparse NMF analysis each row of the matrix S should ideally consist of an expression pattern indicative of a specific biological process. It must

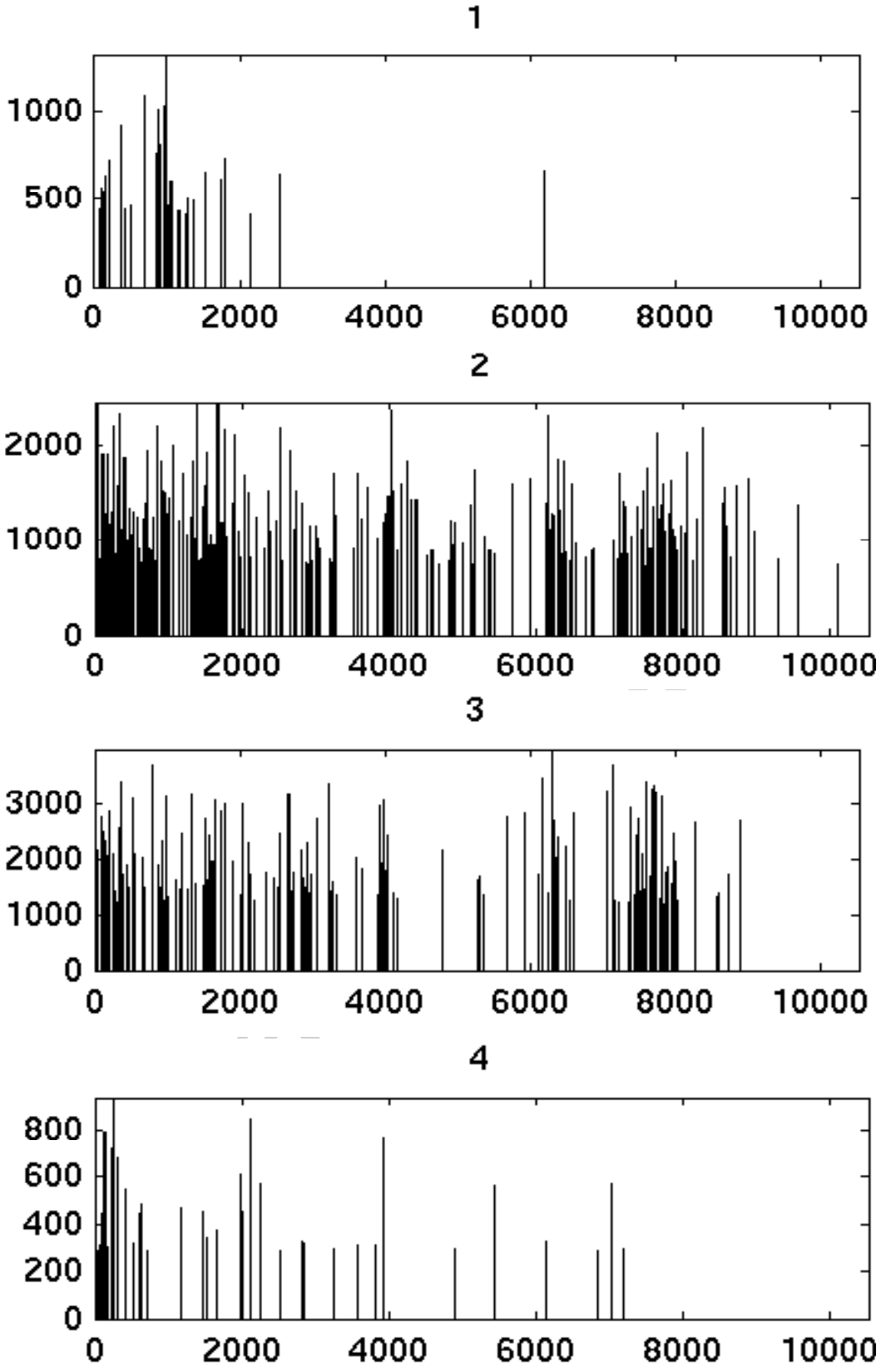


Fig. 15. The first four sources as obtained from sNMF. Abscissa: genes. Ordinate: mRNA levels in arbitrary units.

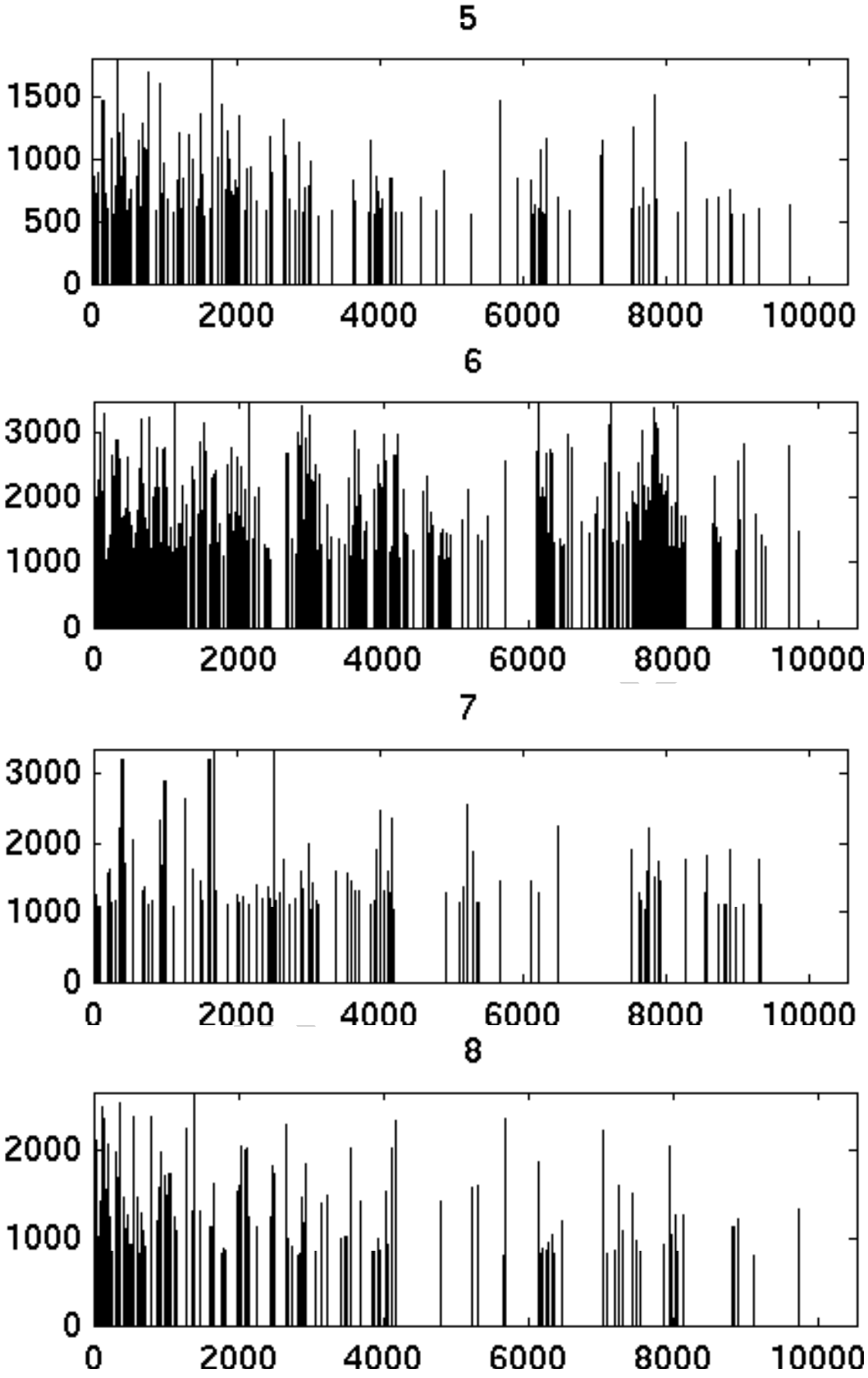


Fig. 16. The second four sources as obtained from sNMF. Abscissa: genes. Ordinate: mRNA levels in arbitrary units.

be noted, however, that at least one hundred of such processes are occurring simultaneously in a biological cell while the number of available observations and hence the number of estimated sources was only eight. Hence, each estimated source is expected to contain signatures from various cellular processes.

But despite these highly overcomplete settings the algorithm succeeded in grouping the majority of genes which are related with calcium ion binding (344 in total) and hence with the disease picture of PXE into the sixth estimated source (see Tab. 9). Furthermore, the calcium ion binding related genes in the sixth source seem to be specific for only one biological process as maximally 14 % of them could be found in any of the remaining sources. Note that for the assignment of genes to molecular functions the Gene Ontology database [9] was queried.

Table 9
Number of genes related with calcium ion binding (#(cib)) for each of the eight estimated sources obtained by sNMF. Only genes which are rated exclusively as having a calcium ion binding molecular function in the Gene Ontology [9] database were considered. Most genes related with calcium ion binding are clustered into source 6.

Source	1	2	3	4	5	**6**	7	8
#(cib)	0	13	7	0	4	**35**	9	6

In [25] the same PXE data set was analyzed by means of the fastICA algorithm. In this analysis the following procedure was carried out: first the data matrix $\mathbf{X}$ was fed into fastICA whereas the Gaussian nonlinearity was used. As can be seen in Fig. 17 and Fig. 18 the independent components extracted had both positive and negative entries. Note that this actually means that the cellular processes would need to produce negative amounts of mRNA. Furthermore, there are no null entries in the sources which would mean that all genes participate at all processes. Hence, in [25] a further postprocessing step was necessary. In this step two clusters were formed for each source. For this purpose the maximum (s_{max}) and the minimum expression level (s_{min}) found in the source was determined. The first cluster then consisted of all those genes which had expression levels higher than $0.26 s_{max}$ while the second consisted of the genes with expression levels lower than $0.26 s_{min}$. Finally, the molecular function of the genes in each cluster was determined by means of the Gene Ontology [9] database.

In this case, at most 22 genes related with calcium ion binding could be grouped into one cluster as can be seen in Tab. 10. Furthermore, other clusters (e.g. clusters 1, 2 and 13) also contained significant amounts of genes related with calcium ion binding. sNMF thus achieved a better grouping of calcium ion binding genes than fastICA. These results are rather preliminary, though. Further cooperation with biologists is necessary to investigate if the calcium

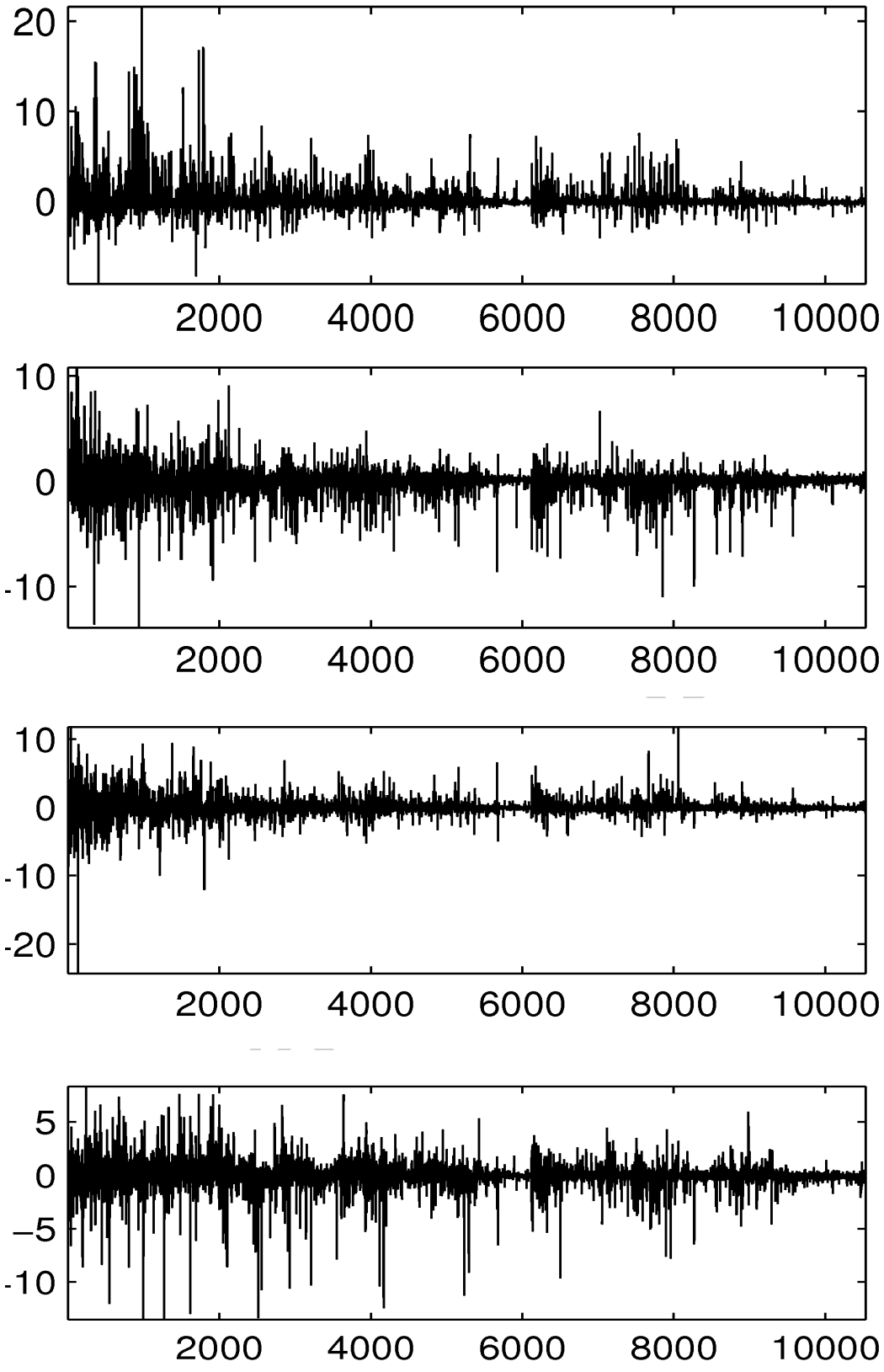


Fig. 17. The first four sources obtained when fastICA was applied to the PXE data.

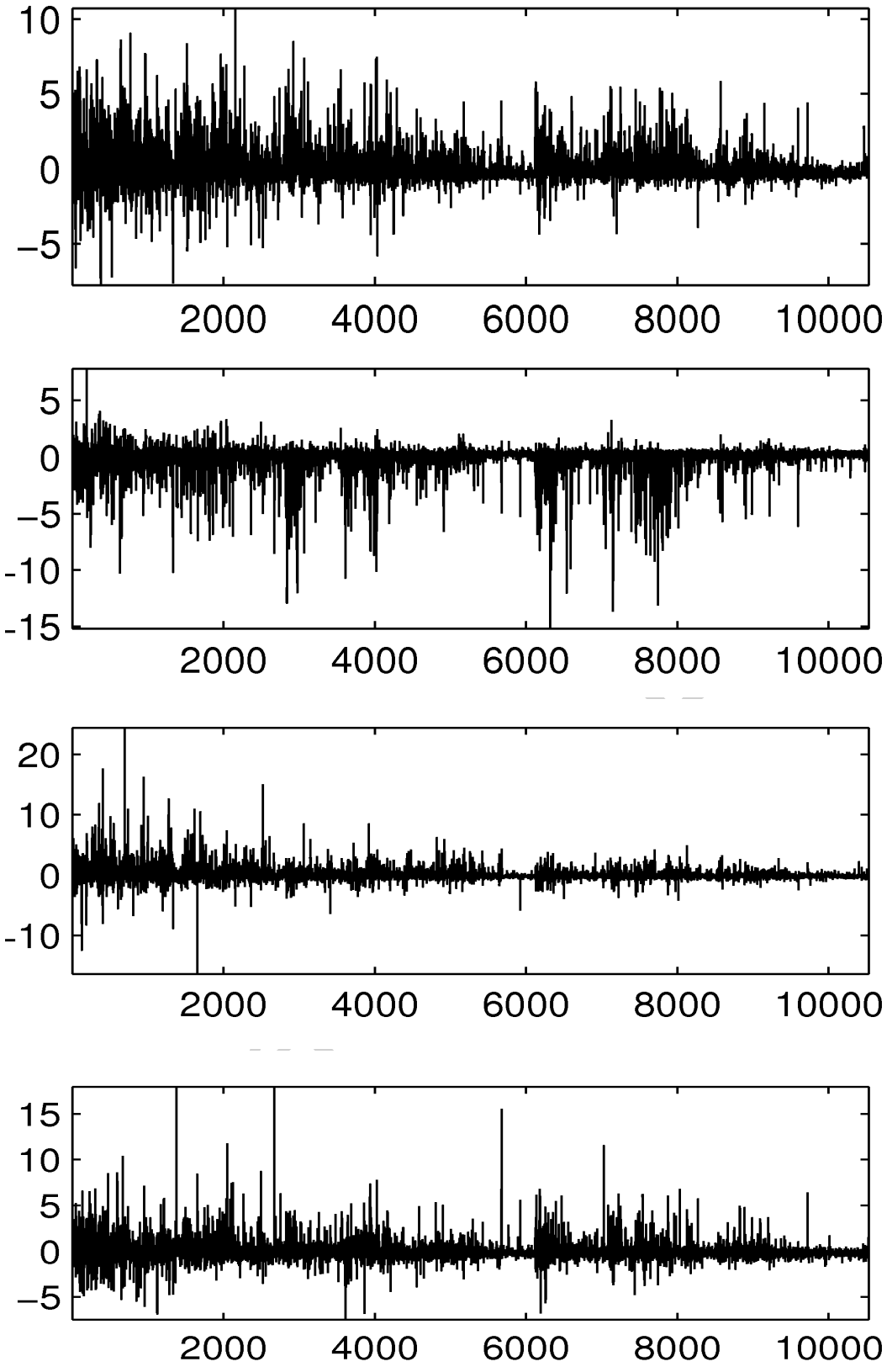


Fig. 18. The second four sources obtained when fastICA was applied to the PXE data.

Table 10
Number of genes related with calcium ion binding (#(cib)) for each of the 16 clusters formed during the analysis of the PXE data by fastICA. Only genes which are rated exclusively as having a calcium ion binding molecular function in the Gene Ontology [9] database were considered. Most genes related with calcium ion binding are clustered into source 10.

cluster	1	2	3	4	5	6	7	8	9	**10**	11	12	13	14	15	16
#(cib)	9	8	3	2	5	1	0	4	1	**22**	2	4	12	3	0	1

ion binding genes appearing in the sixth source as obtained by sNMF are indeed related with PXE.

7 Conclusions

In this paper we have presented a new approach to solve the BSS problem which is based on a nonnegativity constraint for the observations, the mixing matrix and the sources with an additional sparseness constraint concerning the encoding of the source signals. As the cost function considered has many local minima we have used a GA for its minimization. Further, we have discussed which sparseness measure is eligible for our approach and compared the devised algorithm to ICA and geometric BSS methods. In these comparisons we have demonstrated that our approach is also able to solve the BSS problem when the underlying sources are statistically dependent or correlated. Finally, we have applied the proposed algorithm to analyze real world microarray data. The obtained results suggest that our approach is better suited to analyze microarray data than ICA, however, further investigations in cooperation with biologists are necessary to confirm these preliminary results.

8 Acknowledgements

Financial support by Siemens AG, Corporate Technology, Munich (project *Biomarker*), the DFG (GRK 638: Nonlinearity and Nonequilibrium in Condensed Matter), the DAAD-GRICES Acções Integradas Luso - Alemãs (project GEVD-MP) and the DAAD Acciones Integradas Hispano - Alemanas (project Microarrays) is gratefully acknowledged. The PXE data set was kindly provided by PD Dr. Langmann, Institute of Clinical Chemistry (Director: Prof. Dr. G. Schmitz), University Clinic Regensburg.

References

[1] M. Abramowitz, I. A. Stegun (eds.), Handbook of Mathematical Formulas, Graphs and Mathematical Tables, 9th ed., Dover, New York, 1972.

[2] P. Baldi, W. Hatfield, DNA Microarrays and Gene Epression, Cambridge University Press, 2002.

[3] C. B. Barber, D. P. Dopkin, H. Huhdanpaa, The quickhull algorithm for convex hull, Technical report gcg53, The Geometry Center, University of Minesota, Minneapolis (1993).

[4] C. Barber, D. Dobkin, H. Huhdanpaa, The quickhull algorithm for convex hulls, ACM Trans. on Mathematical Software 22(4) (1996) 469 – 483.

[5] P. Chiapetta, M. C. Roubaud, B. Torrésani, Blind source separation and the analysis of microarray data, J. Comp. Biology 11 (2004) 1090–1109.

[6] A. Chipperfield, P. Fleming, H. Pohlheim, C. Fonseca, Genetic Algorithm Toolbox, University of Sheffield, 1994.

[7] A. Cichocki, S.-I. Amari, Adaptive Blind Signal and Image Processing, John Wiley & Sons, 2002.

[8] A. Cichocki, R. Zdunek, Multilayer nonnegative matrix factorization using projected gradient approaches, in: ICONIP 2006, Hong Kong, 2006.

[9] T. G. O. Consortium, Gene ontology: Tool for the unification of biology, Nature Genetics 25 (2000) 25–29.

[10] S. Fiori, S. ichi Amari, Editorial: Special issue on "geometric methods in neural networks and learning", Neurocomputing 67 (2005) 214–244.

[11] F. Gray, Pulse code communications, u.S. Patent 2632058 (March 1953).

[12] P. Gruber, K. Stadlthanner, M. Böhm, F. J. Theis, E. W. Lang, A. M. Tomé, A. R. Teixeira, C. G. Puntonet, J. M. G. Saéz, Denoising using local projective subspace methods, Neurocomputing 69 (2006) 1485–1501.

[13] P. Gruber, F. J. Theis, Grassmann clustering, Proceedings of the European Signal Processing Conference (EUSIPCO), Florence, Italy.

[14] M. Habl, C. Bauer, C. Ziegaus, E. W. Lang, Perspectives in Neuroscience: Artificial Neural Networks in Medicine and Biology, chap. Analyzing Brain Tumor Related EEG Signals With ICA Algorithms, Springer Publishers Berlin, 2000, pp. 131 – 136.

[15] P. O. Hoyer, Non-negative matrix factorization with sparseness constraints, J. Machine Learning Research 5 (2004) 1457 – 1469.

[16] http://www.mpi-forum.org/docs/mpi-20-html/mpi2 report.html, MPI-2: Extensions to the Message-Passing Interface, www.mpi-forum.org.

[17] A. Hyvärinen, J. Karhunen, E. Oja, Independent Componenet Analysis, John Wiley & Sons, 2001.

[18] A. Hyvärinen, E. Oja, P. Hoyer, J. Hurri, Image feature extraction by sparse coding and independent component analysis, in: Proc. ICPR 1998, 1998.

[19] I. R. Keck, F. J. Theis, P. Gruber, E. W. Lang, K. Specht, C. G. Puntonet, 3D spatial analysis of fMRI data on a word perception task, in: C. G. Puntonet, A. Prieto (eds.), Lecture Notes in Computer Science, LNCS 3195, Springer Verlag, Berlin, 2004.

[20] D. D. Lee, H. S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999) 788–791.

[21] S.-I. Lee, S. Batzoglou, Application of independent component analysis to microarrays, Genome Biology 4 (2003) R76.1–R76.21.

[22] D. D. Less, H. S. Seung, Algorithms for non-negative matrix factorization, in: Advances in Neural Information Processing 13 (NIPS'2000), MIT Press, Masachusetts, 2001.

[23] M. S. Lewicki, T. J. Sejnowski, Learning overcomplete representations, Neural Computation 12 (2000) 337–365.

[24] Y. Li, A. Cichocki, S. Amari, Analysis of sparse representation and blind source separation, Neural Computation 16 (2004) 1193–1234.

[25] D. Lutter, K. Stadlthanner, F. J. Theis, E. W. Lang, A. M. Tomé, B. Becker, T. Vogt, Analyzing gene expression profiles with ICA, Proceedings of the fourth IASTED International Conference on Biomedical Engineering, Innsbruck, AustriaAccepted.

[26] J. B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proc. 5th Berkely Symp. Math. Statistics and Probabiltiy, University of California Press, Berkely, 1967.

[27] S. Makeigh, A. Bell, T.-P. Jung, T. Sejnowski, Independent component analysis of electroencephalographic data, in: Advances in Neural Information Processing Systems 8 (NIPS'1995), MIT Press, Cambridge, Masachusetts, 1995.

[28] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer Verlag Berlin, 1999.

[29] B. A. Ohlshausen, D. J. Field, Natural image statistics and efficient coding, Network: Computation in Neural Systems 7 (1996) 333–339.

[30] M. Plumbley, Geometrical methods for non-negative ica: Manifolds, lie groups and toral subalgebras, Neurocomputing 67 (2005) 161–197, special Issue on Geometric Methods in Neural Networks and Learning.

[31] C. G. Puntonet, A. Prieto, Neural net approach for blind separation of sources based on geometric properties, Neurocomputing 18 (1998) 141–164.

[32] J. Quackenbush, Computational analysis of microarray data, Nature 2 (2001) 418 – 427.

[33] D. Ruderman, The statistics of natural images, Network: Computations in Neural Systems 5 (1994) 517–548.

[34] S. A. Saidi, C. M. Holland, D. P. Kreil, D. J. C. MacKay, D. S. Charnock-Jones, C. G. Print, S. K. Smith, Independent component analysis for gene arrays, Oncogene 23 (2004) 6677–6683.

[35] J. Särelä, H. Valpola, Denoising source separation, J. Machine Learning Research 6 (2005) 233–272.

[36] K. Stadlthanner, F. Theis, E. W. Lang, A. M. Tomé, W. Gronwald, H. R. Kalbitzer, A matrix pencil approach to the blind source separation of artifacts in 2D NMR spectra, Neural Information Processing - Letters and Reviews 1 (2003) 103 – 110.

[37] K. Stadlthanner, F. Theis, E. W. Lang, A. M. Tomé, W. Gronwald, H. R. Kalbitzer, Separation of water artefacts in 2D NOESY protein spectra using congruent matrix pencils, Neurocomputing 69 (2006) 497–522.

[38] H. H. Y. Sun-Ichi Amari, A. Cichocki, A new learning algorithm for blind signal separation, in: Adv. Neural Inform. Process. Systems (NIPS) 8, 1996.

[39] F. J. Theis, A. Jung, C. G. Puntonet, E. W. Lang, Linear geometric ica: Fundamentals and algorithms., Neural Computation 15 (2003) 419–439.

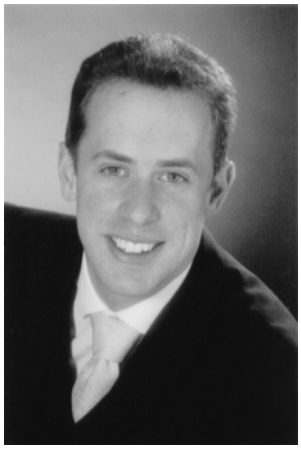
[40] A. M. Tomé, A. R. Teixeira, E. W. Lang, K. Stadlthanner, A. Rocha, Blind source separation using time-delayed signals, in: Proceedings of the International Joint Conference on Neural Networks, IJCNN'2004, vol. CD, Budapest, Hungary, 2004.

[41] R. Vigario, V. Jousmäki, M. Hämäläinen, R. Hari, E. Oja, Independent component analysis for identification of artifacts in magnetoencephalographic recordings, in: Advances in Neural Information Processing Systems 10 (NIPS'1997), MIT Press, Cambridge, Masachusetts, 1997.

[42] K. Yang, J. C. Rajapakse, ICA gives higher-order functional connectivity of brain, Neural Information Processing - Letters and Reviews 2 (2004) 27 – 32.

[43] The Message Passing Interface (MPI) standard, www.mpi-forum.org.

[44] QHull, http://www.qhull.org.



Kurt Stadlthanner received his Diploma and Phd degree in Physics from the University of Regensburg in 2003 and 2007, respectively. Furthermore he obtained a PhD degree in Computer Science from the University of Granada in 2006. He worked as a visiting researcher at the department of Architecture and Computer Technology (University of Granada, Spain). Currently, he is working as a research scientist at the Philips Research Laboratories in Aachen (Germany). His scientific interests are in the fields of biological data processing and analysis by means of blind source separation and support vector machines.

Fabian J. Theis obtained MSc degrees in Mathematics and Physics at the University of Regensburg in 2000. He also received a PhD degree in Physics from the same university in 2002 and a PhD in Computer Science from the University of Granada in 2003. In 2006 he was awarded the Heinz Maier-Leibnitz Prize by the German Research Foundation. He worked as visiting researcher at the department of Architecture and Computer Technology (University of Granada, Spain), at the RIKEN Brain Science Institute (Wako, Japan) and at FAMU-FSU (Florida State University, USA). Currently, he is working for the Max-Planck-Institute for Dynamics and Self-Organisation at Göttingen, Germany, where is heading a Junior Research Group at the Department of Nonlinear Dynamics. His research interests include biostatistics and statistical machine learning with special focus on statistical signal processing using information-theoretic methods for analyzing large-scale biomedical data sets.

Elmar W. Lang received his physics diploma with excellent grade in 1977 and his PhD in physics (summa cum laude) in 1980 and habilitated in Biophysics in 1988 at the University of Regensburg. He is apl. Professor of Biophysics at the University of Regensburg. He is currently associate editor of Neurocomputing and Neural Information Processing - Letters and Reviews. His current research interests focus mainly on machine learning and include biomedical signal processing, independent component analysis and blind source separation, neural networks for classification and pattern recognition as well as stochastic process limits in queuing applications.

Ana M. Tomé received her PhD in Electrical Engineering from University of Aveiro in 1990. Currently she is Associate Professor of Electrical Engineering in the Department of Electronics and Telecomunications/ IEETA of the University of Aveiro where she teaches courses on Digital Signal Processing for Electronics and Computer Engineering Diploms. Her research interests include Digital and Statistical Signal Processing, Independent Component Analysis and Blind Source Separation as well as Classification and Pattern Recognition Applications of Neural Networks.

Carlos G. Puntonet received a B.Sc. degree in 1982, an M.Sc. degree in 1986 and his Ph.D. degree in 1994, all from the University of Granada, Spain. These degrees are in electronics physics. Currently, he is an Associate Professor at the "Departamento de Arquitectura y TecnologÃa de Computadores" at the University of Granada. His research interests lie in the fields of signal processing, linear and nonlinear independent component analysis and blind separation of sources, artificial neural networks and optimization methods.

J.M. Górriz received the B.Sc. in Physics and Electronic Engineering from the University of Granada, Spain and the Ph.D. from the Universities of Cádiz and Granada, Spain in 2000, 2001, 2003 and 2006 respectively. He is currently an Assistant Professor at the University of Granada. His present interests lie in the field of statistical signal processing and its application to speech and image processing.