Blind separation of a large number of sparse sources

C. Kervazo, Jerome Bobin, C. Chenot

Blind separation of a large number of sparse sources

C. Kervazo*, J.Bobin, C. Chenot

IRFU, CEA, Université Paris-Saclay, F91191 Gif-sur-Yvette, France

Abstract

Blind Source Separation (BSS) is one of the major tools to analyze multi-spectral data with applications that range from astronomical to biomedical signal processing. Nevertheless, most BSS methods fail when the number of sources becomes *large*, typically exceeding a few tens. Since the ability to estimate large number of sources is paramount in a very wide range of applications, we introduce a new algorithm, coined block-Generalized Morphological Component Analysis (bGMCA) to specifically tackle sparse BSS problems when large number of sources need to be estimated. Sparse BSS being a challenging nonconvex inverse problem in nature, the role played by the algorithmic strategy is central, especially when many sources have to be estimated. For that purpose, the bGMCA algorithm builds upon block-coordinate descent with intermediate size blocks. Numerical experiments are provided that show the robustness of the bGMCA algorithm when the sources are numerous. Comparisons have been carried out on realistic simulations of spectroscopic data.

*Corresponding author

URL: `christophe.kervazo@cea.fr` (C. Kervazo)

Keywords: Blind source separation, sparse representations,
block-coordinate optimization strategies, matrix factorization

1. Introduction

Problem statement

Blind source separation (BSS) is the major analysis tool to retrieve meaningful information from multichannel data. It has been particularly successful in a very wide range of signal processing applications ranging from astrophysics [1] to spectroscopic data in medicine [2] or nuclear physics [3], to name only a few. In this framework, the observations $\{x_i\}_{i=1,\ldots,m}$ are modeled as a linear combination of n unknown elementary sources $\{s_j\}_{j=1,\ldots,n}$: $x_i = \sum_{j=1}^{n} a_{ij} s_j + z_i$. The coefficients a_{ij} are measuring the contribution of the j-th source to the observation x_i, while z_i is modeling an additive noise as well as model imperfections. Each datum x_i and source s_j is supposed to have t entries. This problem can be readily recast in a matrix formulation:

$$\mathbf{X} = \mathbf{AS} + \mathbf{N} \tag{1}$$

where $\mathbf{X}$ is a matrix composed of the m row observations and t columns corresponding to the entries (or samples), the mixing matrix $\mathbf{A}$ is built from the $\{a_{ij}\}_{i=1,\ldots,m,j=1,\ldots,n}$ coefficients and $\mathbf{S}$ is a $n \times t$ matrix containing the sources. Using this formulation, the goal of BSS is to estimate the unknown matrices $\mathbf{A}$ and $\mathbf{S}$ from the sole knowledge of $\mathbf{X}$.

Blind source separation methods

It is well-known that BSS is an ill-posed inverse problem, which requires additional prior information on either $\mathbf{A}$ or $\mathbf{S}$ to be tackled [4]. Making BSS

a better-posed problem is performed by promoting some discriminant information or diversity among the sources. A first family of standard techniques, such as Independent Component Analysis (ICA), assumes that the sources are statistically independent [4].

In this study, we will specifically focus on the family of algorithms dealing with the case of sparse BSS problems (*i.e.* where the sources are assumed to be sparse), which have attracted a lot of interest during the last two decades [5, 6, 7]. Sparse BSS has mainly been motivated by the success of sparse signal modeling for solving very large classes of inverse problems [8]. The Generalized Morphological Component Analysis (GMCA) algorithm [1, 9] builds upon the concept of morphological diversity to disentangle sources that are assumed to be sparsely distributed in a given dictionary. The morphological diversity property states that sources with different morphologies are unlikely to have similar large value coefficients. This is the case of sparse and independently distributed sources, with high probability. In the framework of Independent Component Analysis (ICA), Efficient FastICA (EFICA) [10] is a FastICA-based algorithm that is especially adapted to retrieve sources with generalized Gaussian distributions, which includes sparse sources. In the seminal paper [11], the author also proposed a Newton-like method for ICA called Relative Newton Algorithm (RNA), which uses quasi-maximum likelihood estimation to estimate sparse sources. A final family of algorithms builds on the special case where it is known that $\mathbf{A}$ and $\mathbf{S}$ are furthermore non-negative, which is often the case on real world data [12].

However, the performances of most of these methods decline when the number of sources n becomes large. As an illustration, Fig. 1 shows the evolution

of the mixing matrix criterion (cf. sec. 3.1, [9]) as a function of the number of sources for various BSS methods. This experiment illustrates that most methods do not perform correctly in the "large-scale"regime. In this case, the main source of deterioration is very likely related to the non-convex nature of BSS. Indeed, for a fixed number of samples t, an increasing number of sources n will make these algorithms more prone to be trapped in spurious local minima, which tends to hinder the applicability of BSS on practical issues with a large n. Consequently, the optimization strategy has a huge impact on the separation performances.

Contribution

In a large number of applications such as astronomical [1] or biomedical signals [2], designing BSS methods that are tailored to precisely retrieve a large number of sources is of paramount importance. For that purpose, the goal of this article is to introduce a novel algorithm dubbed bGMCA (block-Generalized Morphological Component Analysis) to specifically tackle sparse BSS problems when a large number of sources need to be estimated. In this setting, which we will later call the *large*-scale regime, the algorithmic strategy has a huge impact on the separation quality since BSS requires solving highy challenging non-convex problems. For that purpose, the proposed bGMCA algorithm builds upon the sparse modeling of the sources, as well as an efficient minimization scheme based on block-coordinate descent. In contrast to state-of-the art methods [11, 9, 13, 12], we show that making profit of block-based minimization with intermediate block sizes allows the bGMCA to dramatically enhance the separation performances, particularly when the number of sources to be estimated becomes large. Comparisons with the

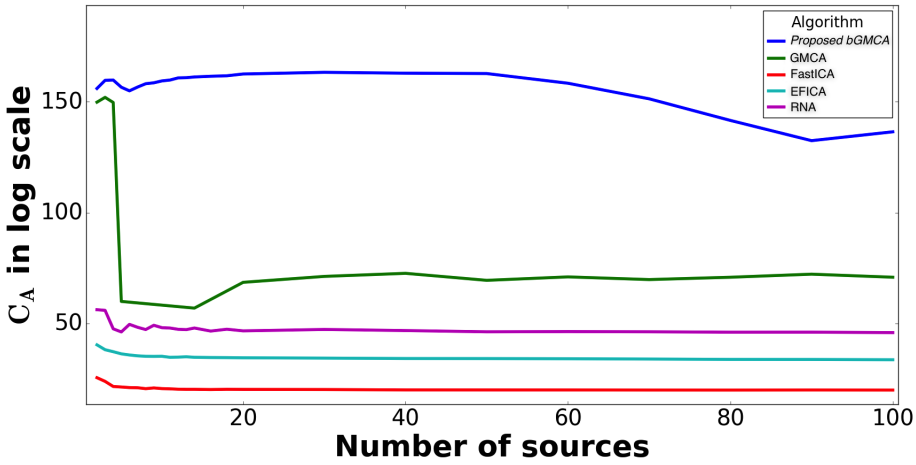


Figure 1: Evolution of the mixing matrix criterion (whose computation is detailed in sec. 3.1) of four standard BSS algorithms for an increasing n. For comparison, the results of the proposed $bGMCA$ algorithm is presented, showing that its use allows for the good results of GMCA for low n (around 160 dB for $n = 3$) to persist for $n < 50$ and to stay much better than GMCA for $n > 50$. The experiment was conducted using exactly sparse sources $\mathbf{S}$, with 10% non-zero coefficients, the other coefficients having a Gaussian amplitude. The mixing matrix $\mathbf{A}$ was taken to be orthogonal. Both $\mathbf{A}$ and $\mathbf{S}$ were generated randomly, the experiments being done 25 times and the median used to draw the figure.

state-of-the art methods have been carried out on various simulation scenarios. The last part of the article will show the flexibility of $bGMCA$, with an application to sparse and non-negative BSS in the context of spectroscopy.

2. Optimization problem and *bGMCA*

2.1. General problem

Sparse BSS [1, 9] aims to estimate the mixing matrix $\mathbf{A}$ and the sources $\mathbf{S}$ by minimizing a penalized least-squares of the form:

$$\min_{\mathbf{A},\mathbf{S}} \frac{1}{2} \|\mathbf{X} - \mathbf{AS}\|_F^2 + \mathcal{J}(\mathbf{A}) + \mathcal{G}(\mathbf{S}) \tag{2}$$

The first term is a classical data fidelity term that measures the discrepancy between the data and the mixture model. The $\|.\|_F$ norm refers to the Frobenius norms, whose use stems from the assumption that the noise is Gaussian. The penalizations $\mathcal{J}$ and $\mathcal{G}$ enforce some desired properties on $\mathbf{A}$ and $\mathbf{S}$ (*e.g.* sparsity, non-negativity). In the following, we will consider that the proximal operators of $\mathcal{J}$ and $\mathcal{G}$ are defined, and that $\mathcal{J}$ and $\mathcal{G}$ are convex. However, the whole matrix factorization problem (2) is non-convex. Consequently, the strategy of optimization has a critical impact on the separation performances, especially to avoid spurious local minimizers and to reduce the sensitivity to initialization. A common idea of several strategies (Block Coordinate Relaxation - BCR [14], Proximal Alternating Linearized Minimization - PALM [15], Alternating Least Squares - ALS) is to benefit from the multi-convex structure of (2) by using blocks [16] in which each sub-problem is convex. The minimization is then performed alternately with respect to one of the coordinate blocks while the other coordinates stay fixed, which entails solving a sequence of convex optimization problems. Most of the already existing methods can then be categorized in one of two families, depending on the block sizes:

- *Hierarchical or deflation methods*: these algorithms use a block of size 1. For instance, Hierarchical ALS (HALS) ([12] and references therein) updates only one specific column of $\mathbf{A}$ and one specific row of $\mathbf{S}$ at each iteration. The main advantage of this family is that each subproblem is often much simpler as their minimizer generally admits a closed-form expression. Moreover, the matrices involved being small, the computation time is much lower. The drawback is however that the errors on some sources/mixing matrix columns propagate from one iteration to the other since they are updated independently.

- *Full-size blocks*: these algorithms use as blocks the whole matrices $\mathbf{A}$ and $\mathbf{S}$ (the block size is thus equal to n). For instance, GMCA [1], which is reminiscent of the projected Alternating Least Squares (pALS) algorithm, is part of this family. One problem compared to hierarchical or deflation methods is that the problem is more complex due to the simultaneous estimation of a high number of sources. Moreover, the computational cost increases quickly with the number of sources.

The gist of the proposed *bGMCA* algorithm is to adopt an alternative approach that uses intermediate block sizes. The underlying intuition is that using blocks of intermediate size can be recast as small-scale source separation problems, which are simpler to solve as testified by Fig. 1. As a byproduct, small-size subproblems are also less costly to tackle.

2.2. Block *based optimization*

In the following, *bGMCA* minimizes the problem in eq. (2) with *blocks*, which are indexed by a set of indices I of size r, $1 \leqslant r \leqslant n$. In practice, the

minimization is performed at each iteration on submatrices of $\mathbf{A}$ (keeping only the columns indexed by I) and $\mathbf{S}$ (keeping only the rows indexed by I).

2.2.1. Minimizing multi-convex problems

Block coordinate relaxation (BCR, [14]) is performed by minimizing (2) according to a single block while the others remain fixed. In this setting, Tseng [14] proved the convergence of BCR to minimize non-smooth optimization problems of the form (2). Although we adopted this strategy to tackle sparse NMF problems in [13], BCR requires an exact minimization for one block at each iteration, which generally leads to a high computational cost. We therefore opted for Proximal Alternating Linearized Minimization (PALM), which was introduced in [15]. It rather performs a single proximal gradient descent step for each coordinate at each iteration. Consequently, the PALM algorithm is generally much faster than BCR and its convergence to a stationary point of the multi-convex problem is guaranteed under mild conditions. In the framework of the proposed *bGMCA* algorithm, a PALM-based algorithm requires minimizing at each iteration eq. (2) over blocks of size $1 \leqslant r \leqslant n$ and alternating between the update of some *submatrices* of $\mathbf{A}$ and $\mathbf{S}$ (these submatrices will be noted $\mathbf{A}_I$ and $\mathbf{S}_I$). This reads at iteration (k) as:

1 - Update of a submatrix of $\mathbf{S}$ using a fixed $\mathbf{A}$:

$$\mathbf{S}_I^{(k)} = \texttt{prox}_{\frac{\gamma \mathcal{G}(.)}{\left\| \mathbf{A}_I^{(k-1)^T} \mathbf{A}_I^{(k-1)} \right\|_2}} \left(\mathbf{S}_I^{(k-1)} - \frac{\gamma}{\left\| \mathbf{A}_I^{(k-1)^T} \mathbf{A}_I^{(k-1)} \right\|_2} \mathbf{A}_I^{(k-1)^T} (\mathbf{A}^{(k-1)} \mathbf{S}^{(k-1)} - \mathbf{X}) \right) \tag{3}$$

2 - Update of a submatrix of $\mathbf{A}$ using a fixed $\mathbf{S}$:

$$\mathbf{A}_I^{(k)} = \texttt{prox}_{\frac{\delta \mathcal{J}(.)}{\left\|\mathbf{s}_I^{(k)}\mathbf{s}_I^{(k)^T}\right\|_2}} \left(\mathbf{A}_I^{(k-1)} - \frac{\delta}{\left\|\mathbf{S}_I^{(k)}\mathbf{S}_I^{(k)^T}\right\|_2} (\mathbf{A}^{(k-1)}\mathbf{S}^{(k)} - \mathbf{X})\mathbf{S}_I^{(k)^T} \right) \tag{4}$$

In eq. (3) and (4), the operator $\texttt{prox}_f$ is the proximal operator of f (cf. Appendix and [17] [18]). The scalars γ and δ are the gradient path lengths. The $\|.\|_2$ norm is the matrix norm induced by the ℓ_2 norm for vectors. More specifically, if x is a vector and $\|.\|_{\ell_2}$ is the ℓ_2 norm for vectors, the $\|.\|_2$ induced matrix norm is defined as:

$$\|\mathbf{M}\|_2 = \sup_{x \neq 0} \frac{\|\mathbf{M}x\|_{\ell_2}}{\|x\|_{\ell_2}} \tag{5}$$

Block choice

Several strategies for selecting at each iteration the block indices I have been investigated: i) *Sequential*: at each iteration, r sources are selected sequentially in a cyclic way; ii) *Random*: at each iteration, r indices in $[1, n]$ are randomly chosen following a uniform distribution and the corresponding sources updated; iii) *Random sequential*: this strategy combines the sequential and the random choices to ensure that all sources are updated an equal number of times. In the experiments, random strategies tended to provide better results. Indeed, compared to a sequential choice, randomness is likely to make the algorithm more robust with respect to spurious local minima. Since the results between the random strategy and the random sequential one are similar, the first was eventually selected.

Examined cases and corresponding proximal operators

In several practical examples, an explicit expression can be computed for

the proximal operators. In the next, the following penalizations have been considered:

1 - Penalizations $\mathcal{G}$ for the sources $\mathbf{S}$:

- ℓ_1 *sparsity constraint in some transformed domain*: The sparsity constraint on $\mathbf{S}$ is enforced with a ℓ_1-norm penalization: $\mathcal{G}(\mathbf{S}) = \left\|\Lambda_{\mathbf{S}} \odot (\mathbf{S}\mathbf{\Phi}_{\mathbf{S}}^{\mathbf{T}})\right\|_1$, where the matrix $\mathbf{\Lambda_S}$ contains regularization parameters and $\odot$ denotes the Hadamard product. $\mathbf{\Phi_S}$ is a transform into a domain in which $\mathbf{S}$ can be sparsely represented. In the following, $\mathbf{\Phi_S}$ will be supposed to be orthogonal. The proximal operator for $\mathcal{G}$ in (3) is then explicit and corresponds to the soft-thresholding operator with threshold $\Lambda_{\mathbf{S}}$, which we shall denote $\mathcal{S}_{\Lambda_{\mathbf{S}}}(.)$ (cf. Appendix). Using $\gamma = 1$ and assuming $\mathbf{\Phi_S}$ orthogonal, the update is then:

$$\mathbf{S}_I^{(k)} = \mathcal{S}_{\Lambda_{\mathbf{S}}}\left(\mathbf{S}_I^{(k-1)}{\mathbf{\Phi_S}}^T - \frac{1}{\left\|\mathbf{A}_I^{(k-1)}{\mathbf{A}_I^{(k-1)}}^T\right\|_2}{\mathbf{A}_I^{(k-1)}}^T(\mathbf{A}^{(k-1)}\mathbf{S}^{(k-1)} - \mathbf{X}){\mathbf{\Phi_S}}^T\right)\mathbf{\Phi_S} \tag{6}$$

- *Non-negativity in the direct domain and* ℓ_1 *sparsity constraint in some transformed domain*: due to the non-negativity constraint, all coefficients in $\mathbf{S}$ must be non-negative in the direct domain in addition to the sparsity constraint in a transformed domain $\mathbf{\Phi_S}$. It can be formulated as $\mathcal{G}(\mathbf{S}) = \|\Lambda_{\mathbf{S}} \odot \left(\mathbf{S}{\mathbf{\Phi_S}}^T\right)\|_{\ell_1} + \iota_{\{\forall j,k;\mathbf{S}[j,k]\geq 0\}}(\mathbf{S})$ where ι_U is the indicator function of the set U. The difficulty is to enforce at the same time two constraints in two different domains, since the proximal operator of $\mathcal{G}$ is not explicit. It can

either be roughly approximated by composing the proximal operators of the individual penalizations to produce a cheap update or computed accurately using the Generalized Forward-Backward splitting algorithm [19].

2 - Penalizations $\mathcal{J}$ for the mixing matrix $\mathbf{A}$:

- *Oblique constraint*: to avoid obtaining degenerated $\mathbf{A}$ and $\mathbf{S}$ matrices ($\|\mathbf{A}\| \to \infty$ and $\|\mathbf{S}\| \to 0$), the columns of $\mathbf{A}$ are constrained to be in the ℓ_2 ball, i.e. $\forall j \in [1, n], \|\mathbf{A}_j\|_2 \leqslant 1$. More specifically, $\mathcal{J}$ can be written as $\mathcal{J}(\mathbf{A}) = \iota_{\{\forall i; \|\mathbf{A}^i\|_2^2 \leq 1\}}(\mathbf{A})$. Following this constraint, the proximal operator for $\mathcal{J}$ in eq. (4) is explicit and can be shown to be the projection $\Pi_{\|.\|_2}$ (cf. Appendix) on the ℓ_2 unit ball of each column of the input. The update (4) of $\mathbf{A}_I$ becomes:

$$\mathbf{A}_I^{(k)} = \Pi_{\|.\|_2 \leqslant 1} \left(\mathbf{A}_I^{(k-1)} - \frac{1}{\left\| \mathbf{S}_I^{(k)} \mathbf{S}_I^{(k)^T} \right\|_2} (\mathbf{A}^{(k-1)} \mathbf{S}^{(k)} - \mathbf{X}) \mathbf{S}_I^{(k)^T} \right) \tag{7}$$

- *Non-negativity and oblique constraint*: Adding the non-negativity constraint on $\mathbf{A}$ reads: $\mathcal{J}(\mathbf{A}) = \iota_{\forall i; \|\mathbf{A}^i\|_2^2 \leq 1}(\mathbf{A}) + \iota_{\forall i,j; \mathbf{A}[i,j] \geq 0}(\mathbf{A})$. The proximal operator can be shown to be the composition of the proximal operator corresponding to non-negativity followed by $\Pi_{\|.\|_2 \leqslant 1}$. The proximal operator corresponding to non-negativity is the projection Π_{K^+} (cf. Appendix) on the positive orthant K^+.

The update is then:

$$\mathbf{A}_I^{(k)} = \Pi_{\|.\|_2 \leqslant 1} \left(\Pi_{K^+} \left(\mathbf{A}_I^{(k-1)} - \frac{1}{\left\| \mathbf{S}_I^{(k)} \mathbf{S}_I^{(k)^T} \right\|_2} (\mathbf{A}^{(k-1)} \mathbf{S}^{(k)} - \mathbf{X}) \mathbf{S}_I^{(k)^T} \right) \right) \tag{8}$$

2.2.2. Minimization: introduction of a warm-up stage

While being provably convergent to a stationary point of (2), the above PALM-based algorithm suffers from a lack of robustness with regards to a bad initialization, which makes it more prone to be trapped in spurious local minima. Moreover, it is quite difficult to automatically tune the thresholds $\mathbf{\Lambda}$ so that it yields reasonable results. On the other hand, algorithms based on GMCA [1, 9] have been shown to be robust to initialization. Furthermore, in this framework, fixing the parameters $\mathbf{\Lambda}$ can be done in an automatic manner. However, GMCA-like algorithms are based on heuristics, which preclude provable convergence to a minimum of (2).

The proposed strategy consists in combining the best of both approaches to build a two-stage minimization procedure (cf. Algorithm 1): i) a warm-up stage building upon the GMCA algorithm to provide a fast and reliable first guess, and ii) a refinement stage based on the above PALM-based algorithm that provably yields a minimizer of (2). Moreover, the thresholds $\mathbf{\Lambda}$ in the refinement stage will be naturally derived from the first stage. Based on the GMCA algorithm [1, 9], the warm-up stage is summarized below:

0 - Initialize the algorithm with random $\mathbf{A}$. For each iteration (k):

1 - The sources are first updated assuming a fixed $\mathbf{A}$. A submatrix $\mathbf{S}_I$ is however now updated instead of $\mathbf{S}$. This is performed using a projected

least square solution:

$$\mathbf{S}_I^{(k)} = \texttt{prox}_{\mathcal{G}(.)}(\mathbf{A}_I^{(k-1)\dagger}\mathbf{R}_I) \tag{9}$$

where: $\mathbf{R}_I$ is the residual term defined by $\mathbf{R}_I = \mathbf{X} - \mathbf{A}_{I^C}^{(k)}\mathbf{S}_{I^C}^{(k)}$ (with I^C the indices of the sources outside the block), which is the part of $\mathbf{X}$ to be explained by the sources in the current block I. $\mathbf{A}_I^{(k)\dagger}$ is the pseudo-inverse of $\mathbf{A}_I^{(k)}$, the estimate of $\mathbf{A}_I$ at iteration (k).

2 - The mixing sub-matrix $\mathbf{A}_I$ is similarly updated with a fixed $\mathbf{S}$:

$$\mathbf{A}_I^{(k)} = \texttt{prox}_{\mathcal{J}(.)}(\mathbf{R}_I\mathbf{S}_I^{(k)\dagger}) \tag{10}$$

The warm-up stage stops after a given number of iterations. Since the penalizations are the same as in the refinement stage, the proximal operators can be computed with the formulae described previously, depending on the implemented constraints. For $\mathbf{S}$, eq. (6) can be used to enforce sparsity. To enforce non-negativity and sparsity in some transformed domain, the cheap update described in section 2.2.1 consisting in composing the proximal operators of the individual penalizations can be used. For $\mathbf{A}$, equations (7) and (8) can be used depending on the implemented constraint.

2.2.3. Heuristics for the warm-up stage

In the spirit of GMCA, the *bGMCA* algorithm exploits heuristics to make the separation process more robust to initialization, which mainly consists in making use of a decreasing thresholding strategy. In brief, the entries of the threshold matrix $\mathbf{\Lambda}$ first start with large values and then decrease along the iterations towards final values that only depend on the noise level. This

stategy has been shown to significantly improve the performances of the separation process [1, 9] as it provides: i) *a better unmixing*, ii) an *increased robustness to noise*, and iii) an *increased robustness to spurious local minima*.

In the *bGMCA* algorithm, this strategy is deployed by first identifying the coefficients of each source in I that are not statistically consistent with noise. Assuming that each source is contaminated with a Gaussian noise with standard deviation σ, this is performed by retaining only the entries whose amplitude is larger than $\tau\,\sigma$, where $\tau \in [2, 3]$. In practice, the noise standard deviation is estimated empirically using the Median Absolute Deviation (MAD) estimator. For each source in I, the actual threshold at iteration k is fixed based on a given percentile of the available coefficients with the largest amplitudes. Decreasing the threshold at each iteration is then performed by linearly increasing the percentage of retained coefficients at each iteration: $\texttt{Percentage} = \frac{k}{\sharp\texttt{iterations}} \times 100$.

2.2.4. Convergence

The *bGMCA* algorithm combines sequentially the above warm-up stage and the PALM-based refinement stage. Equipped with the decreasing thresholding strategy, it cannot be proved that the warm-up stage neither converges to a stationary point of eq. (2) nor converges at all. In practice, after consecutive iterates, the warm-up stage tends to stabilize. However, it plays a key role to provide a reasonable starting point, as well as threshold values $\mathbf{\Lambda}$ for the refinement procedure. In the refinement stage, the thresholds are computed from the matrices estimated in the warm-up and fixed for the whole refinement step. Based on the PALM algorithm, and with these fixed

thresholds, the refinement stage converges to a stationary point of eq. (2). The convergence is also guaranteed with the proposed block-based strategy, as long as the blocks are updated following an essentially cyclic rule [20] or even if they are chosen randomly and updated one by one [21].

2.2.5. Required number of iterations

Intuitively, the required number of iterations should be inversely proportional to r, since only r sources are updated at each iteration, requiring $\lceil n/r \rceil$ times the number of iterations needed by an algorithm using the full matrices. As will be emphasized later on, the number of required iterations will be smaller than expected, which reduces the computation time.

In the refinement stage, the stopping criterion is based on the angular distance for each column of $\mathbf{A}$, i.e. the angle between the current column and that of the previous iteration. Then, the mean over all the columns is taken:

$$\Delta = \frac{\sum_{j \in [1,n]} \left\| \mathbf{A}_j^{(k)} \odot \mathbf{A}_j^{(k-1)} \right\|_1}{n} \tag{11}$$

The stopping criterion itself is then a threshold τ used to stop the algorithm when $\Delta > \tau$. In addition, we also fixed a maximal number of iterations.

3. Numerical experiments on simulated data

In this part, we present our results on simulated data. The goal is to show and to explain on simple data how *bGMCA* works.

3.1. Experimental protocol

The simulated data were generated in the following way:

Algorithm 1 *bGMCA*

Warm-up step

for $0 \leqslant k < n_{\max}$ **do**

Choose a set of indices I

Estimation of $\mathbf{S}$ with a fixed $\mathbf{A}$: $\mathbf{S}_I^{(k)} = \texttt{prox}_{\mathcal{G}(.)}(\mathbf{A}_I^{(k-1)\dagger}\mathbf{R}_I)$

Estimation of $\mathbf{A}$ with a fixed $\mathbf{S}$: $\mathbf{A}_I^{(k)} = \texttt{prox}_{\mathcal{J}(.)}(\mathbf{R}_I\mathbf{S}_I^{(k)\dagger})$

Choice of a new threshold $\Lambda^{(k)}$ ▷ heuristic - see section 2.2.3

end for

Refinement step

while $\Delta > \tau$ and $k < n_{\max}$ **do**

Choose a set of indices I

$$\mathbf{S}_I^{(k)} = \texttt{prox}_{\frac{\gamma\mathcal{G}(.)}{\left\|\mathbf{A}_I^{(k-1)^T}\mathbf{A}_I^{(k-1)}\right\|_2}}\left(\mathbf{S}_I^{(k-1)} - \frac{\gamma}{\left\|\mathbf{A}_I^{(k-1)^T}\mathbf{A}_I^{(k-1)}\right\|_2}\mathbf{A}_I^{(k-1)^T}(\mathbf{A}^{(k-1)}\mathbf{S}^{(k-1)} - \mathbf{X})\right)$$

$$\mathbf{A}_I^{(k)} = \texttt{prox}_{\frac{\delta\mathcal{J}(.)}{\left\|\mathbf{S}_I^{(k)}\mathbf{S}_I^{(k)^T}\right\|_2}}\left(\mathbf{A}_I^{(k-1)} - \frac{\delta}{\left\|\mathbf{S}_I^{(k)}\mathbf{S}_I^{(k)^T}\right\|_2}(\mathbf{A}^{(k-1)}\mathbf{S}^{(k)} - \mathbf{X})\mathbf{S}_I^{(k)^T}\right)$$

$$\Delta = \frac{\sum_{j\in[1,n]}\left\|\mathbf{A}_j^{(k)} \odot \mathbf{A}_j^{(k-1)}\right\|_1}{n}$$

$k = k + 1$

end while

return $\mathbf{A}$, $\mathbf{S}$

1 - Source matrix $\mathbf{S}$: the sources are sparse in the sample domain without requiring any transform (the results would however be identical for any source sparse in an orthogonal representation). The sources in $\mathbf{S}$ are exactly sparse and drawn randomly according to a Bernoulli-Gaussian distribution: among the t samples ($t = 1,000$), a proportion p (called sparsity degree—unless specified, $p = 0.1$) of the samples

is taken non-zero, with an amplitude drawn according to a standard normal distribution.

2 - Mixing matrix $\mathbf{A}$: the mixing matrix is drawn randomly according to a standard normal distribution and modified to have unit columns and a given condition number C_d (unless specified, $C_d = 1$).

The number of observations m is taken equal to the number of sources: $m = n$. In this first simulation, no noise is added. The algorithm was launched with 10, 000 iterations. It has to be emphasized that since neither $\mathbf{A}$ nor $\mathbf{S}$ are non-negative, the corresponding proximal operators we used did not enforce non-negativy. Thus, we used soft-thresholding for $\mathbf{S}$ and the oblique constraint for $\mathbf{A}$ according to section 2.2.1.

To measure the accuracy of the separation, we followed the definition in [9] to use a global criterion on $\mathbf{A}$: $C_A = \texttt{median}(|\mathbf{P}\mathbf{A}^\dagger\mathbf{A}^*| - \mathbf{I_d})$, where $\mathbf{A}^*$ is the true mixing matrix and $\mathbf{A}$ is the solution given by the algorithm, corrected through $\mathbf{P}$ for the permutation and scale factors indeterminacies. $\mathbf{I_d}$ is the identity matrix. This criterion quantifies the quality of the estimation of the mixing directions, that is the columns of $\mathbf{A}$. If they are perfectly estimated, $|\mathbf{P}\mathbf{A}^\dagger\mathbf{A}^*|$ is equal to $\mathbf{I_d}$ and $C_A = 0$. The data matrices being drawn randomly, each experiment was performed several times (typically 25 times) and the median of $-10\log(C_A)$ over the experiments will be displayed. The logarithm is used to simplify the reading of the plots despite the high dynamics.

3.2. Modeling block minimization

In this section, a simple model is introduced to describe the behavior of the *bGMCA* algorithm. As described in section 2.2, updating a given block

is performed at each iteration from the residual $\mathbf{R}_I = \mathbf{X} - \mathbf{A}_{I^C}\mathbf{S}_{I^C}$. If the estimation were perfect, the residual would be equal to the part of the data explained by the true sources in the current block indexed by I, which would read: $\mathbf{R}_I = \mathbf{A}^*_I\mathbf{S}^*_I$, $\mathbf{A}^*$ and $\mathbf{S}^*$ being the true matrices.
It is nevertheless mandatory to take into account the noise $\mathbf{N}$, as well as a variety of flaws in the estimation by adding a term $\mathcal{E}$ to model the estimation error. This entails:

$$\mathbf{R}_I = \mathbf{X} - \mathbf{A}_{I^C}\mathbf{S}_{I^C} = \mathbf{A}^*_I\mathbf{S}^*_I + \mathcal{E} + \mathbf{N} \tag{12}$$

A way to further describe the structure of $\mathcal{E}$ is to decompose the $\mathbf{S}$ matrix in the true matrix plus an error: $\mathbf{S}_I = \mathbf{S}^*_I + \epsilon_I$ and $\mathbf{S}_{I^C} = \mathbf{S}^*_{I^C} + \epsilon_{I^C}$, where $\mathbf{S}$ is the estimated matrix, and ϵ is the error on $\mathbf{S}^*$. Assuming that the errors are small and neglecting the second-order terms, the residual $\mathbf{R}_I$ can now be written as:

$$\mathbf{R}_I = \mathbf{X} - \mathbf{A}_{I^C}\mathbf{S}_{I^C} = \mathbf{A}^*_I\mathbf{S}^*_I + \mathbf{A}^*_{I^C}\mathbf{S}^*_{I^C} - \mathbf{A}_{I^C}\mathbf{S}^*_{I^C} - \mathbf{A}_{I^C}\epsilon_{I^C} + \mathbf{N} \tag{13}$$

This implies that:

$$\mathcal{E} = (\mathbf{A}^*_{I^C} - \mathbf{A}_{I^C})\mathbf{S}^*_{I^C} - \mathbf{A}_{I^C}\epsilon_{I^C} \tag{14}$$

Equation (14) highlights two terms. The first term can be qualified as interferences in that it comes from a leakage of the *true* sources that are *outside* the currently updated block. This term vanishes when $\mathbf{A}_{I^C}$ is perfectly estimated. The second term corresponds to interferences as well as artefacts. It originates indeed from the *error* on the sources *outside* the block I. The artefacts are the errors on the sources induced by the soft thresholding corresponding to the ℓ_1-norm.

Equation (14) also allows us to understand how the choice of a given block size $r \leqslant n$ will impact the separation process:

- Updating small-size blocks can be recast as a small-size source separation problem where the actual number of sources is equal to r. The residual of the sources that are not part of the block I then plays the role of extra noise. As testified by Fig. 1, updating small-size block problems should be easier to tackle.

- Small-size blocks should also yield larger errors $\mathcal{E}$. It is intuitively due to the fact that many potentially badly estimated sources in I^C are used for the estimation of $\mathbf{A}_I$ and $\mathbf{S}_I$ through the residual, deteriorating this estimation. It can be explained in more details using equation (14): with more sources in I^C, the energy of $\mathbf{A}_{I^C}$, $\mathbf{A}^*_{I^C}$, $\mathbf{S}^*_{I^C}$ and ϵ_{I^C} increases, yielding bigger error terms $(\mathbf{A}^*_{I^C} - \mathbf{A}_{I^C})\mathbf{S}^*_{I^C}$ and $-\mathbf{A}_{I^C}\epsilon_{I^C}$. Therefore the errors $\mathcal{E}$ become higher, deteriorating the results.

3.3. Experiment

In this section, we investigate the behavior of the proposed block-based GMCA algorithm with respect to various parameters such as the block size, the number of sources, the conditioning of the mixing matrix and the sparsity level of the sources.

3.3.1. Study of the impact of r and n

In this subsection, *bGMCA* is evaluated for different numbers of sources $n = 20, 50, 100$. Each time the block sizes vary in the range $1 \leq r \leq n$. In this experiment and to complete the description of section 3.1, the parameters

for the matrices generation were: $p = 0.1$, $t = 1,000$, $C_d = 1$, $m = n$, with a Bernoulli-Gaussian distribution for the sources. These results are displayed in Fig. 2a. Interestingly, three different regimes characterize the behavior of the *bGMCA* algorithm:

- For intermediate and relatively large block sizes (*typically* $r > 5$ and $r < n - 5$): we first observe that after an initial deterioration around $r = 5$, the separation quality does not vary significantly for increasing block sizes. A degradation of several dB can then be observed for r close to n. In all this part of the curve, the error term $\mathcal{E}$ is composed of residuals of sparse sources, and thus $\mathcal{E}$ will be rather sparse when the block size is large. Based on the MAD, the thresholds are set according to dense and not to sparse noise. Consequently the automatic thresholding strategy of the *bGMCA* algorithm will not be sensitive to the estimation errors.

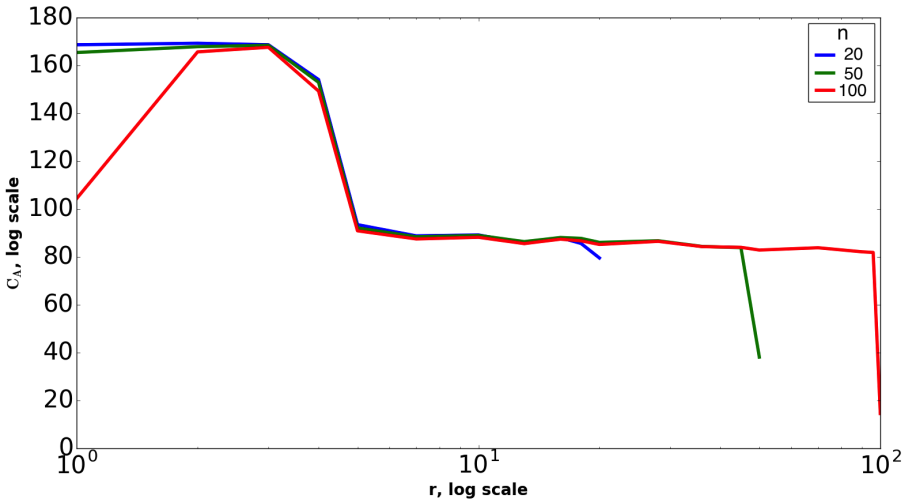
- A very prominent peak can be observed when the block size is of the order of 3. Interestingly, the maximum yields a mixing matrix criterion of about 10^{-16}, which means that perfect separation is reached up to numerical errors. This value of 160 dB is at least 80 dB larger than in the standard case $r = n$, for which the values for the different n are all below 80 dB. In this regime, error propagation is composed of the mixture of a larger number of sparse sources, which eventually entails a densely distributed contribution that can be measured by the MAD-based thresholding procedure. Therefore, the threshold used to estimate the sources is able to filter out both the noise and the estimation errors. Moreover, $r = 5$ is quite small compared to n. Following

the modeling introduced in section 3.2, small block sizes can be recast as a sequence of low-dimensional blind source separation problems, which are simpler to solve.

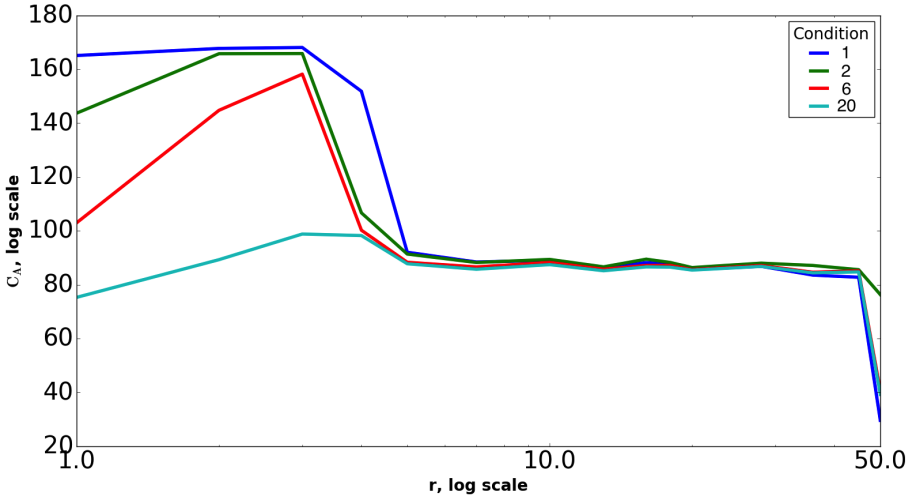
- For small block sizes (*typically* $r < 4$), the separation quality is deteriorated when the block size decreases, especially for large n values. In this regime, the level of estimation error $\mathcal{E}$ becomes large, which entails large values for the thresholds $\mathbf{\Lambda}$. Consequently, the bias induced by the soft-thresholding operator increases, which eventually hampers the performance quality. Furthermore, for a fixed block size r, $\mathcal{E}$ increases with the number of sources n, making this phenomenon more pronounced for higher n values.

3.3.2. Condition number of the mixing matrix

In this section, we investigate the role played by the conditioning of the mixing matrix on the performances of the *bGMCA* algorithm. Fig. 2b displays the empirical results for several condition numbers C_d of the $\mathbf{A}$ matrix. There are $n = 50$ sources generated in the same way as in the previous experiment: with a Bernoulli-Gaussian distribution and $p = 0.1$, $t = 1,000$. One can observe that when C_d increases, the peak present for r close to 5 tends to be flattened, which is probably due to higher projection errors. At some iteration k, the sources are estimated by projecting $\mathbf{X} - \mathbf{A}_{I^c}\mathbf{S}_{I^c}$ onto the subspace spanned by $\mathbf{A}_I$. In the orthogonal case, the projection error is low since $\mathbf{A}_{I^c}$ and $\mathbf{A}_I$ are close to orthogonality at the solution. However, this error increases with the condition number C_d.



(a) Number of sources.



(b) Condition number.

Figure 2: Left: mixing matrix criterion as a function of r for different n. Right: mixing matrix criterion as a function of r for different C_d.

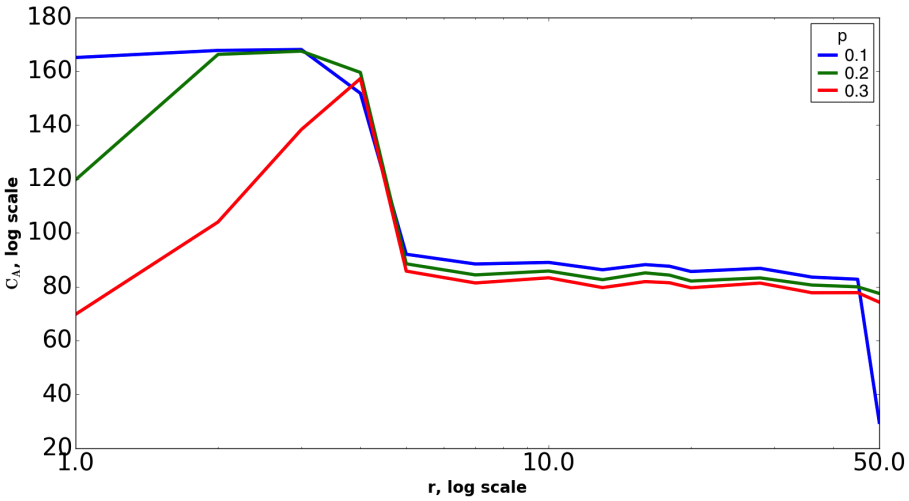


Figure 3: Mixing matrix criterion as a function of r for different sparsity degrees.

3.3.3. Sparsity level p

In this section, the impact of the sparsity level of the sources is investigated. The sources are still following a Bernoulli-Gaussian distribution. The parameters are: $n = 50$, $t = 1,000$, $C_d = 1$. As featured in Figure 3, the separation performances at the maximum value decrease slightly with larger p, while a slow shift of the transition between the small/large block size regimes towards larger block sizes operates. Furthermore, the results tend to deteriorate quickly for small block sizes ($r < 4$). Indeed, owing to the model of subsection 3.2, the contribution of $\mathbf{S}^*_{IC}$ and ϵ_{IC} in the error term (14) increases with p, this effect being even more important for small r (which could also explain the shift of the peak for $p = 0.3$, by a deterioration of the results at its beginning, $r = 3$). When p increases, the sources in $\mathbf{S}_I$ become denser. Instead of being mainly sensitive to the noise and $\mathcal{E}$, the MAD-based thresholding tends to be perturbated by $\mathbf{S}_I$, resulting in more

artefacts, which eventually hampers the separation performances. This effect increases when the sparsity level of the sources decreases.

3.3.4. Complexity and computation time

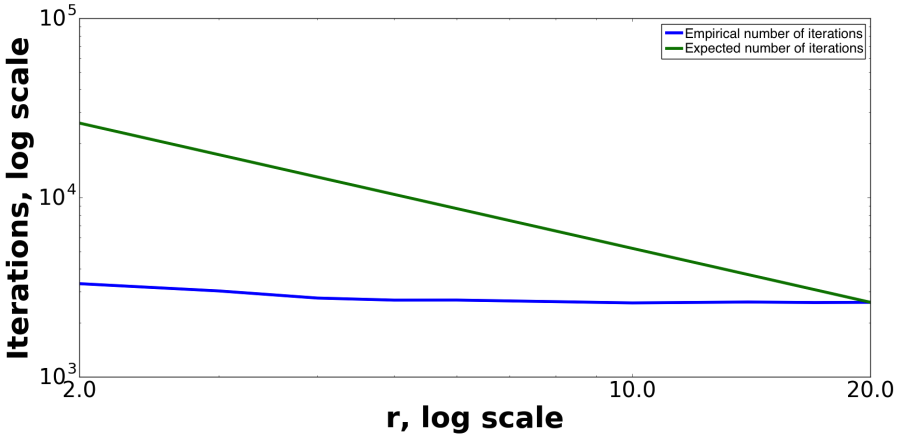


Figure 4: Right: number of iterations in logarithmic scale as a function of r.

Beyond improving the separation performances, the use of small block sizes decreases the computational cost of each iteration of the *bGMCA* algorithm. Since it is iterative, the final running time will depend on both the complexity of each iteration and of the number of iterations. In this part, we focus only on the warm-up stage, which is empirically the most computationally expensive stage. Each iteration of the warm-up stage can be decomposed into the following elementary steps: i) a residual term is computed with a complexity of $\mathcal{O}(mtr)$, where m is the number of observations, t the number of samples and r the block size; ii) the pseudo-inverse is performed with the singular value decomposition of a $r \times r$ matrix, which yield an overall complexity of $\mathcal{O}(r^3 + r^2m + m^2r)$; iii) the thresholding-strategy

first requires the evaluation of the threshold values, which has a complexity of rt; iv) then the soft-thresholding step which has complexity $\mathcal{O}(rt)$; and v) updating $\mathbf{A}$ is finally performed using a conjugate gradient algorithm, whose complexity is known to depend on the number of non-zero entries in $\mathbf{S}$ and on the condition of this matrix $C_d(\mathbf{S})$. An upperbound for this complexity is thus $\mathcal{O}(rt\sqrt{C_d(\mathbf{S})})$. The final estimate of the complexity of a single iteration is finally given by:

$$r[mt + rm + m^2 + r^2 + t\sqrt{C_d(\mathbf{S})}] \tag{15}$$

With $C_d(\mathbf{S})$ the conditioning number of $\mathbf{S}$. Thus, both the r factor and the behavior in r^3 show that small r values will lower the computational budget of each iteration. We further assess the actual *number of iterations* required by the warm-up stage to yield a good initialization. To this end, the following experiment has been conducted:

1. First, the algorithm is launched with a large number of iterations (*e.g.* 10000) to give a good initialization for the $\mathbf{A}$ and $\mathbf{S}$ matrices. The corresponding value of C_A is saved and called C_A^*.

2. Using the same initial conditions, the warm-up stage is re-launched and stops when the mixing matrix criterion reaches $1.05 \times C_A^*$ (*i.e.* 5% of the "optimal"initialization for a given setting).

The number of iterations needed to reach the 5% accuracy is reported in Fig. 4. Intuitively, one would expect that when the block size decreases, the required number of iterations should increase by about n/r to keep the number of updates per source constant. This trend is displayed with the

straight curve of Fig. 4. Interestingly, Fig. 4 shows that the actual number of iterations to reach the 5% accuracy criterion almost does not vary with r. Consequently, on top of leading to computationally cheaper iterations, using small block sizes does not require more iterations for the warm-up stage to give a good initialization. Therefore, the use of blocks allows a huge decrease of the computational cost of the warm-up stage and thus of sparse BSS.

4. Experiment using realistic sources

4.1. Context

The goal of this part is to evaluate the behavior of *bGMCA* and show its efficiency in a more realistic setting. Our data come from a simulated LC - ^{1}H NMR (Liquid Chromatography - ^{1}H Nuclear Magnetic Resonance) experiment. The objective of such a experiment is to identify each of the chemicals compounds present in a fluid, as well as their concentrations. The LC - ^{1}H NMR experiment enables a first physical imperfect separation during which the fluid goes through a chromatography column and its chemicals are separated according to their speeds (which themselves depend on their physical properties). Then, the spectrum of the output of the column is measured at a given time frequency. These measurements of the spectra at different times can be used to feed a *bGMCA* algorithm to refine the imperfect physical separation.

The fluids on which we worked could for instance correspond to drinks. The goal of *bGMCA* is then to identify the spectra of each compound (*e.g.* caffein, saccharose, menthone...) and the mixing coefficients (which are proportional to their concentrations) from the LC - ^{1}H NMR data. BSS has already been

successfully applied [22] to similar problems but generally with lower number of sources n.

The sources (40 sources with each 10, 000 samples) are composed of elementary sparse non-negative theoretical spectra of chemical compounds taken from the SDBS database[1], which are further convolved with a Laplacian having a width of 3 samples to simulate a given spectral resolution. Therefore, each convolved source becomes an approximately sparse non-negative row of $\mathbf{S}$. The mixing matrix $\mathbf{A}$ of size (m,n) = (320,40) is composed of Gaussians (see Fig. 5), the objective being to have a matrix that could be consistent with the first imperfect physical separation. It is designed in two parts: the first columns have relatively spaced Gaussian means while the others have a larger overlap to simulate compounds for which the physical separation is less discriminative. More precisely, an index $\bar{m} \in [1, m]$ is chosen, with $\bar{m} > m/2$ (typically, $\bar{m} = \lceil 0.75m \rceil$). A set of $\lfloor n/2 \rfloor$ indices $(m_k)_{k=1\ldots\lfloor n/2 \rfloor}$ is then uniformly chosen in $[0, \bar{m}]$ and another set of $\lceil n/2 \rceil$ indices $(m_k)_{k=\lceil n/2 \rceil \ldots n}$ is chosen in $[\bar{m}+1, m]$. Each column of $\mathbf{A}$ is then created as a Gaussian whose mean is m_k. Monte-carlo simulations have been carried out by randomly assigning the sources and the mixing matrix columns. The median over the results of the different experiments will be displayed.

4.2. Experiments

There are two main differences with the previous experiments of section 3: i) the sources are sparse in the undecimated wavelet domain $\mathbf{\Phi_S}$, which is

[1] National Institute of Advanced Industrial Science and Technology (AIST), Spectral database for organic compounds: `http://sdbs.db.aist.go.jp`

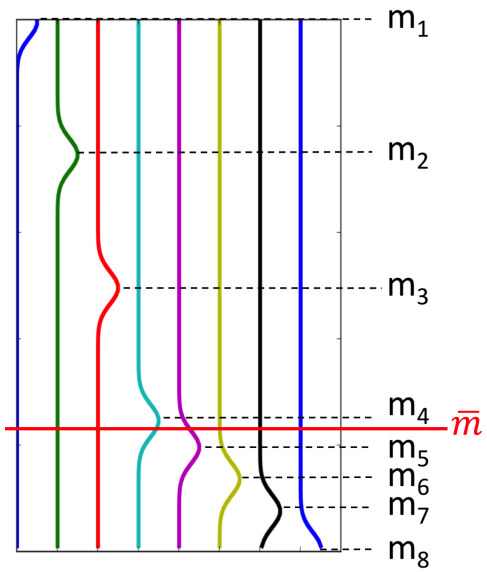


Figure 5: Exemple of **A** matrix with 8 columns: the four first columns have spaced means, while the last ones are more correlated

chosen as the starlet transform [23] in the following, and ii) the non-negativity of **S** and **A** is enforced. Fig. 6 (left) displays the evolution of the mixing matrix criterion with varying block sizes with and without the non-negativity constraints. The algorithm was launched with 2, 000 iterations.

These results show that non-negativity yields a huge improvement for all block sizes r, which is expected since the problem is more constrained. This is probably due to the fact that all the small negative coefficients are set to 0, thus artificially allowing lower thresholds and therefore less artefacts. This is especially advantageous in the present context with very low noise[2] (the

[2]Depending on the instrumentation, high SNR values can be reached in such an experiment

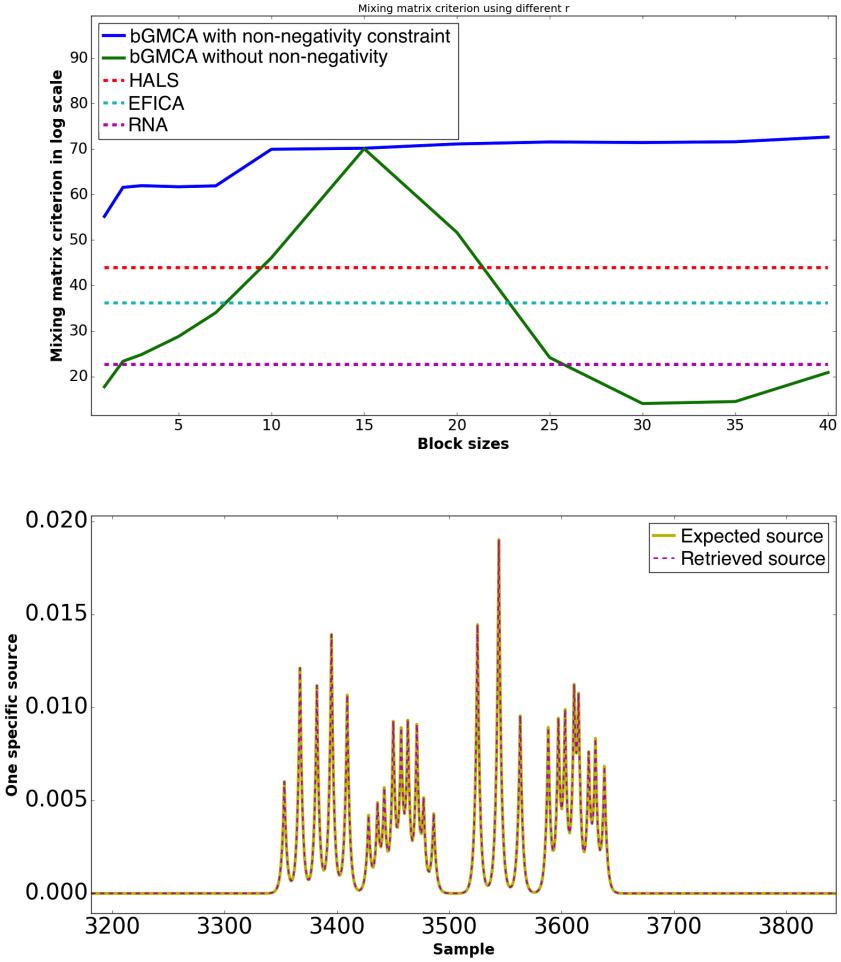


Figure 6: Left: mixing criterion on realistic sources, with and without a non-negativity constraint. Right: example of a retrieved source, which is almost perfectly superimposed on the true source, therefore showing the quality of the results.

Signal to Noise Ratio - SNR - has a value of 120 dB) where the thresholds do not need to be high to remove noise.

Furthermore, the separation quality tends to be constant for $r \geq 10$. In this particular setting, non-negativity helps curing the failure of sparse BSS

when large blocks are used. However, using smaller block sizes still allows reducing the computation cost while preserving the separation quality. The *bGMCA* with non-negativity also compares favorably with respect to other tested standard BSS methods (cf. Section 1 for more details), yielding better results for all values of r. In particular, it is always better than HALS, which also uses non-negativity. As an illustration, a single original source is displayed in the right panel of Fig. 6 after its convolution with a Laplacian. Its estimation using *bGMCA* with a non-negativity constraint is plotted in dashed line on the same graph, showing the high separation quality because of the nearly perfect overlap between the two curves. Both sources are drawn in the direct domain.

The robustness of the *bGMCA* algorithm with respect to additive Gaussian noise has further been tested. Fig. 7 reports the evolution of the mixing matrix criterion for varying values of the signal-to-noise ratio. It can be observed that *bGMCA* yields the best performances for all values of SNR. Although it seems to particularly benefit from high SNR compared to HALS and EFICA, it still yields better results than the other algorithms for low SNR despite the small block size used ($r = 10$), which could have been particularly prone to error propagations.

Conclusion

While being central in numerous applications, tackling sparse BSS problems when the number of sources is large is highly challenging. In this article, we describe the block-GMCA algorithm, which is specifically tailored to solve sparse BSS in the *large*-scale regime. In this setting, the minimiza-

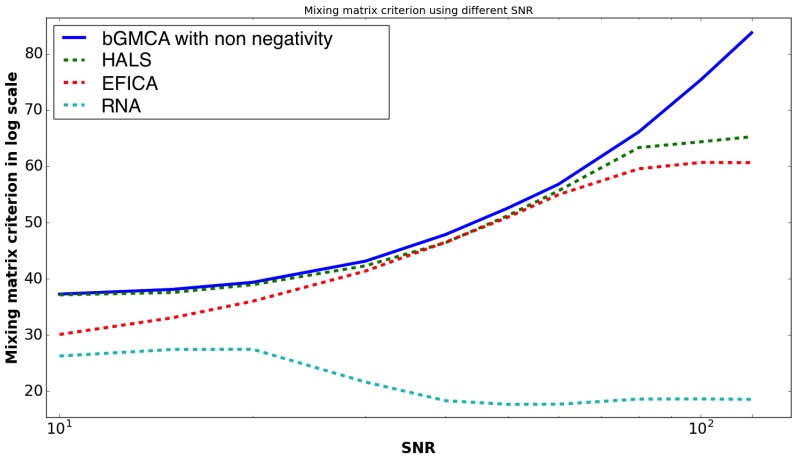


Figure 7: Mixing criterion on realistic sources, using a non-negative constraint with r = 10

tion strategy plays a key role in the robustness of BSS methods due to the non-convex nature of the problem. Therefore, and in contrast to the state-of-the-art algorithms, *bGMCA* builds upon block-coordinate optimization with intermediate-size blocks. Experiments on exactly sparse simulated data and a model presented in this work hightlight the mechanisms improving the results over the full block version, which can potentially lead to some numerically perfect separations. Furthermore, comparisons have been carried on simulated spectroscopic data, which demonstrate the reliability of the proposed algorithm in a realistic setting and its superior performances for high SNR. All the numerical comparisons conducted show that *bGMCA* performs at least as well as standard sparse BSS on mixtures of a high number of sources and most of the experiments even show dramatically enhanced separation performances. As a byproduct, the proposed block-based strategy yields a significant decrease of the computational cost of the separation process.

Acknowledgement

This work is supported by the European Community through the grant LENA (ERC StG - contract no. 678282).

Appendix

Definition of proximal operators

The proximal operator of an extended-valued proper and lower semi-continuous convex function $f : \mathbb{R}^n \to (-\infty, \infty]$ is defined as:

$$\texttt{prox}_f(v) = \texttt{argmin}_x(f(x) + \frac{1}{2}\|x - v\|_2^2) \tag{16}$$

Definition of the soft thresholding operator

The soft thresholding operator $\mathcal{S}_\lambda(.)$ is defined as:

$$\forall \mathbf{M}, \forall i \in [1, n], \forall j \in [1, m], \mathcal{S}_\lambda(\mathbf{M}_{ij}) = \begin{cases} \mathbf{M}_{ij} - \lambda \times sign(\mathbf{M}_{ij}) \texttt{ if } |\mathbf{M}_j| \geqslant \lambda \\ 0 \texttt{ otherwise} \end{cases} \tag{17}$$

Definition of the projection of the columns of a matrix $\mathbf{M}$ *on the* ℓ_2 *ball*

$$\forall \mathbf{M} \in \mathbb{R}^{m \times n}, \forall j \in [1, n], \Pi_{\|.\|_2 \leqslant 1}(\mathbf{M}_j) = \begin{cases} \mathbf{M}_j \texttt{ if } \|\mathbf{M}_j\|_2 \leqslant 1 \\ \mathbf{M}_j / \|\mathbf{M}_j\|_2 \texttt{ otherwise} \end{cases} \tag{18}$$

Definition of the projection of a matrix $\mathbf{M}$ *on the positive orthant* K^+

$$\forall \mathbf{M} \in \mathbb{R}^{m \times n}, \forall i \in [1, m], \forall j \in [1, n], \Pi_{K^+}(\mathbf{M}_{ij}) = \begin{cases} \mathbf{M}_{ij} \texttt{ if } \mathbf{M}_{ij} \geqslant 0 \\ 0 \texttt{ otherwise} \end{cases} \tag{19}$$

References

[1] J. Bobin, J.-L. Starck, Y. Moudden, M. J. Fadili, Blind source separation: The sparsity revolution, Advances in Imaging and Electron Physics 152 (1) (2008) 221–302.

[2] B. B. Biswal, J. L. Ulmer, Blind source separation of multiple signal sources of fMRI data sets using independent component analysis, Journal of Computer Assisted Tomography 23 (2) (1999) 265–271.

[3] D. Nuzillard, J.-M. Nuzillard, Application of blind source separation to 1-D and 2-D nuclear magnetic resonance spectroscopy, IEEE Signal Processing Letters 5 (8) (1998) 209–211.

[4] P. Comon, C. Jutten, Handbook of Blind Source Separation: Independent component analysis and applications, Academic Press, 2010.

[5] M. Zibulevsky, B. A. Pearlmutter, Blind source separation by sparse decomposition in a signal dictionary, Neural Computation 13 (4) (2001) 863–882.

[6] A. M. Bronstein, M. M. Bronstein, M. Zibulevsky, Y. Y. Zeevi, Sparse ICA for blind separation of transmitted and reflected images, International Journal of Imaging Systems and Technology 15 (1) (2005) 84–91.

[7] Y. Li, S.-I. Amari, A. Cichocki, D. W. Ho, S. Xie, Underdetermined blind source separation based on sparse representation, IEEE Transactions on Signal Processing 54 (2) (2006) 423–437.

[8] J.-L. Starck, F. Murtagh, J. M. Fadili, Sparse Image and Signal Processing: Wavelets, Curvelets, Morphological Diversity, Cambridge University Press, 2010.

[9] J. Bobin, J. Rapin, A. Larue, J.-L. Starck, Sparsity and adaptivity for the blind separation of partially correlated sources., IEEE Transactions on Signal Processing 63 (5) (2015) 1199–1213.

[10] Z. Koldovsky, P. Tichavsky, E. Oja, Efficient variant of algorithm FastICA for independent component analysis attaining the Cramér-Rao lower bound, IEEE Transactions on Neural Networks 17 (5) (2006) 1265–1277.

[11] M. Zibulevsky, Blind source separation with relative Newton method, in: Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Source Separation (ICA 2003), Nara, Japan, 2003, pp. 897–902.

[12] N. Gillis, F. Glineur, Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization, Neural Computation 24 (4) (2012) 1085–1105.

[13] J. Rapin, J. Bobin, A. Larue, J.-L. Starck, NMF with sparse regularizations in transformed domains, SIAM Journal on Imaging Sciences 7 (4) (2014) 2020–2047.

[14] P. Tseng, Convergence of a block coordinate descent method for nondifferentiable minimization, Journal of Optimization Theory and Applications 109 (3) (2001) 475–494.

[15] J. Bolte, S. Sabach, M. Teboulle, Proximal alternating linearized minimization for nonconvex and nonsmooth problems, Mathematical Programming 146 (1-2) (2014) 459–494.

[16] Y. Xu, W. Yin, A globally convergent algorithm for nonconvex optimization based on block coordinate update, arXiv preprint arXiv:1410.1386.

[17] N. Parikh, S. Boyd, et al., Proximal algorithms, Foundations and Trends® in Optimization 1 (3) (2014) 127–239.

[18] P. L. Combettes, J.-C. Pesquet, Proximal splitting methods in signal processing, in: Fixed-point algorithms for inverse problems in science and engineering, Springer, 2011, pp. 185–212.

[19] H. Raguet, J. Fadili, G. Peyré, A generalized forward-backward splitting, SIAM Journal on Imaging Sciences 6 (3) (2013) 1199–1226.

[20] E. Chouzenoux, J.-C. Pesquet, A. Repetti, A block coordinate variable metric forward–backward algorithm, Journal of Global Optimization 66 (3) (2016) 457–485.

[21] A. Patrascu, I. Necoara, Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization, Journal of Global Optimization 61 (1) (2015) 19–46.

[22] I. Toumi, B. Torrésani, S. Caldarelli, Effective processing of pulse field gradient NMR of mixtures by blind source separation, Analytical Chemistry 85 (23) (2013) 11344–11351.

[23] J.-L. Starck, J. Fadili, F. Murtagh, The undecimated wavelet decomposition and its reconstruction, IEEE Transactions on Image Processing 16 (2) (2007) 297–309.