University of Liège
Faculty of Applied Sciences
Academic year 2010 - 2011

Structured sparse principal component analysis for fMRImaging

Raphaël Liégeois

Master Thesis conducted to obtain
the degree of Master in Biomedical Engineering

Promoter :	Prof. Rodolphe Sepulchre
Jury :	Dr. Pierre Geurts
	Dr. Christophe Phillips
	Dr. Andrea Soddu
	Prof. Louis Wehenkel

June 17, 2011

Abstract

The use of component analysis on fMRI data allows to extract some interesting hidden components out of the data such as neuronal networks. While independent component analysis (ICA) is currently preferred in this application, we try to know whether sPCA could give better results than ICA when applied to fMRI data by highlighting the strengths and weaknesses of both techniques. Indeed, the neuronal networks are mostly intrinsically very sparse and it could thus be interesting to include explicitely this feature in the decomposition technique.

In the experimental section we first show on simulated fMRI data that in an ideal example of fMRI data, sPCA gives better results than ICA when the sparsity of the networks composing the simulated data is higher than approximately 80%. However, using the same model it appears that sPCA seems to be less robust than ICA to some perturbations that we can find in real fMRI data such as the motion of the patient during acquisition of the data.

We then use real fMRI data and we design three different experiments in order to evaluate the decomposition performed by both techniques. In each experiment ICA gives better results than sPCA. We can retain two important drawbacks of sPCA compared to ICA. First, the neuronal networks extracted through sPCA appear to be more affected by perturbations such as the motion of the patient, making the extraction of neuronal components from one subject to another less robust than with ICA. Second, sPCA does not seem to be able to isolate neuronal information in a few components only, whereas ICA does.

Acknowledgments

First I would like to thank all the members of the Jury of this Master Thesis. I particularly address my gratitude to my supervisor Prof. Rodolphe Sepulchre for his wise advice about the research directions to explore and his advised comments of the different results of this master thesis as well as to Dr. Andrea Soddu for his warm welcome, his availability, his support all along the project and for sharing his passion for research.

My deep thanks also go to Dr. Pierre Geurts, Dr. Christophe Phillips and Prof. Louis Wehenkel for accepting to be part of the Jury and for their commentaries about the first versions of this project that allowed me to improve this final report.

Finally, I want to express my special thanks to Mr. Sunny Mahajan for sharing his knowledge and his work on the topic of this master thesis.

(In French)

Je tiens à remercier mon père, ma mère et ma soeur qui m'ont apporté leur amour et un soutien indéfectible tout au long de mes études. Mes pensées vont aussi à Mandy qui a su rendre mes instants difficiles de ce travail juste un peu plus drôles.

J'ajoute dans ces remerciements mes amis des homes du Sart-Tilman qui m'ont permis de réaliser ce travail dans un contexte idéal de convivialité, de rêves, d'échanges, de plaisir. Merci à Maïlis, Mandy, Cécile, Laurent, Nico et Simon.

Enfin, je remercie Philou, Cédric, Kevin et Linh ainsi que mes amis du chalet : Amory, Loic, Yannick et ses idées un peu folles, Renaud, Karim et tous les autres pour les moments de ressourcement précieux qu'ils m'ont apportés ces derniers mois.

Contents

1 Introduction

The topic of this master thesis is composed of two active research topics in the domain of bioengineering and engineering. On the one hand *component analysis* is a powerful mathematical tool to analyze large datasets recovering several variants, some of which have been developed for more than a century. Through the years the approaches presented in the literature have evolved as well as applications or even nomenclature, making their understanding not always straightforward. On the other hand the biomedical facet of this work is the application of those techniques to the domain of neuroscience, more particularly to *functional Magnetic Resonance Imaging* (fMRI).

As suggested in the acknowledgements, another student, Mr. Sunny Mahajan, already got into this topic during an internship under Prof. R. Sepulchre's guidance. To make a clear separation between his work and this master thesis we explicitely mention in Appendix B the codes used in this work and written by Mr. Mahajan. The other algorithms and results are of course personal productions unless otherwise specified.

In the first part we develop different variants of component analysis, in particular *sparse* principal component analysis that has met an increasing interest in the recent years. We try to deeply understand and clearly present the techniques and we do our best to draw coherent parallels between them in order to build a global view of component analysis as well as a highlight of the specificities of each technique.

The second section theoretically presents fMRI data and explains to what extent component analysis techniques can be relevant to analyze such type of data. We try to emphasize the features of the component analysis techniques that are desirable in this application.

After completing the theoretical bases, the next chapters develop experimental approaches aiming to evaluate the relevance of each component analysis technique in fMRI data analysis with a particular focus on *sparse* principal component analysis. In the first part we try to keep a high level of abstraction in the experiments in order to derive the most general conclusions. This is done by the application of the component analysis techniques on a toy example. We then present several experiments looking for quantitative results and a way to verify the relevance of the first part of the experimental section.

In the conclusion sections we highlight the main results by gathering outcomes from the different sections and we give some ideas about future directions that could be studied.

2 Component Analysis

Finding an optimal representation of a set of data has been a key-question for a long time and in a very large range of domains going from economics to database treatment or medical imaging. This optimal representation has a dual aim : reducing the dimension of the dataset and/or finding any interesting structure hidden in the initial data.

Let us consider $x_i(t)$ a dataset with $i \in \{1, \ldots, m\}$ and $t \in \{1, \ldots, T\}$. This dataset can be considered as a set of m initial random variables with T observations for each of them. Given this dataset component analysis is looking for a better representation of the set according to one of the aforementioned criteria. For computational reasons and in order to build on all the results of matrix theory, the best representation is searched among the *linear* transformations of the data :

$$y_i(t) = \sum_j w_{ij} x_j(t) \text{ for } i = 1, \ldots, n \text{ and } j = 1, \ldots, m \tag{1}$$

where the w_{ij} are the coefficients that determine the linear transformation and n the number of variables in the new space of representation. We can rewrite (1) in a matrix form as follows :

$$\mathbf{Y}_{(n,T)} = \mathbf{W}_{(n,m)} \mathbf{X}_{(m,T)} \tag{2}$$

where $\mathbf{X}_{(m,T)}$ is the initial dataset in a m-dimensional space, $\mathbf{Y}_{(n,T)}$ is the new representation of the dataset in a n-dimensional space with $n \leq m$ since we want to reduce the dimensions of the data, and $\mathbf{W}_{(n,m)}$ is the matrix that expresses the base changement from the m to n-dimensional space.

The question is now to find an optimality criterion to determine $\mathbf{W}$. Depending on the criterion several variants of component analysis have been developped. I will here present two familiar techniques : principal component analysis (PCA), independent component analysis (ICA) and a more recent alternative to PCA : sparse principal component analysis (sPCA).

2.1 Principal Component Analysis

Starting from m possibly correlated pre-centered signals in $\mathbf{X}_{m,T}$, PCA aims to decorrelate those signals through an *orthonormal transformation*[1] of the initial data matrix, as

[1] An orthonormal tranformation is a multiplication by an orthogonal matrix

shown in (2). This technique also allows to reduce efficiently the dimensionality of the data ($n << m$), if wanted.

Remark : Two random variables x_1 and x_2 are said to be uncorrelated if we have :

$$E\{x_1x_2\} = E\{x_1\}E\{x_2\} \tag{3}$$

Since the data are pre-centered in this case this relation will become $E\{x_1x_2\} = 0$: x_1 and x_2 are orthogonal. ◇

We must here make things clear about PCA nomenclature. The question is to know what are the principal *components* in the matrix representation given in (2) : the lines of $\mathbf{W}_{n,m}$ or the lines of $\mathbf{Y}_{n,T}$? Some authors (see for example [7]) call the lines of $\mathbf{W}_{n,m}$ the principal *directions* whereas the lines of $\mathbf{Y}_{n,T}$ are called the principal *components*. This vocabulary seems quite consistent with the nature of the matrices we are dealing with. Indeed, as suggested in the first paragraph of this section the lines of $\mathbf{W}_{n,m}$ are *orthonormal*, they indicate directions. However, the large majority of authors ([43],[27]) call the lines of $\mathbf{W}_{n,m}$ the principal *components* without really giving a name to the lines of $\mathbf{Y}_{n,T}$. In order to use the most common terms we will adopt this nomenclature :

- The lines of $\mathbf{W}_{n,m}$ are the principal *components*.
- The lines of $\mathbf{Y}_{n,T}$ are the principal *projections*.

Let us motivate the choice of the word *projection*. The matrix product (2) has several interpretations. Geometrically, it can be seen [43] as a rotation and a stretching of the initial data matrix $\mathbf{X}_{m,T}$ into $\mathbf{Y}_{n,T}$. We can also see $\mathbf{Y}_{n,T}$ as the projection of $\mathbf{X}_{m,T}$ on the n first principal components. Indeed, if we call w_i, x_i and y_i the i^{th} line of $\mathbf{W}_{n,m}$, the i^{th} column of $\mathbf{X}_{m,T}$ and i^{th} line of $\mathbf{Y}_{n,T}$, respectively, (2) can be rewritten as :

$$\mathbf{Y}_{n,T} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} \begin{pmatrix} x_1 & x_2 & \cdots & x_T \end{pmatrix} = \begin{pmatrix} w_1x_1 & \cdots & w_1x_T \\ \vdots & \ddots & \vdots \\ w_nx_1 & \cdots & w_nx_T \end{pmatrix} \tag{4}$$

Each line of $\mathbf{Y}_{n,T}$ can thus be seen as the projection of the initial dataset on the corresponding principal component.

The dimensionality reduction has to be done while preserving as much information as possible. Naturally, information is related to variance for sufficiently high SNR signals.

The first principal component is thus the direction along which the data has maximum variance. The next principal components are obtained by looking for the component, orthogonal to all the previous principal components, along which the variance of the data is maximum.

Thus, after centering the data vectors x_i by substracting their means to simplify the computations, the first step is to maximize the variance of the first principal projection $y_1 = w_1^T\mathbf{X}$. Its variance is given by :

$$var(y_1) = E\{y_1^2\} = E\{(w_1^T\mathbf{X})^2\} = w_1^T E\{\mathbf{X}\mathbf{X}^T\}w_1 = w_1^T\mathbf{C}_x w_1 \tag{5}$$

where $\mathbf{C}_x$ is the $m \times m$ covariance matrix of $\mathbf{X}_{m,T}$. Since that amount increases indefinitely with w_1 we must impose a condition on the principal component w_1 to make that maximization problem relevant and we impose $w_1^T w_1 = 1$.

We have to maximize $w_1^T\mathbf{C}_x w_1$ subject to $w_1^T w_1 = 1$. Using Lagrange multiplier techniques ([27] pp.4-9) this optimization problem leads to maximize

$$w_1^T\mathbf{C}_x w_1 - \lambda(w_1^T w_1 - 1) \tag{6}$$

Taking the derivative of (6) with respect to w_1 gives

$$(\mathbf{C}_x - \lambda\mathbf{I}_m)w_1 = \mathbf{0} \tag{7}$$

where $\mathbf{I}_m$ is the $m \times m$ identity matrix. We see from (7) that the Lagrange multiplier is an eigenvalue of $\mathbf{C}_x$ related to the eigenvector w_1. There are thus as many local maxima as there are eigenvalues of $\mathbf{C}_x$. The quantity to be maximized is

$$w_1^T\mathbf{C}_x w_1 = w_1^T\lambda w_1 = \lambda w_1^T w_1 = \lambda \tag{8}$$

Since we want to find the maximum value of (8), w_1 must be the eigenvector of $\mathbf{C}_x$ related to the highest eigenvalue of $\mathbf{C}_x$. Furthermore the variance of the data expressed by that first component is λ.

It is further shown in [27] (Chap.1) that for $k > 1$ the k^{th} principal component is $w_k = \mathbf{e}_k$ where $\mathbf{e}_k$ is the eigenvector of $\mathbf{C}_x$ corresponding to the k^{th} largest eigenvalue of $\mathbf{C}_x$, and that the variance of the data along that principal component is equal to that eigenvalue. The rationale is the same as for the first component but it includes the additional condition that w_k must be orthogonal to all the previous components w_l with

$l < k$. We finally have :

$$\max var(y_k) = \max E\{y_k^2\} = \lambda_k \tag{9}$$

The first principal projection explains most of the variance in the data. Then, the k^{th} principal projection explains most of the variance of the data projected on the subspace formed by the $k-1$ first principal components. Keeping only the first few principal components is thus relevant only in the case of data with a SNR high enough. Indeed, if it is not the case the first components will account for noise and not for the signal. Here is an example with a high SNR signal :

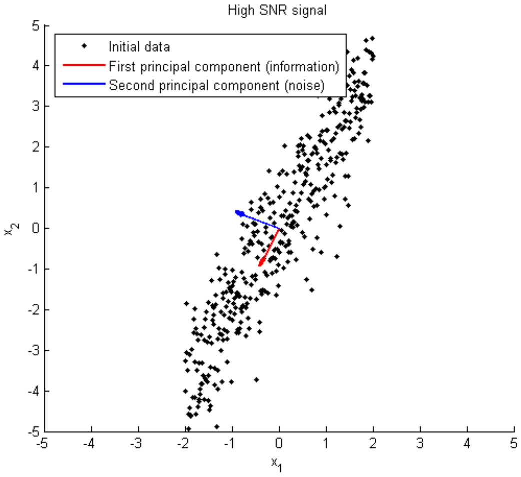

Figure 1: Principal components of a high SNR signal

The first principal component points along the expected informative direction and the second one goes in the direction of noise. In this example the second component could be omitted because it accounts for noise and does not give more information on the initial data. This example shows us that more than just reducing the dimensionality of the data, PCA also allows to eliminate noise. This can be done only if we know to what level of variance a noisy component corresponds.

Finally, we can check that doing so decorrelates the data, i.e. the principal projections are uncorrelated. Showing that amounts showing that the correlation matrix $\mathbf{C}_y$ (which is equal to the covariance matrix because the data are centered) of $\mathbf{Y}_{n,T}$ is diagonal :

$$\begin{aligned}
\mathbf{C}_y &= \frac{1}{T}\mathbf{Y}\mathbf{Y}^T && (10)\\
&= \frac{1}{T}(\mathbf{W}\mathbf{X})(\mathbf{W}\mathbf{X})^T && (11)\\
&= \mathbf{W}\mathbf{C}_x\mathbf{W}^T && (12)
\end{aligned}$$

And if we consider the fact that $\mathbf{W}$ is composed of the eigenvectors of $\mathbf{C}_x$ (hence the spectral decomposition of $\mathbf{C}_x$ is $\mathbf{C}_x = \mathbf{W}^T\mathbf{D}\mathbf{W}$ where $\mathbf{D}$ is a diagonal matrix containing the eigenvalues of $\mathbf{C}_x$ and $\mathbf{W}$ is an orthogonal matrix containing its eigenvectors), we have :

$$\begin{aligned}
\mathbf{C}_y &= (\mathbf{W}\mathbf{W}^T)\mathbf{D}(\mathbf{W}\mathbf{W}^T) && (13)\\
&= (\mathbf{W}\mathbf{W}^{-1})\mathbf{D}(\mathbf{W}\mathbf{W}^{-1}) && (14)\\
&= \mathbf{D} && (15)
\end{aligned}$$

where we also used the fact that $\mathbf{W}$ is orthogonal.

2.1.1 Link with the singular value decomposition

The singular value decomposition is a result from matrix theory that states that any matrix $\mathbf{X}_{m,T}$ can be rewritten as :

$$\mathbf{X}_{m,T} = \mathbf{U}_{m,r}\mathbf{L}_{r,r}\mathbf{A}_{r,T}^T \qquad (16)$$

where

- U and A are orthogonal matrices, that is $\mathbf{U}^T\mathbf{U} = \mathbf{A}^T\mathbf{A} = \mathbf{I}_r$
- L is a $(r \times r)$ diagonal ordered matrix containing the singular values
- r is the rank of $\mathbf{X}$

We can thus rewrite the covariance matrix as :

$$\mathbf{C}_x = \frac{1}{T}\mathbf{X}\mathbf{X}^T = \mathbf{U}\mathbf{L}^2\mathbf{U}^T \qquad (17)$$

And by comparing this to the spectral decomposition of any real symmetric matrix $A = U\Lambda U^T$ where U is an orthogonal matrix whose columns are the eigenvectors of A and Λ is a diagonal matrix containing the eigenvalues of A, we can say that the columns of U are the eigenvectors of $\mathbf{C}_x$ and L contains the square roots of the eigenvalues of $\mathbf{C}_x$. The columns of U are thus the directions of the principal components and each element of L is the standard deviation of the data along the corresponding direction.

2.1.2 Number of principal components

As explained in the introductory part of this section, it can be interesting to find d the optimal number of principal components to be selected. As we saw in (9) we can write $E\{y_k^2\} = \lambda_k$. The variances of the different principal components are expressed by the eigenvalues of $\mathbf{C}_x$ or, as shown in the previous section by the square of the singular values of the decomposition of $\mathbf{X}$, the data matrix.

Besides, the variance of the data explained by the principal components (or the singular values) often follows a characteristic curve :

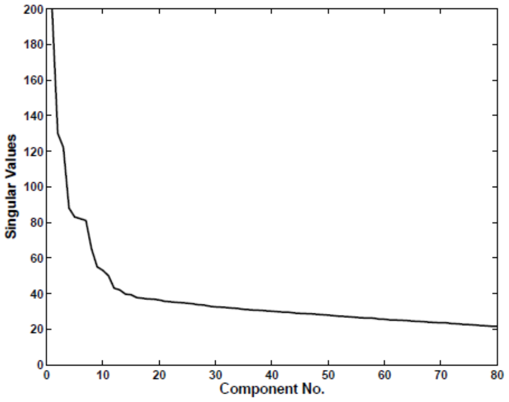

Figure 2: Variance for the ordered singular values, from [32]

Based on those results, a usual way ([25] pp 129-131, [32]) to determine d is to consider the number of components where the variance explained by the components (or value of the singular value of the data matrix) becomes quite constant, those components will account typically for noise. In this case d would be set at around twelve components.

2.1.3 Pros and cons of PCA

PCA is a powerful tool in data dimensionality reduction and deals well with noise in the case of high SNR signals. We here resume the main advantages and disadvantages of this approach :

- The new representation is a linear combination of *all* the initial variables. This can yield some difficulties in the interpretation of the results.
- PCA uses only statistical moments of order lower or equal to two. This leads to the hypothesis that the data can be described using only those quantities (mean and variance). This is not true in the case of non-gaussian data.
- The signal needs to have SNR high enough for the first components to describe the directions of interest and not noise.
- PCA gives a natural ordering of the components based on the amplitude of the variance explained.

2.2 Independent Component Analysis

Whereas PCA aims to find an uncorrelated representation of the data so that the new variables are orthogonal, ICA seeks to find a linear transformation of the data that makes the new variables as *independent* as possible.

Remark : Let us here briefly define statistical independence. Intuitively, it means that knowing one of the two variables does not give any information on the other variable :

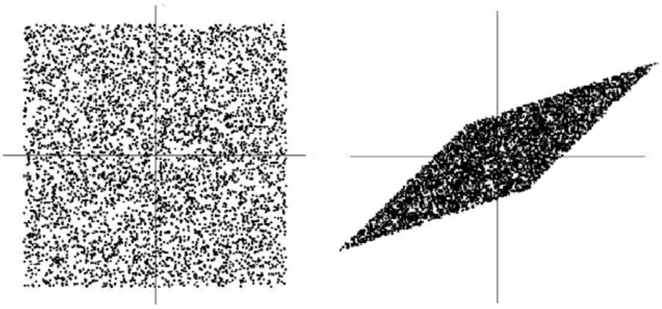

Figure 3: Example of independent variables (left) and dependent variables (right), from [26]

From a mathematical point of view, considering two random variables X and Y with $f_X(x)$ and $f_Y(y)$ their probability density functions, they are said to be statistically independent if and only if the combined variable X,Y has a joint probability density function

$f_{X,Y}(x,y)$ that is factorable in $f_{X,Y}(x,y) = f_X(x)f_Y(y)$. Another important property of independent random variables X and Y is that for two functions f and g we have [26]:

$$E\{f(X)g(Y)\} = E\{f(X)\}E\{g(Y)\} \tag{18}$$

If we take $f(X) = X$ and $g(Y) = Y$ we see that statistical independence of two variables implies uncorrelatedness of those variables (see equation (3) for a definition of uncorrelatedness). ◇

For the ICA approach we can also start from the matrix formulation given in (2). We will see further that ICA is not designed for dimensionality reduction, hence we can consider $m = n$ and write (2) as:

$$\mathbf{Y}_{(m,T)} = \mathbf{W}_{(m,m)}\mathbf{X}_{(m,T)} \tag{19}$$

If we relate this formulation to the classical cocktail-party problem, $\mathbf{X}_{(m,T)}$ would here be the m unknown sources, $\mathbf{W}_{(m,m)}$ the unknown *mixing* matrix and $\mathbf{Y}_{(m,T)}$ the m observations of the microphones in the cocktail-party problem. The goal is to recover the original signals $\mathbf{X}_{(m,T)}$ from the observations $\mathbf{Y}_{(m,T)}$ which is known as *blind source separation*. This can be done if we make some assumptions ([25] pp152-153) on the initial signals :

- They are statistically *independent*.
- They have *non-gaussian* distributions.

The original signals $\mathbf{X}_{(m,T)}$ are unknown and only $\mathbf{Y}_{(m,T)}$ is known. Practically we will thus start from $\mathbf{Y}_{(m,T)}$ to compute the independent components. If we suppose that $\mathbf{W}_{(m,m)}$ is invertible we can write :

$$\mathbf{X}_{(m,T)} = \mathbf{B}_{(m,m)}\mathbf{Y}_{(m,T)} \tag{20}$$

where $\mathbf{B}_{(m,m)}$ is the *unmixing* matrix and is equal to $\mathbf{W}^{-1}$. Considering (20) we may once again clarify the nomenclature used in this section :

- The lines of $\mathbf{X}_{(m,T)}$ are the *independent components*.
- The lines of $\mathbf{B}_{(m,m)}$ are the *unmixing components*.

2.2.1 Pre-Processing

The first preprocessing step is to basically center the observed data by substracting the mean. Once centering has been done, the second preprocessing step is *whitening* of the observed matrix, or in other words making its components uncorrelated and with unity variance. This step is done by linearly transforming the observed matrix $\mathbf{Y}$ by a whitening matrix $\mathbf{V}$ such that all the components of the whitened data matrix $\tilde{\mathbf{Y}}$ are uncorrelated and have unit variance. In other words that means that the covariance matrix of $\tilde{\mathbf{Y}}$ is equal to the identity matrix.

One method to do so is to take $\mathbf{V} = \mathbf{E}\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T$ ([25] p. 159) where $\mathbf{E}$ and $\mathbf{D}$ are the matrices resulting from the spectral decomposition of the covariance matrix of $\mathbf{Y}$: $E\{\mathbf{Y}\mathbf{Y}^T\} = \mathbf{E}\mathbf{D}\mathbf{E}^T$. We have then :

$$\begin{aligned} \tilde{\mathbf{Y}} &= \mathbf{V}\mathbf{Y} = \mathbf{V}\mathbf{W}\mathbf{X} && (21) \\ &= \tilde{\mathbf{W}}\mathbf{X} && (22) \end{aligned}$$

where $\tilde{\mathbf{W}}$ is the whitened mixing matrix. We can check that the covariance matrix of the whitened data is equal to the identity matrix :

$$\begin{aligned} E\{\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\} &= \mathbf{E}\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T E\{\mathbf{Y}\mathbf{Y}^T\}\mathbf{E}\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T && (23) \\ &= \mathbf{E}\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T\mathbf{E}\mathbf{D}\mathbf{E}^T\mathbf{E}\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T = \mathbf{I} && (24) \end{aligned}$$

using the fact that $\mathbf{E}$ is an orthogonal matrix and hence $\mathbf{E}^T\mathbf{E} = \mathbf{I}$. The interest of whitening comes from the fact that the new mixing matrix $\tilde{\mathbf{W}}$ is orthogonal (and thus has less degrees of freedom) :

$$E\{\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\} = \tilde{\mathbf{W}}E\{\mathbf{x}\mathbf{x}^T\}\tilde{\mathbf{W}}^T = \tilde{\mathbf{W}}\tilde{\mathbf{W}}^T = \mathbf{I} \qquad (25)$$

$\tilde{\mathbf{W}}$ is thus an orthogonal matrix. Hence, [26], its number of degrees of freedom is led from m^2 to determine all the parameters of $\mathbf{W}$ in (2) to $m(m-1)/2$ of $\tilde{\mathbf{W}}$ in (22).

Note that (20) can be rewritten as :

$$\mathbf{X} = \tilde{\mathbf{B}}\tilde{\mathbf{Y}} \qquad (26)$$

where $\tilde{\mathbf{B}} = \mathbf{B}\mathbf{V}^{-1}$ is the whitened unmixing matrix.

2.2.2 The fastICA algorithm

There are two widely used algorithms designed to perform ICA. The first one, *Infomax*, aims to maximize the information transferred from the mixed signals to the independent signals [4]. The second one, *fastICA*, maximizes non-gaussianity of the components to achieve statistical independence of the components [24]. We here describe this latter algorithm.

As just mentioned, the key idea of fastICA is to maximize non-gaussianity as a measure of independence. This link intuitively comes from the central limit theorem that states that under certain conditions the distribution of the sum of independent random variables tends to a gaussian distribution. From there, it can be shown ([25] pp. 166-167) that if we extract variables of the sum by maximizing their non-gaussianity we recover in fact the independent variables. This is also why we can not consider random variables with a gaussian distribution.

The question is now to estimate non-gaussianity. Several quantities can be used to do this :

- *Kurtosis* of the random variable y is defined by

$$kurt(y) = E\{y^4\} - 3E^2\{y^2\} \tag{27}$$

  If y has a gaussian distribution, we have $kurt(y) = 0$. For non-gaussian variables, the kurtosis is in general different from zero and when it is negative, the random variable is called *subgaussian* and in the other case, *supergaussian.*
  However, the fourth power in the expression (27) makes kurtosis highly non-robust to outliers and that's why another measure of non-gaussianity is sometimes used.

- *Negentropy* is based on the notion of entropy coming from information theory. An important result from this theory states that among all random variables of equal variance, the gaussian variable has the largest entropy. Negentropy of y is thus defined as the difference between the entropy (H) of a gaussian variable y_{gauss} with the same covariance matrix as y, and the entropy of y :

$$J(y) = H(y_{gauss}) - H(y) \tag{28}$$

  The difficulty here comes from the computation of entropy. A usual way to approximate negentropy is [26] :

$$J(y) \approx (E\{G(y)\} - E\{G(y_{gauss})\})^2 \tag{29}$$

where G is a non-quadratic function.

Once the preprocessing steps have been done, the goal is to find the *linear combination* of the data that has the highest negentropy. If we consider the formulation given in (26) we can denote by $\tilde{\mathbf{b}}$ a line of $\tilde{\mathbf{B}}$. $\tilde{\mathbf{b}}$ is thus what we call a *whitened unmixing component* and since all the $\tilde{\mathbf{b}}$ will be requested to be orthonormal, we can say that the "linear combination" mentioned before is in fact a projection of the data on $\tilde{\mathbf{b}}$: $\tilde{\mathbf{b}}^T\tilde{\mathbf{Y}}$.

The quantity we want to maximize is $J(\tilde{\mathbf{b}}^T\tilde{\mathbf{Y}})$. Considering (29) we see that we are looking for an optimum of $E\{G(\tilde{\mathbf{b}}^T\tilde{\mathbf{Y}})\}$ and as in the PCA approach, we impose a condition on the whitened unmixing component $\tilde{\mathbf{b}}$: $|\tilde{\mathbf{b}}| = 1$. Starting from there, using the Kuhn-Tucker conditions [31] to solve this constrained optimization problem and then the Newton Raphson's method [26], we obtain the fastICA fixed-point iteration that is here represented in a global view of the fastICA algorithm :

Algorithm 1 Global fastICA algorithm inspired from [47] and [26]

```
Centering of the data $\mathbf{Y}_{m,T}$
Whitening of the data
Random initialization of $\tilde{\mathbf{b}}$
for each whitened unmixing component $\tilde{\mathbf{b}}_i$ with $i = 1$ to $m$ do
    // Fixed point iteration
    $\tilde{\mathbf{b}}_i^+ = E\{\tilde{\mathbf{Y}}G(\tilde{\mathbf{b}}_i^T\tilde{\mathbf{Y}})\} - E\{g(\tilde{\mathbf{b}}_i^T\tilde{\mathbf{Y}})\}\tilde{\mathbf{b}}$;
    Decorrelation of $\tilde{\mathbf{b}}_i^+$ relatively to the previous vectors
    Normalization of $\tilde{\mathbf{b}}_i^+$
    If $\tilde{\mathbf{b}}_i^+$ did not converge, back to step 6
end for
De-Whitening of the whitened unmixing components
De-Centering of the independent components
```

where g denotes the derivative of G and decorrelation is achieved by projecting $\tilde{\mathbf{b}}_i^+$ on the subspace orthogonal to all the previous whitened unmixing components.

This detailed structure allows to answer two questions. The first one is related to a common result used to show the difference between the ICA and the PCA approaches :

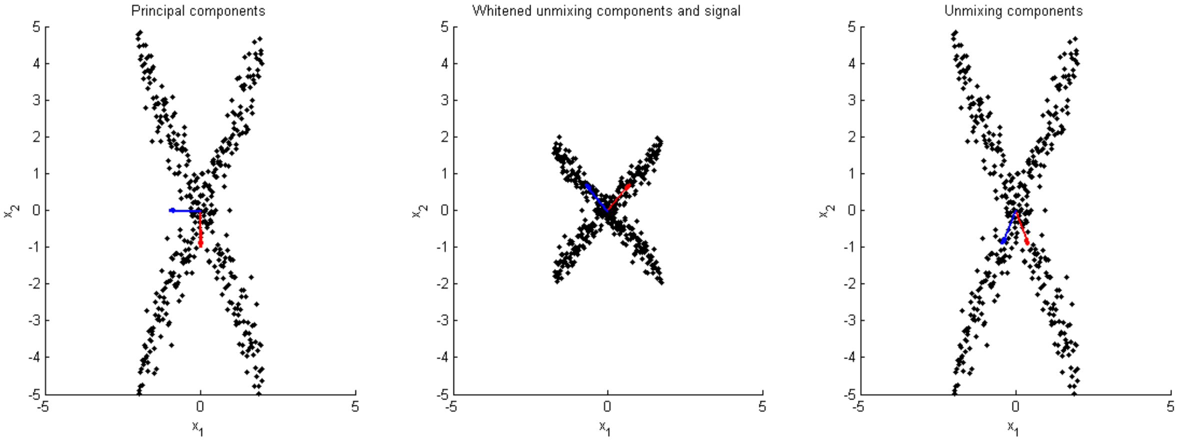

Figure 4: ICA vs PCA approach and whitening

On the first and the third pictures we see that the arrows match the data in the case of ICA but not in the case of PCA. Indeed, the PCA components must be orthogonal and this is why it is not possible for those to match such type of data. The question is then to understand why independent component analysis allows the arrows to be non-orthogonal (as shown before independence implies uncorrelatedness which implies othogonality for centered data). In fact this is just because the arrows represent the unmixing components that are not requested to be independent. However, we saw in *Algorithm 1* that the whitened unmixing components are requested to be orthogonal and that is why we represented the central figure. Indeed, the whitened unmixing components are orthogonal and match well the *whitened* signal. It is in fact the dewhitening step that allows the unmixing components to be non-orthogonal.

The second question is related to the fact that experimentally the independent components are sometimes non-orthogonal. And unlike the unmixing components, the independent components are supposed to be independent. This comes from the fact that independence is achieved by maximizing non-gaussianity through high-order statistical moments as shown in *Algorithm 1*. Indeed, we saw that the fixed point algorithm 1 releases on two approximations. The first one is the quantification of independence through non-gaussianity and the second one is the approximation of non-gaussianity by negentropy expressed in (29).

The following picture shows an example of behaviour of the fastICA algorithm for two components :

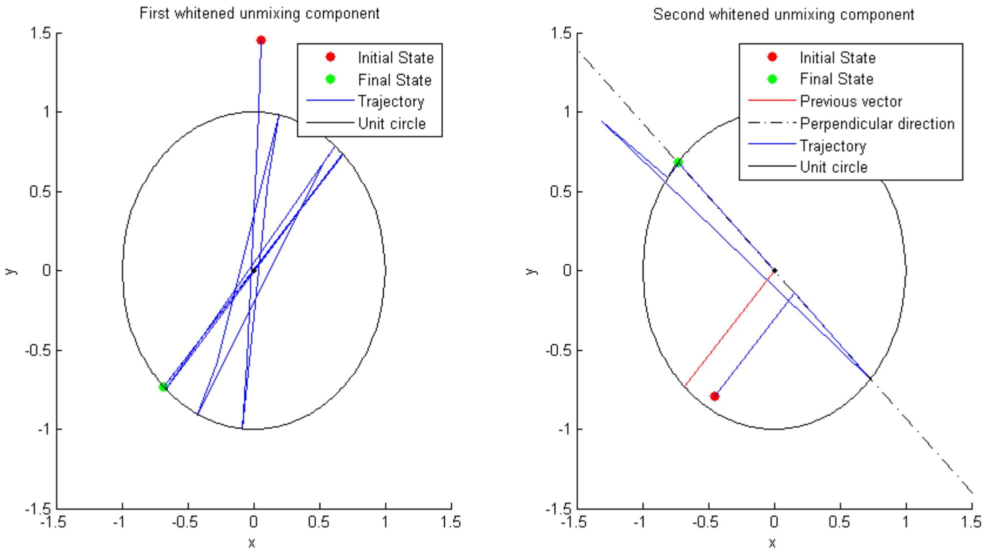

Figure 5: Example of fastICA iterations

On the left side of this figure, the first whitened unmixing component is computed. Since it is the first one, step 7 in algorithm 1 has no effect and after the fixed point iteration only normalization is performed that means projection on the unit circle. On the right side, the computations for the second whitened unmixing component are presented. The red point is the initial random unmixing component that is immediately normalized. The directions resulting from either the random initialization or the fixed point iteration are first projected on the subspace orthogonal to the first whitened unmixing component before being normalized.

2.2.3 Pros and cons of ICA

We have seen that the ICA approach has differences and complementarities with the PCA approach. We here summarize the main advantages and drawbacks of this approach.

- Contrarily to PCA, ICA deals with data that has non-gaussian distributions.
- ICA uses high-order statistical moments to achieve independence whereas PCA uses statistical moments of order one and two to achieve uncorrelatedness, that can be better or not depending on the type of data and the features in the data we want to highlight.
- FastICA does not provide the same solution for different runs because the initialization is random.

2.2.4 Limitations of ICA ?

We often read that a problematic limitation of ICA is that the components can be determined only up to a multiplicative constant (see for example [25] or [26]). Indeed since both **X** and **W** are unknown, a multiplication by a scalar of the source **X** could be cancelled by a division of the corresponding column of **W** making impossible to determine the variances of the different components. However, this limitation appears also in the PCA approach and we have to impose a constraint either on **X** or **W** as explained in section 2.1. We can do exactly the same in ICA (and fastICA does it), keeping in mind that considering the same constraint as in PCA : $w_1^T w_1 = 1$ still allows a multiplication of both **X** and **W** by -1.

Another limitation mentioned in [25] is that it is not possible to order the components as in PCA. Indeed the principal components are naturally ordered based on the variance they explain. As we have seen, the ICA algorithms do not use variance and in consequence it is not straightforward to use variance to classify the components. However, this could be done to measure an energy of each component (see [34]) but the question is then to know whether this classification is relevant and this question is still open in litterature [52]. We could for example use non-gaussianity of the components or alignment of the independent components with the unmixing components as a classification tool.

2.2.5 The fixed point iteration

This section was written because in our efforts to better understand the fastICA algorithm some interesting features appeared concerning the fixed point iteration.

We start from the same dataset as the one represented in figure 4. The iteration process to obtain the two whitened unmixing components is quite the same as the one showed on figure 5 :

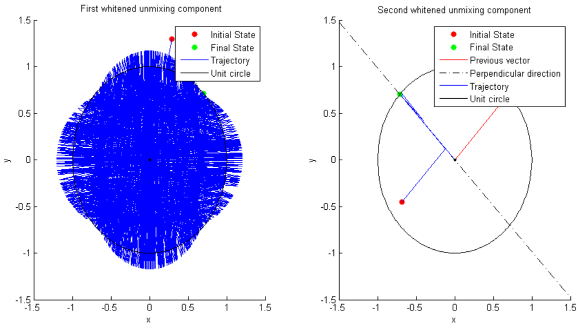

Figure 6: Fixed point iteration for the data of figure 4

However, the first component requested a high number of iterations and we can guess a shape in this left part of figure 6. As explained on the previous example the projection step does not change anything for the first whitened unmixing component since no component has been determined yet. Only remain the fixed point iteration (step 6) and the normalization step.

If we mark the outputs of step 6 with red dots, we get the next figure (the iterations for the second component are not represented anymore) :

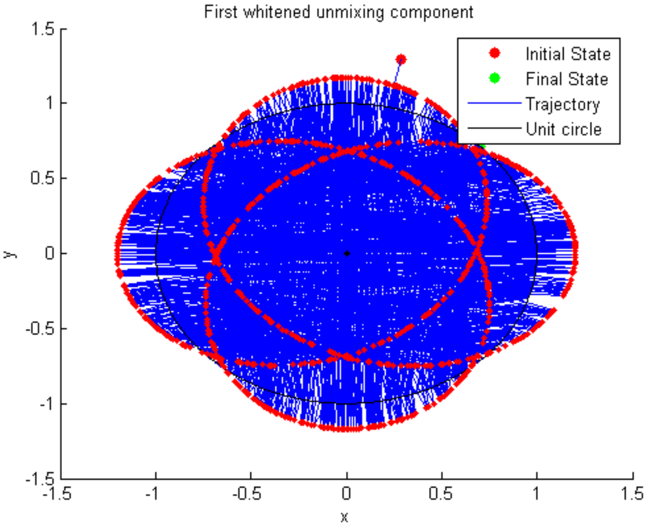

Figure 7: Outputs of step 6

There is definitely a link between the outputs of the fixed point iteration. Let us try to find out which analytical formula this curve corresponds to and why.

Let us denote by $\tilde{\mathbf{Y}}_{2,N}$ the whitened data considered in this case with N the number of points represented on figure 4. In the default configuration of fastICA, we have $G(y) = y^3$. The fixed point iteration (step 6 of Algorithm 1) can thus be written as :

$$\tilde{\mathbf{b}}^+ = \frac{\tilde{\mathbf{Y}}(\tilde{\mathbf{b}}^T\tilde{\mathbf{Y}})^3}{N} - 3\tilde{\mathbf{b}}E\{(\tilde{\mathbf{b}}^T\tilde{\mathbf{Y}})^2\} \tag{30}$$

Since $\tilde{\mathbf{Y}}$ has been whitened, its components are centered, orthogonal and of unit variance. Hence the following properties hold :

$$\left\{\begin{array}{ll} \frac{1}{N}E\{\tilde{\mathbf{Y}}_{i,.}\} = 0 & \text{for } i = 1,2 \\ \frac{1}{N}E\{\tilde{\mathbf{Y}}_{i,.}^2\} = 1 & \text{for } i = 1,2 \\ \tilde{\mathbf{Y}}_{1,.} \cdot \tilde{\mathbf{Y}}_{2,.} = 0 & \end{array}\right. \tag{31}$$

The second property implies that $\frac{1}{N}E\{(\tilde{\mathbf{b}}^T\tilde{\mathbf{Y}})^2\} = \tilde{\mathbf{b}}^T \frac{1}{N}E\{\tilde{\mathbf{Y}}_{i,.}^2\}\tilde{\mathbf{b}} = \tilde{\mathbf{b}}^T\tilde{\mathbf{b}} = |\tilde{\mathbf{b}}|^2 = 1$ and we finally get :

$$\tilde{\mathbf{b}}^+ = \frac{\tilde{\mathbf{Y}}(\tilde{\mathbf{b}}^T\tilde{\mathbf{Y}})^3}{N} - 3\tilde{\mathbf{b}} \tag{32}$$

Since at each iteration $\tilde{\mathbf{b}}$ is normalized before the fixed point iteration, this vector points towards a point on the unit circle. We will thus rewrite (32) using polar coordinates for $\tilde{\mathbf{b}}$: $\tilde{\mathbf{b}} = (\cos\theta, \sin\theta)^T$:

$$\begin{pmatrix} b_1^+ \\ b_2^+ \end{pmatrix} = \frac{1}{N}\begin{pmatrix} \sum_{i=1}^N y_{1,i}(y_{1,i}\cos\theta + y_{2,i}\sin\theta)^3 \\ \sum_{i=1}^N y_{2,i}(y_{1,i}\cos\theta + y_{2,i}\sin\theta)^3 \end{pmatrix} - 3\begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \tag{33}$$

Before developing the cubic terms, let us show that in this case we have :

$$\frac{1}{N}\sum_{i=1}^N y_{1,i}^3 y_{2,i} \approx \frac{1}{N}\sum_{i=1}^N y_{1,i} y_{2,i}^3 \approx 0 \tag{34}$$

We can see this on the central picture of figure 4. $y_{1,i}$ is the vector containing the x-coordinates of the points and $y_{2,i}$ contains their y-coordinates. Since the points are equally distributed around the two bissectors and N is quite large, we can say[2] that (34) holds. Experimentally we get a value of ≈ 0.01 allowing us to neglect those terms in our analysis, writing (33) as :

[2] It is tempting to try to demonstrate (34) with the strict equality to zero using the relations (31). In fact we can demonstrate this result for $N = 3$ but for $N = 4$ we have the following counter-example $y_{1,.} = (\frac{2}{\sqrt{50}}, \frac{-12}{\sqrt{50}}, \frac{6}{\sqrt{50}}, \frac{4}{\sqrt{50}})$ and $y_{2,.} = (\frac{2}{\sqrt{6}}, 0, \frac{2}{\sqrt{6}}, \frac{-4}{\sqrt{6}})$ showing that (31) does not always imply (34) with a strict equality to zero.

$$\begin{aligned}
\begin{pmatrix} b_1^+ \\ b_2^+ \end{pmatrix} &= \begin{pmatrix} \cos^3\theta \frac{1}{N}\sum_{i=1}^N y_{1,i}^4 + 3\cos\theta\sin^2\theta\frac{1}{N}\sum_{i=1}^N y_{1,i}^2 y_{2,i}^2 - 3\cos\theta \\ \sin^3\theta \frac{1}{N}\sum_{i=1}^N y_{2,i}^4 + 3\sin\theta\cos^2\theta\frac{1}{N}\sum_{i=1}^N y_{1,i}^2 y_{2,i}^2 - 3\sin\theta \end{pmatrix} \qquad (35)\\
&= \begin{pmatrix} \cos^3\theta q_1 + 3\cos\theta\sin^2\theta q_{12} - 3\cos\theta \\ \sin^3\theta q_2 + 3\sin\theta\cos^2\theta q_{12} - 3\sin\theta \end{pmatrix} \qquad (36)\\
&= \begin{pmatrix} \cos\theta((1-\sin^2\theta)q_1 + 3\sin^2\theta q_{12} - 3) \\ \sin\theta((1-\cos^2\theta)q_2 + 3\cos^2\theta q_{12} - 3) \end{pmatrix} \qquad (37)\\
&= \begin{pmatrix} \cos\theta(kurt(y_{1,.}) + \sin^2\theta(3q_{12} - q_1)) \\ \sin\theta(kurt(y_{2,.}) + \cos^2\theta(3q_{12} - q_2)) \end{pmatrix} \qquad (38)
\end{aligned}$$

where $q_1 = \frac{1}{N}\sum_{i=1}^N y_{1,i}^4$, $q_2 = \frac{1}{N}\sum_{i=1}^N y_{2,i}^4$, $q_{12} = \frac{1}{N}\sum_{i=1}^N y_{1,i}^2 y_{2,i}^2$ and $kurt(y_{i,.}) = q_i - 3$ is the kurtosis of the i^{th} component.

If we plot the curves given in (38) on the previous picture, we get the result of figure 8 :

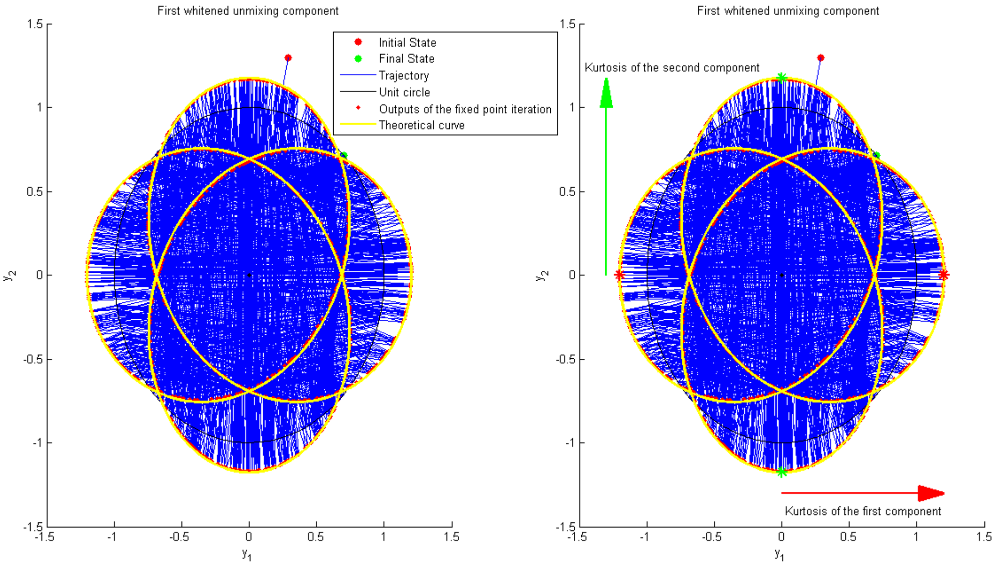

Figure 8: Theoretical curve and graphical interpretation of the kurtosis of the two components

It is easy to check that the kurtosis of the first component can be read on the red stars and the kurtosis of the second component on the green stars plotted on the right picture.

We will stop here the investigations on that topic because it is not directly related to the object of this master thesis. However, it could be interesting to study for example in which conditions the assumption (34) is valid or how the kurtosis of the whitened components $\tilde{\mathbf{Y}}_i$ influences convergence of fastICA.

2.3 Sparse Principal Component Analysis

When considering a traditionnal PCA approach for the problem (2), the new variables are linear combinations of *all* the initial variables. That makes the interpretation of the results sometimes difficult. In response to that drawback a variant of PCA for which the new variables are linear combinations of only few of the original ones has arisen : *sparse* PCA (sPCA).

Remark : a vector $\boldsymbol{a} = (a_1, \cdots, a_N)$ *is sparse if a large number of its entries* a_i *are equal to zero. Note that this notion is basis-dependent.* ◇

The notations of this section are inspired from [28] because we will use the algorithm presented in that paper. The initial formulation of (2) is slightly different than the one adopted for PCA and ICA and it is important to notice that when using the matlab codes related to that article for example. In [28] the matrix formulation is in fact the transpose of (2) :

$$\mathbf{Y}_{(T,n)} = \mathbf{X}_{(T,m)}\mathbf{W}_{(m,n)} \tag{39}$$

where the columns of $\mathbf{W}$ are the *sparse principal components* (that we denote w) and the columns of $\mathbf{Y}$ the principal projections.

The PCA optimization problem (5) can be rewritten [28] as :

$$\text{Find } w^* = \arg_w \max_{ww^T \leq 1} w^T \mathbf{C}_x w \tag{40}$$

where $\mathbf{C}_x$ is the sample covariance matrix of the data matrix $\mathbf{X}_{T,m}$.

In the sPCA single-unit approach this optimization problem is modified and contains a penalization term to induce sparsity in w [28] :

$$\phi_{l_1}(\gamma) = \max_{w \in B^m} \sqrt{w^T \mathbf{C}_x w} - \gamma ||w||_1 \tag{41}$$

where B^m is the unit ball, l_1 indicates that the norm-1 of w is used (l_0 is also possible, norm-0 or cardinality of w is then used) and γ is the *sparsity-controlling* parameter.

In fact there are four formulations of the algorithm and hence of the optimization problem (41) : using single-unit or block sPCA and using the l_1 or l_0 norm. We here present the case of a single-unit approach using l_1 norm in order to understand the sPCA approach.

The solution of the opimization problem (41) is denoted by $w^*(\gamma)$. If γ is equal to zero, w^* is the first principal component of the data matrix $\mathbf{X}$ with (a priori) no sparsity in w^*. On the other hand, the following relation shows that if γ is too high the solution $w^*(\gamma)$ is zero :

$$\max_{w\neq 0} \frac{||\mathbf{X}w||_2}{||w||_1} = \max_{w\neq 0} \frac{||\sum_i w_i x_i||_2}{||w||_1} \leq \max_{w\neq 0} \frac{\sum_i |w_i| \cdot ||x_i||_2}{\sum_i |w_i|} = \max_i ||x_i||_2 = ||x_{i^*}||_2 \tag{42}$$

Thus, if $\gamma > ||x_{i^*}||_2 = \gamma_{max}$ we have $\phi_{l_1}(\gamma) < 0$ for all $w \neq 0$ and $w = 0$ would then be a better solution because in that case we have $\phi_{l_1}(\gamma) = 0$. So if $\gamma = 0$ we have no sparsity in w^* and if $\gamma > \gamma_{max}$ we have total sparsity in w^* (since in that case $w^* = 0$). From now on we will thus consider $0 < \gamma < \gamma_{max}$ and we see that this parameter can be adjusted to induce more or less sparsity in w^*.

Let us come back to the optimization problem (41). This function is neither convex nor concave. Hence, we will relax the constraints on the problem to make it convex [28] :

$$\begin{aligned}
\phi_{l_1}(\gamma) &= \max_{w\in B^m} ||\mathbf{X}w||_2 - \gamma||w||_1 && (43)\\
&= \max_{w\in B^m} \max_{z\in B^T} z^T\mathbf{X}w - \gamma||w||_1 && (44)\\
&= \max_{z\in B^T} \max_{w\in B^m} \sum_{i=1}^{m} w_i(x_i^T z) - \gamma|w_i| && (45)\\
&= \max_{z\in B^T} \max_{\bar{w}\in B^m} \sum_{i=1}^{m} |\bar{w}_i|(|x_i^T z| - \gamma) && (46)
\end{aligned}$$

with $\bar{w}_i = \text{sign}(x_i^T z)w_i$. If we fix some $z \in B^T$ such that $|x_i^T z| > \gamma$ the inner maximization problem has the following solution :

$$w_i^*(\gamma) = \frac{\text{sign}(x_i^T z)[|x_i^T z| - \gamma]_+}{\sqrt{\sum_{k=1}^m [|x_k^T z| - \gamma]_+^2}}, \text{ for } i = 1, \cdots, m \tag{47}$$

This formulation leads to the final optimization problem, by replacing w_i^* in (46) and then taking the square of this relation:

$$\phi_{l_1}^2(\gamma) \quad = \quad \max_{z \in S^T} \sum_{i=1}^m [|x_i^T z| - \gamma]_+^2 \tag{48}$$

This function is convex which has two interesting consequences : first the search space for z is lead from a T-dimensional ball to a sphere S^T in a T-dimensional space and second we can now apply efficient gradient methods to solve this problem, as explained in the next section. Moreover, the search space is interestingly reduced in the common case where $T << m$, compared to (41).

The solution x^* of (48) induces sparsity in the component w. Indeed, considering (47) the elements w_i of w are different from zero only if $|x_i^T z| > \gamma$, the other elements of w being zero. Hence, the degree of sparsity in w increases with the value of γ.

2.3.1 Algorithm for sPCA

This section presents a gradient method for maximizing a convex function on a compact set. This method leads to the developement of an algorithm for sPCA with two steps :

1. A *pattern-finding* step that consists of distinguishing the entries of w that are zero from the others
2. Considering the non-zero entries of w, performing a classical principal component analysis on the corresponding subdata of $\mathbf{X}$

The optimization problem can be written as :

$$f^* = \max_{y \in Q} f(x) \tag{49}$$

where f is the convex function and Q the compact set. In the particular case of Q being a sphere of radius r, it is shown in [28] that the fixed point iteration af the gradient method can be expressed as :

$$x_{k+1} = r \frac{\nabla f(x_k)}{||\nabla f(x_k)||} \tag{50}$$

In our case $r = 1$ and the gradient method algorithm designed to determine the patterns can be written :

Algorithm 2 Single-unit sPCA method with l_1-penalty [28]

```
input :  Data matrix $\mathbf{X}_{T,m}$
         Sparsity-controlling parameter $\gamma \geq 0$
         Initial iterate z $\in S^T$
output : A locally optimal sparsity pattern P
repeat
  $z = \sum_{i=1}^{m} [|x_i^T z| - \gamma]_+ \text{sign}(x_i^T z) x_i$
  $z = z/||z||$
until a stopping criterion is satisfied
Construct a vector $P \in R^m$ such that $\begin{cases} p_i = 1 & \text{if } |x_i^T z| - \gamma > 0 \\ p_i = 0 & \text{otherwise} \end{cases}$
```

Once the pattern has been filled in we have to compute non-zero entries by maximizing the variance of the initial data matrix explained by those entries, as in classical PCA. This can be done [28] by solving the optimization problem :

$$(\mathbf{Z}^*, \mathbf{W}^*) = \arg \max_{\mathbf{Z} \in S_n^T, \mathbf{W} \in [S^m]^n} \text{Tr}(\mathbf{Z}^T \mathbf{X} \mathbf{W} \mathbf{N}) \tag{51}$$

where $\mathbf{N}$ is a diagonal matrix composed of strictly positive terms. This optimization problem [28] has a solution in the single-unit case :

$$\begin{aligned} \mathbf{Z}^* &= u && (52) \\ \mathbf{W}^* &= v \text{ for the non-zero entries of } \mathbf{W}^* && (53) \end{aligned}$$

where u and v result from the rank one singular value decomposition of the matrix $\mathbf{X}_P$, a submatrix of the data matrix $\mathbf{X}$ from which the columns related to the zero entries have been removed. In other words we perform here classical PCA on the data, ignoring the variables that have been judged not significant enough in the pattern-finding step.

Finally, a classical deflation scheme [28] is used to compute the following components by ensuring that the consecutive sparse principal components are orthonormal [16].

2.3.2 Pros and cons of sPCA

The same conclusions as the one done for PCA hold here except for two points :

- The new variables are linear combinations of only few of the original variables which simplifies interpretation.
- As in the case of ICA, different runs of this algorithm can result in different outputs since we try to maximize a convex function and there is a random initialization step.

3 Component analysis in fMRI data

In this section we present the theoretical background related to fMRI data and how component analysis can be applied to such type of data.

3.1 fMRI data and BOLD signal

Functional magnetic resonance imaging (fMRI) is a widely used technique to explore the functional behaviour of the brain. It allows to highlight regions of the brain that have more or less neuronal activity during a task or at rest. Here is an example of raw fMRI data obtained with a patient at rest and at the MNI coordinates[3] (0,-22,-26) :

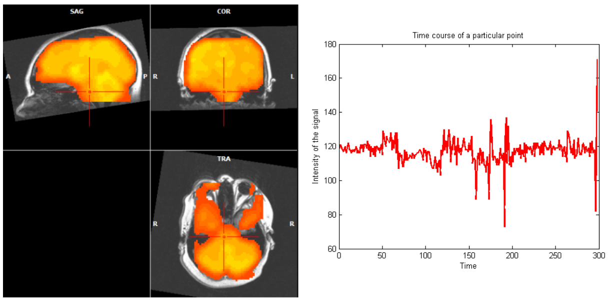

Figure 9: Example of fMRI data. Left : intensity of the signal at several points for one time sample. Right : intensity of the signal as a funtion of time, for one point in the brain

The dimensions of the data are the following :

- 298 time samples.
- $58 * 40 * 46 = 106720$ voxels of size $3 * 3 * 3\ mm^3$.

fMRI uses the magnetic properties of an element contained in the brain : *deoxyhemoglobin*. It is known that neuronal activity is related to the blood oxygenation in the brain [42]. In regions of high neuronal activity, the consumption of glucose and oxygen is modified and that results in a significant change in the ratio of hemoglobin to deoxyhemoglobin in the local blood flow. Since those two elements have different magnetic properties [37], it is possible to measure a signal through magnetic resonance : the BOLD (Blood Oxygen Level Dependent) signal.

[3] The Montreal Neurological Institute coordinates system was designed to facilitate the comparison between different brains. It allows to characterize the brain structures independently from their particular sizes or global shapes.

However, the metabolic processes related to neuronal activity seem to be quite complex and the relevance of the BOLD signal is questioned by some authors [15], [29]. Others argue that BOLD signal is sometimes affected by non-neuronal activity [1], or that the interpretation of positive and negative BOLD responses is difficult [44].

We will thus make our analysis considering that the BOLD signal reflects satisfactorily the neuronal activity in the brain, keeping in mind the pre-mentioned limitations.

3.2 Neuronal networks

Starting from data such as the one represented in figure 9 the idea is now to decompose that total signal into several correlated signals called *neuronal networks*. [38] gives the following definition : "neuronal networks are distant regions that share anatomical connections and functional interactions". From a mathematical point of view, the total initial data can be represented by a matrix S_{T,N_V} where T codes for the time and N_V for the 3-dimensional space. Figure 9 (*left*) is for instance the representation of the tenth line of S because it represents the functional activity of the patient at the 10^{th} time sample. The right part of this same figure is a representation of a column of S.

The global functional structure of the brain has been widely discussed but we can extract two important features to qualify its organization [51] : *segragation* and *integration*. The first one refers to the fact that the different parts of a same network are sometimes dispersed all over the brain whereas the latter refers to the unification of those parts.

The total signal is thus supposed to be the superposition of all the neuronal networks (visual, sensorimotor, default mode, dorsal attentional, etc.) and noise. An example of network, the *default mode* network, is shown on the following picture :

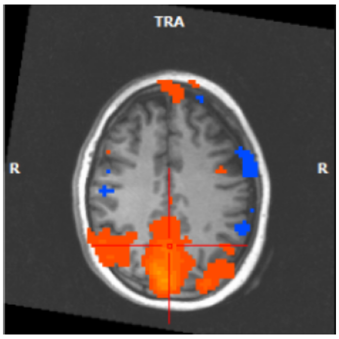

Figure 10: Default Mode network

The default mode network has been a quite discussed topic in the recent years but it is now agreed that this network is relevant [39],[41] and [30]. It is active when the brain is at rest and deactivates when the brain is involved in a specific task, confirming the fact that the energy consumption of the brain is the same whether it is involved in a task or not [40].

The sparsity of the default mode network varies among the subjects between $\approx 70\%$ and $\approx 80\%$.

3.3 Statistical tools

To perform the wanted decomposition and bring neuronal networks out of the initial data, two categories of approaches were used. The first one needed to include prior information on the network (correlation methods) whereas with the development of powerful algorithms unsupervised methods can now be used to extract neuronal network without any information on those networks.

3.3.1 Correlation methods

These methods compute correlation of the whole map with a specific region of the brain. This initial region or voxel, often called *seed* region is delimited by a specialist. After that, by computing the correlation of other regions of the brain across time, it is possible to detect other regions belonging to the same network [6],[23].

3.3.2 Unsupervised methods

With the oncoming of efficient algorithms to perform different kinds of component analysis or clustering, the use of those tools was prefered because it allows to consider all the brain when looking for networks and not only focusing on a region. Also, the use of unsupervised methods does not request prior information on the networks we want to study. We can mention here hierarchical clustering presented in [13], K-means [22], principal component analysis [2], [50] or [53], independent component analysis [35],[10] or [3] and more recently sparse principal component analysis [45] or [46].

For the component analysis techniques (PCA, ICA and sPCA), we can see it as a decomposition[4] of the initial matrix S_{T,N_V} :

$$S_{T,N_V} = TC_{T,N_C} * C_{N_C,N_V} \tag{54}$$

[4] This decomposition is a *spatial* decomposition since the lines of C represent spatial maps. We will see in sections 3.4.1 and 4.3 that it is also possible to perform *temporal* decompositions.

where T is the number of time samples, N_V is the number of voxels and N_C is the number of components extracted and :

- TC is the time courses matrix, each of the N_C columns is a time course
- C is the components matrix, each of the N_C lines is a component (a potential network)

We can further decompose this product in a sum of N_C terms, each term being the product of a network and its corresponding time course. This is shown on the following picture :

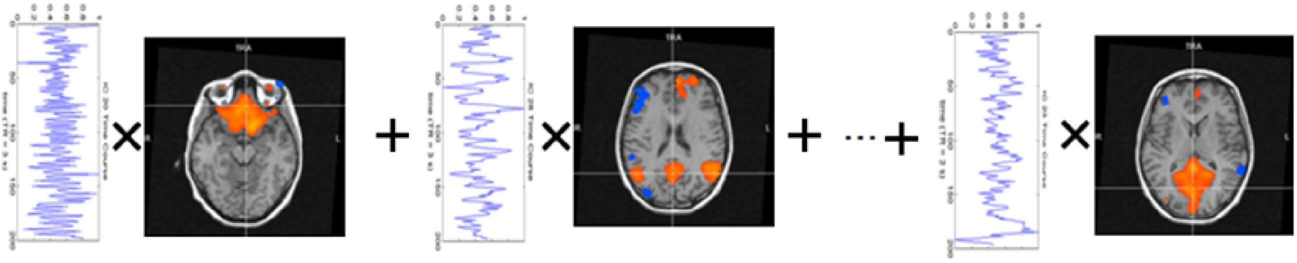

Figure 11: Decomposition in a sum of N_C terms, from [48]

This reformulation is important because it highlights the correspondance between a network and its time course. Indeed, we know that the frequency spectrum of the time courses of neuronal networks have some particular properties, this will be discussed in the next section.

Concretely, the matrix of the time courses is obtained by a matricial multiplication between the initial signal S_{T,N_V} and the inverse[5] of the components matrix C_{N_C,N_V}.

3.3.3 What is neuronal ?

When looking at the decomposition given on figure 11 it could be interesting to determine a criteria telling whether a component is a *neuronal* component (due to neuronal activity) or not. This could a priori be determined based on the spatial map or on the related time course.

According to [21] or [5] the criteria is based on the power spectrum of the time course related to a component. If the frequency bands between ≈ 0.01 Hz and ≈ 0.1 Hz have

[5] In matlab we use the function `pinv()` since C_{N_C,N_V} is not a square matrix.

high contributions the component is likely to be due to neuronal activity.

3.4 Why is ICA commonly used ?

For a few years independent component analysis has been the most used tool to study fMRI data. Two arguments seem reasonable to explain this fact : first the ICA approach allowed to extract easily recognizable networks and in a quite robust manner i.e. the same networks were obtained among plenty of subjects [38]. The second is related to the validity of two hypothesis underlying the use of ICA : the components are independent and added linearly. These hypothesis have been shown to be relevant in the case of fMRI data [36].

3.4.1 Spatial or Temporal independence ?

Coming back to section 2.2, we can see that the usual way to present ICA is by using *temporal* ICA (TICA). It is for example the case in the cocktail-party problem where the data matrix is composed of few signals over a large number of time samples. FastICA then aims to maximize non-gaussianity of the *lines* of **X** as represented in (20), the *independent components.* Non-gaussianity is computed for each independent component along a temporal subscript, that is why we call this variant *temporal* ICA.

However, in our case we have proportionally a small number of time samples (≈ 300) with respect to the number of voxels (≈ 100000). In order to compute independence on the largest number of elements we will use *spatial* ICA (SICA) that basically just transposes the data matrix :

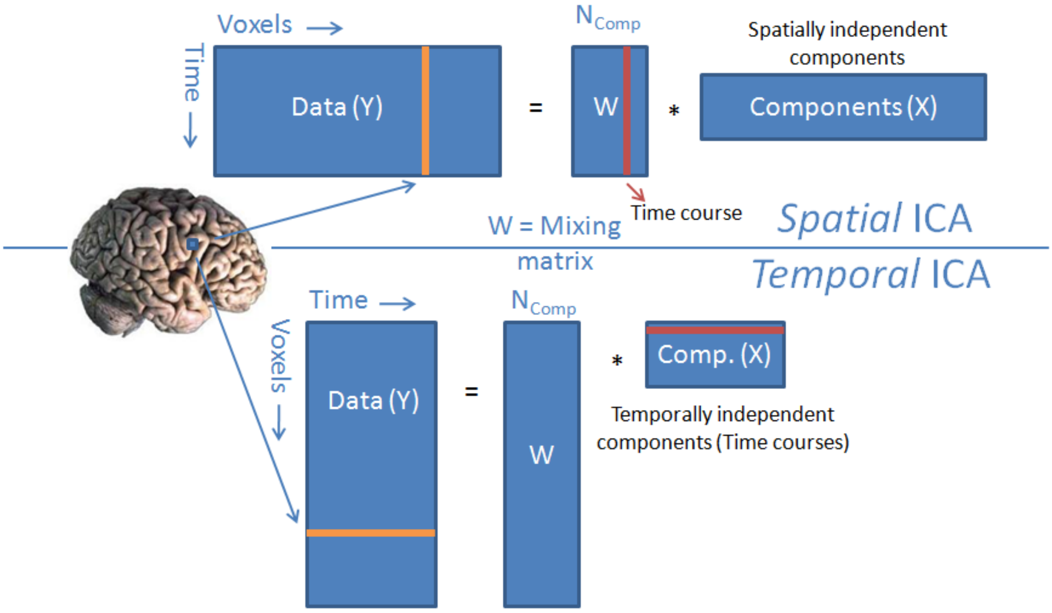

Figure 12: SICA vs. TICA inspired from [10]

3.5 Could sPCA give better results than ICA ?

In the more recent years, the sparse PCA approach has become a potential competitor of ICA. The main reason is that the networks, such as the one shown in figure 10, are usually intrinsically highly *sparse* and sPCA algorithms manage to "isolate relevant sparse effects" [46]. Moreover, a recent paper [17] argues that ICA works well because the "true" components are sparse and sparse vectors are often close to independance. So ICA algorithms work well only indirectly due to the independence of the components, the initial reason being that the components are sparse. Therefore, always according to these authors, the algorithms used in fMRI data analysis should include sparseness as a crucial factor.

4 Simulation of fMRI data and decomposition

The difficulty of evaluating the results given by either sPCA or ICA algorithms on real fMRI data comes from the fact that the true networks are unknown. Therefore, as in [8] for example we create a toy example to simulate fMRI data and study the results given by the algorithms knowing the true initial networks. To simplify visual analysis of the results we consider a 2-dimensional case with the same properties as the real fMRI data : 300 time samples, and a 58*40 pixels map. As suggested in the formulation given in (54) we build S_{T,N_V} by explicitly computing the matrix product $TC_{T,N_C} \cdot S_{N_C,N_V}$ with $T = 300$, $N_V = 58 \times 40 = 2320$, and with four components ($N_C = 4$).

Reexpressing this product as suggested in figure 11, our example is in fact a superposition of four components each one associated with a specific time course. The total signal is thus composed of a cross network associated with a square time course, mimicking a task related activation. The second network is a line following a sawtooth time course, the third network is a corner varying with a random time course and the last network is a disk with sinusoidal fluctuations as can be seen on the following picture :

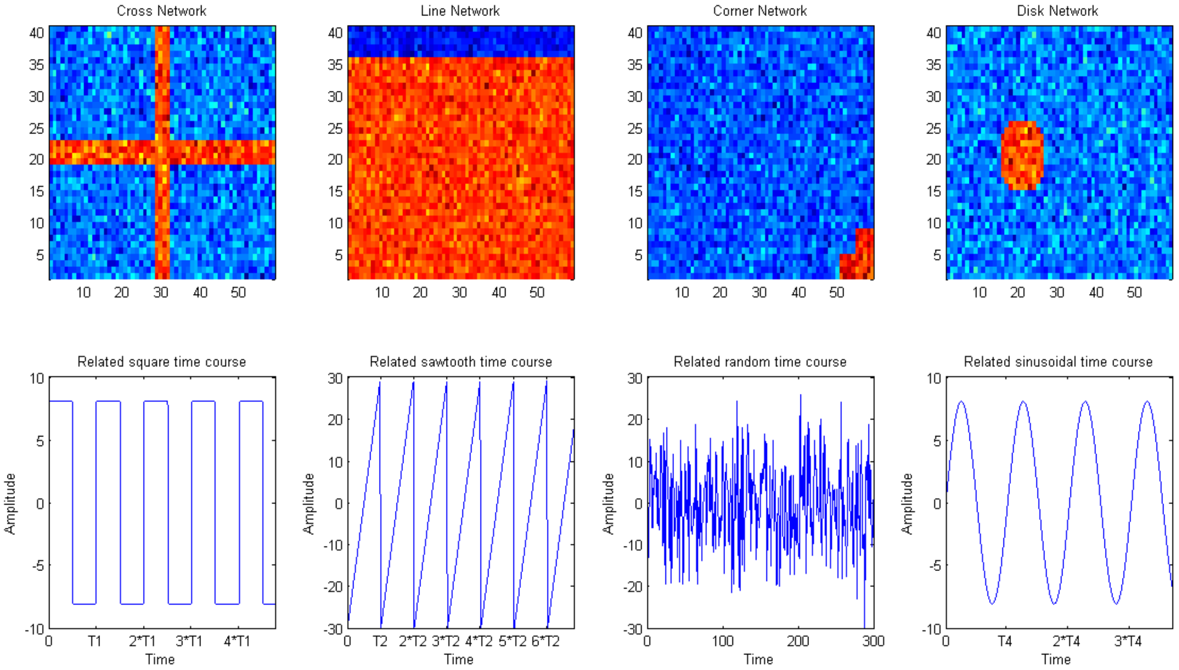

Figure 13: Composition of the total signal

Some noise has been added to the networks[6] and we will see in a further section that we can also simulate motion of the patient or the presence of a global physiological related signal.

The total signal is the sum of those four networks each one following a different time course. The classical approach was used to extract the components, either by ICA or sPCA :

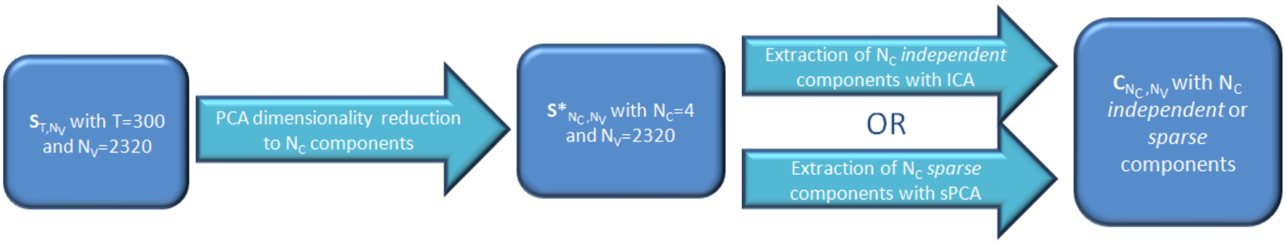
S_{T,N_V} with T=300 and N_V=2320
PCA dimensionality reduction to N_C components
$S^*_{N_C,N_V}$ with N_C=4 and N_V=2320
Extraction of N_C *independent* components with ICA
OR
Extraction of N_C *sparse* components with sPCA
C_{N_C,N_V} with N_C *independent* or *sparse* components

Figure 14: Approach to extract components

It is common to use PCA as a dimensionality reduction step (see for example [11], [49], [18] or [12]) using the variance explained by the components to retain only the ones that explain most of the variance in the initial data. Then, ICA or sPCA is used to achieve independence of the components in the first case or to induce sparsity in the components in the second case. The number of components is chosen manually, in this case it is naturally chosen equal to four, and the analysis of the singular values can help in this choice as explained in section 2.1.1. Note that it is also possible to determine automatically the number of components to be selected in the dimensionality reduction step (see [14] or [9]).

4.1 Default configuration and results

Here are the characteristics of the default configuration :

- The networks are very sparse to allow an optimized result for the sPCA algorithm.
- T_1, T_2 and T_4, the periods of the time courses represented on figure 13 are chosen such that they guarantee the independence between the initial networks. Indeed if one of those periods is equal to or a multiple of another period, we can experimentally

[6] A video of this model can be displayed by running the Matlab code `toy_example` (see Appendix B).

observe that the ICA algorithm won't work efficiently. We thus choose sufficiently large coprime numbers $T_1 = 2\pi \cdot 5$, $T_2 = 2\pi \cdot 7$ and $T_4 = 2\pi \cdot 13$ to ensure there is no repetition of the same template in the 300 time samples.

- The amplitudes of the default time courses are 8, 30, 10 and 8 respectively for the four networks described in figure 13. It is to note that the amplitudes do not influence the results of the algorithms, as long as they are significantly larger than noise.
- A gaussian noise of mean zero and unity variance is added to each network.
- The number of components extracted in the PCA dimensionality reduction phase is equal to the number of networks : four.
- The values of the sparsity parameters are 0.01 and norm l_1 is used.

Here are the results after the different steps of the approach exposed in figure 14. If we apply a PCA algorithm only, we obtain the following results for the four extracted components :

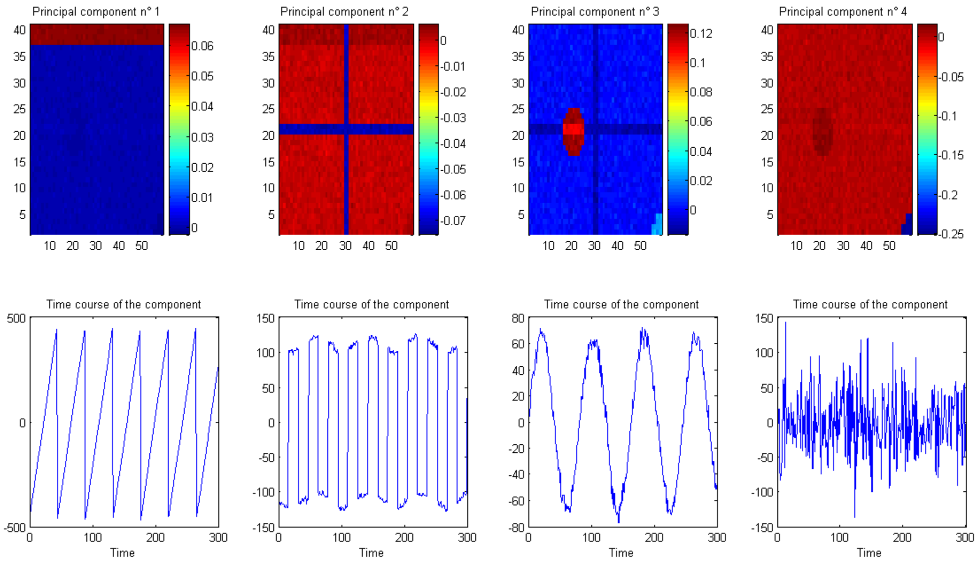

Figure 15: Networks and related time courses after PCA only

If we then apply the ICA algorithm the result is the following :

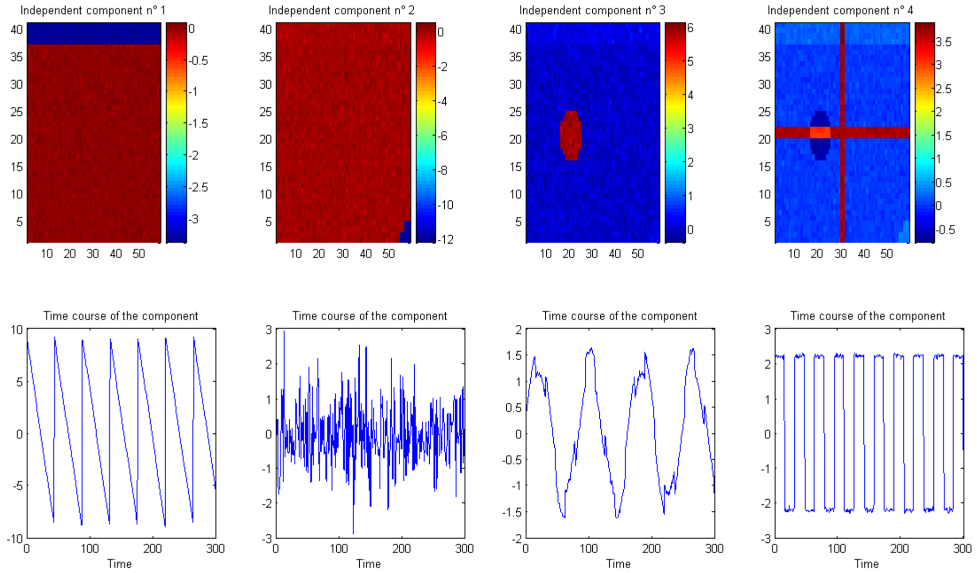

Figure 16: Networks and related time courses after PCA followed by ICA

And if we apply sPCA instead of ICA after the dimensionality reduction step we get :

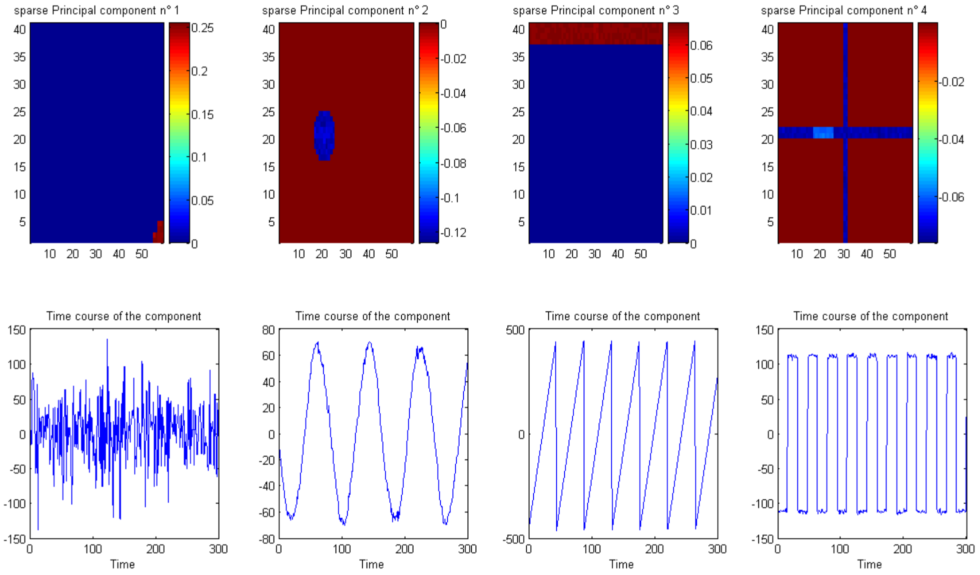

Figure 17: Networks and related time courses after PCA followed by sPCA

The results are significantly better if sPCA or ICA is applied after PCA. We can see it clearly in the time courses : in the PCA case the square time course seems to be affected by the sinusoidal time course and by some noise. The sinusoidal time course is also quite noisy and the second step of the treatment must thus be done to obtain the best results.

We will now focus our attention on the comparison between the ICA and sPCA algorithms. By comparing the last two figures, we can see two main differences between the two approaches :

1. The sPCA algorithm seems to eliminate noise whereas the independent components are still noisy. The pattern-finding step of sPCA probably allowed to set the values of unsignificant voxels to zero, removing noise from those voxels.
2. The sinusoidal time course is still very noisy and affected by other perturbations in the ICA case whereas the same time course with the sPCA approach is very close to the initial one.

In this case, sPCA gives better results than ICA from a qualitative point of view. But the question is now to find a way to compare quantitatively the results. Several techniques are exposed in [8] and [17] such as the use of the Kullback-Leibler divergence to evaluate the independence of the resulting components, or the comparison between the true mixing matrix and the algorithms estimated mixing matrices. Since this toy example is quite simple, we choose to compare the resulting time courses. This technique is both computationally efficient and allows an easy comparison of two time courses. Concretely the steps to compare two time courses TC_a and TC_b are the following :

Algorithm 3 Compare TCa and TCb

```
$TC_{a,b} = TC_a - mean(TC_{a,b})$
$TC_{a,b} = \frac{TC_{a,b}}{norm(TC_{a,b})}$
$D^2 = (abs(TC_a) - abs(TC_b))^2$ // Due to the limitations of ICA we have to consider both $TC_a - TC_b$ and $TC_a + TC_b$, hence we use abs()
return $D$
```

Dr. P. Geurts proposed another measure of distance : $D' = \min(TC_a - TC_b, TC_a + TC_b)$. This measure is better than D because it allows to deal with more complex time courses such as a comparison between $sin(x)$ and $|sin(x)|$. However, in this case D and D' are equivalent and give the same results.

In the default configuration, we have the following results[7] to compare ICA and sPCA for varying values of the width of the simulated networks and after averaging over six simulations :

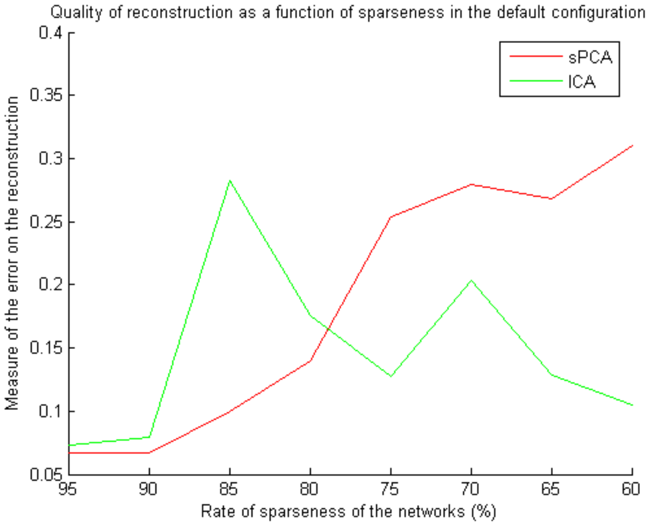

Figure 18: Measure of the quality of the reconstruction in the default configuration

For comparison, a sparseness of 90% is used in this section and that gives the results presented in figures 16 and 17.

We can see that the error on the reconstruction is a decreasing function of sparseness for the sPCA algorithm, which means that the quality of reconstruction is the best when the networks are the sparsest. On the other hand, in the case of ICA the reconstruction seems less linked to the sparseness of the networks. For high sparseness of the networks (greater than ≈ 80%) the sPCA approach gives better results whereas for networks with less sparsity ICA is better.

4.2 Sensitivity analysis

We are now studying the impact of the variation of some parameters on the quality of the reconstruction. The results are to be compared with the default configuration results.

[7] Note that the random time courses are not compared, and the other distances are computed as the distance to the nearest initial time course. There are thus in each case three distances considered to obtain the measure of the error on the reconstruction.

4.2.1 Presence of a perturbation

In the fMRI data acquisition, the data are often noised by motion of the patient. We tried here to simulate motion during the acquisition by adding a component that has the same amplitude on the whole map and which time course is composed by successive peaks. In this case we get :

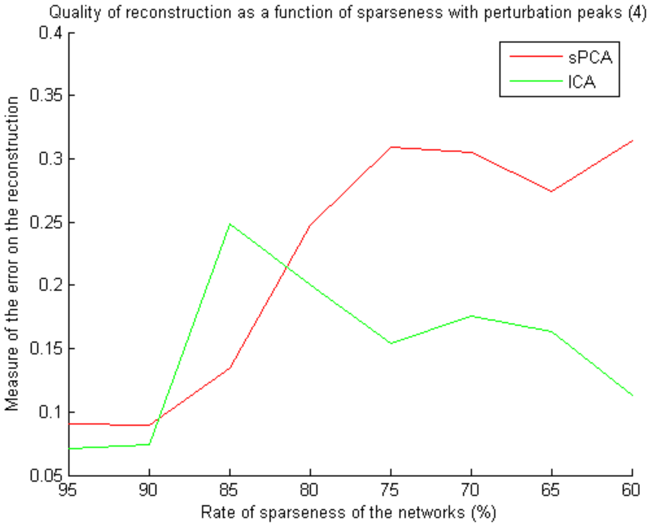

Figure 19: Measure of the quality of the reconstruction in the case of a perturbed signal

Even in the case of high sparseness of the networks, sPCA gives worse results than ICA which seems less affected than sPCA. We can see this by looking at the interval of sparseness of the networks for which sPCA gives better results than ICA in the default configuration and in this case. Here this interval is between 82% and 88% whereas it is between 78% and 95% in the default configuration. sPCA reconstruction seems thus less robust to the perturbation than ICA.

4.2.2 Overdecomposition

In the default configuration there are four networks and we are looking for four networks. In the case of real data we don't know the number of networks. We simulated here the case of overdecomposition, that is the algorithms seek for more components (five) than there are (four) :

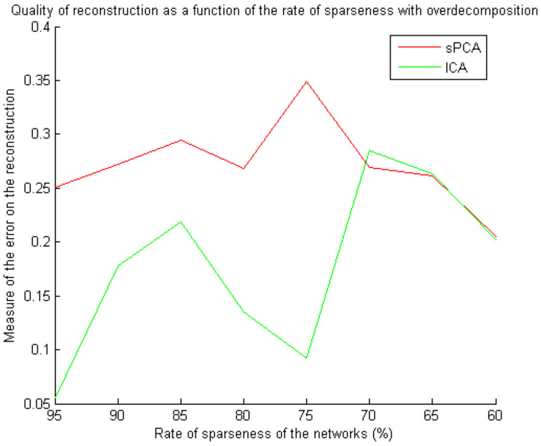

Figure 20: Measure of the quality of the reconstruction in the case of overdecomposition

In this case, sPCA seems affected even for networks with very high rates of sparseness whereas ICA keeps better results. This latter fact is probably due to fastICA automatically reducing the dimensionality of the data if the singular value corresponding to the component is too low (see section 2.1.2). The number of components extracted with ICA is thus equal to four, avoiding overdecomposition. We can also note that surprisingly the quality of the reconstruction increases when sparsity decreases in the case of sPCA.

4.2.3 Underdecomposition

The opposite case is when the algorithms are looking for less components (three) than there are (four) :

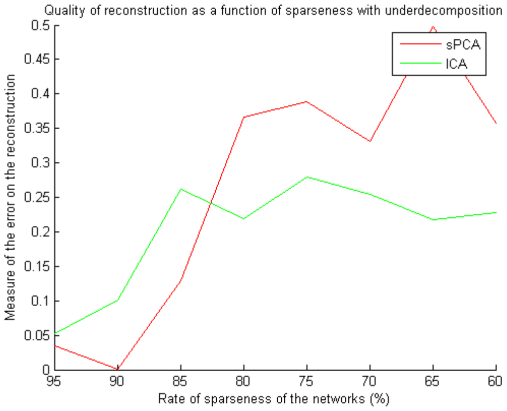

Figure 21: Measure of the quality of the reconstruction in the case of Underdecomposition

In the case of high sparseness of the networks sPCA deals well with underdecomposition. In fact it separates correctely two out of the four networks and mixes the last two networks in the last sparse principal components. In some cases (90% of sparseness), it even omits the less significant network (according to the amplitude) and keeps the three other networks. On the other hand ICA seems to mix more generally the components even at quite high levels of sparseness.

4.2.4 Simulation of a global signal

A part of the total BOLD signal is composed of fluctuations over the whole brain : the *global signal* [20]. This signal is characterized by low frequency fluctuations and a high mean and can be simulated by a function of the form :

$$G_S = A_1 + A_2 \sin(2\pi f t) \tag{55}$$

with $A_1 >> A_2$ and a small f.

In order to get rid of this signal we can apply two techniques :

1. Substract the mean over each line from the initial data [20].
2. [33] suggested that this could also be done by ignoring the first principal component obtained in the decomposition represented in figure 14 for the further processing.

Those two techniques were tested on our example :

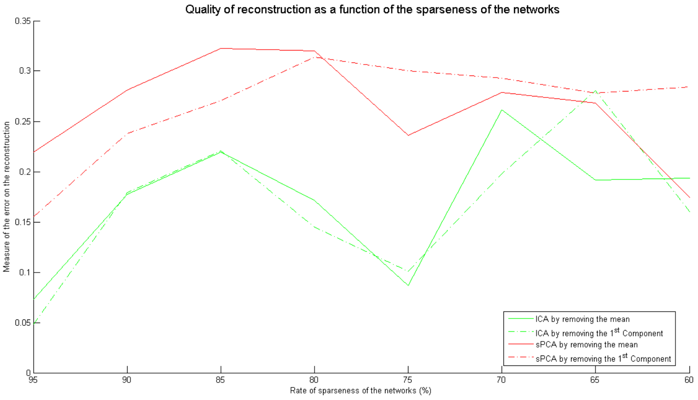

Figure 22: Comparison of two techniques used to remove a global signal

The results from ICA are once again significantly better than the ones given by sPCA. Moreover, substracting the mean seems more or less equivalent to removing the first principal component for both ICA and sPCA.

4.2.5 Amplitude of noise

We here show the influence of the amplitude of noise on the reconstruction processes. We considered the case of a sparseness giving similar results for ICA and sPCA : 95%.

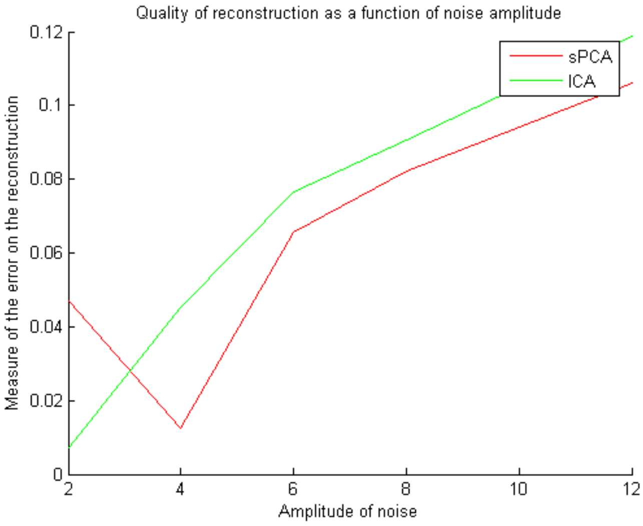

Figure 23: Measure of the quality of the reconstruction as a function of noise

Both algorithms deal with noise in the same way, the error increasing slightly with noise.

4.3 Isolating the perturbation

We have seen in section 3.4.1 that independence had to be achieved within the spatial components. In the case of sPCA, it seems natural to induce sparsity in the *spatial* components.

However, we have seen in section 4.2.1 that the presence of a perturbation deteriorates the decomposition of both sPCA and ICA techniques. The perturbation being composed of some peaks it is very sparse in *time*. The question is thus here to see whether sPCA could be used in order to induce sparsity in the time domain. Doing so could allow to isolate the perturbation and possibly to remove it from the initial data matrix.

We use the actual motion of a patient to simulate the perturbation that is introduced in the same way as presented in section 4.2.1 and we get :

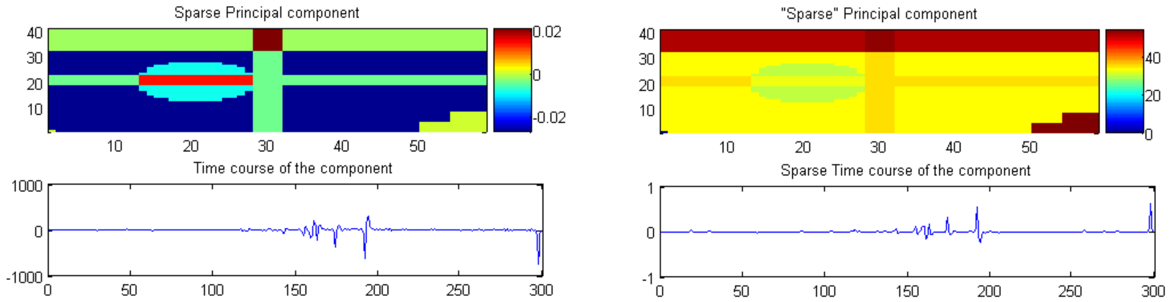

Figure 24: Spatial (left) and temporal (right) sparsity

The results are not very different. One can verify that there is slightly more sparsity in the time course represented on the right side of the picture. However, the spatial map associated to that time course still mixes with the other components and makes the removal of the motion component a difficult issue. This topic would need a specific attention and since it is not directly related to our master thesis we will let it as a future direction to explore.

4.4 Determining the number of components

As explained in section 2.1.1, it is possible to have an estimation of the number of components based on the values of the singular values. Let us just here show what we get in this ideal case :

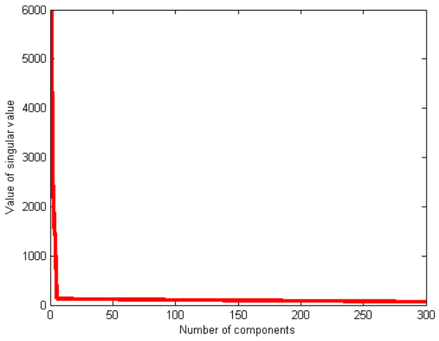

Figure 25: Evolution of the singular value as a function of the number of the component

We clearly see that in this case there are four networks to be extracted from the data. Unfortunately, we will see that with real data the conclusions are not so clear anymore.

4.5 Independence of the components

In this section we compute the *kurtosis* of the components as defined in (27) before and after treatment by the algorithms in the default case with a sparseness of the networks of 90% as shown on figures 13,15,16 or 17. After centering and normalizing all the components we get the following results :

Algorithm \Network	Line	Cross	Disk	Corner
True networks	3.11	7.12	46.56	139.88
PCA	3.04	7.08	35.87	112.94
PCA+ICA	3.02	7.08	44.53	134.85
PCA+sPCA	3.11	7.21	46.64	139.98

Table 1: Kurtosis of the components in several cases

We can see that the kurtosis of the components after PCA+sPCA is the highest. That comes from the fact that sPCA, more than finding the right networks, also eliminates noise allowing higher values in the network patterns after normalization. This results in an increase of the fourth power term in the definition of kurtosis (27). Noise also seems to be the reason why kurtosis is higher in the sPCA results than for the true networks.

ICA also gives good results, increasing the kurtosis of each component compared to PCA except for the line. This is probably due to the fact that fastICA uses another measure of independence than kurtosis.

5 Applications

We here first define the methodology adopted in each of the three following experiments. We then present those experimentations that aim to verify the results of the toy example analysis exposed in the previous section.

To make things clear we summarize in the following tab the patients and software that are used in the three experiments :

Experiment	Number of patients	Software used
Analysis of a "half brain" patient	1	Matlab 7.0 for the computational part and BrainVoyager Viewer 1.8 for the display of the results
Extracting the default mode	5 Controls and 2 LIS patients (and the 9 patients of the next experiment for figure 37)	Matlab 7.0 for the computational part and BrainVoyager Viewer 1.8 for the display of the results
From fMRI to PET	9	Matlab 7.0 and SPM5 for both the computation and the display of the results

Table 2: Patients and software in the three experiments

5.1 Methodology

We follow the classical scheme presented in figure 14 and the approach presented in section 3.1 but adapted to the dimensionality of real fMRI data. In this case, we have :

- $N_V = 40 \times 46 \times 58 = 106720$.
- $T = 298$ time samples at a sampling rate of 2 seconds, the highest detectable frequency is thus 0.25 Hz.
- The number of components is fixed at $N_C = 30$.

This last value is motivated by the singular value decomposition of the data matrix. The values of the singular values associated to the components are represented on the following graph :

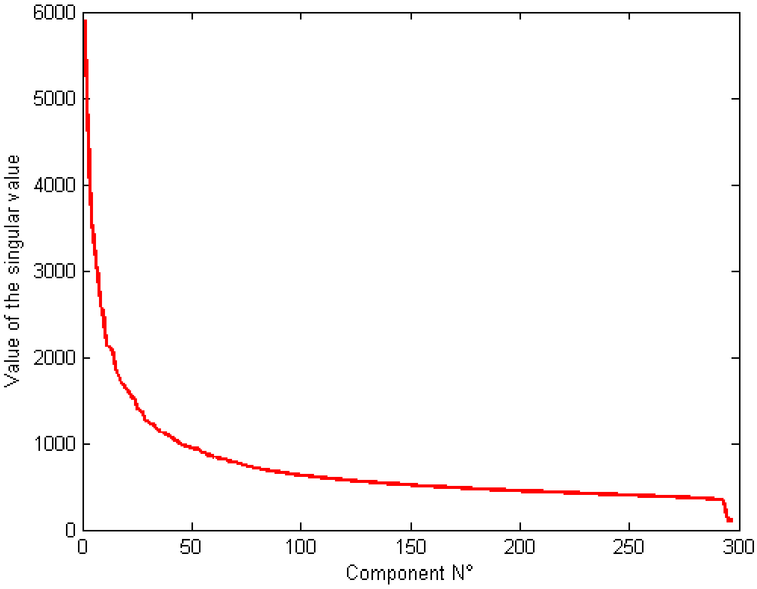

Figure 26: Values of the singular values of the components

The choice of $N_C = 30$ seems to be a reasonable tradeoff between computational efficiency and considering most of the variance in the retained data. This choice also gives good experimental results.

We mentioned the issue of removing the global signal in section 4.2.4 and presented two solutions. Let us present here why it is important to remove this signal properly. It is important to first introduce the *neuronal coefficient*.

As explained before, each extracted component is related to a time course and we saw in section 3.3.3 that the power spectrum analysis theoretically allows to say whether a component is due to neuronal activity. A typical power spectrum of a time course related to a principal component is given on the following picture :

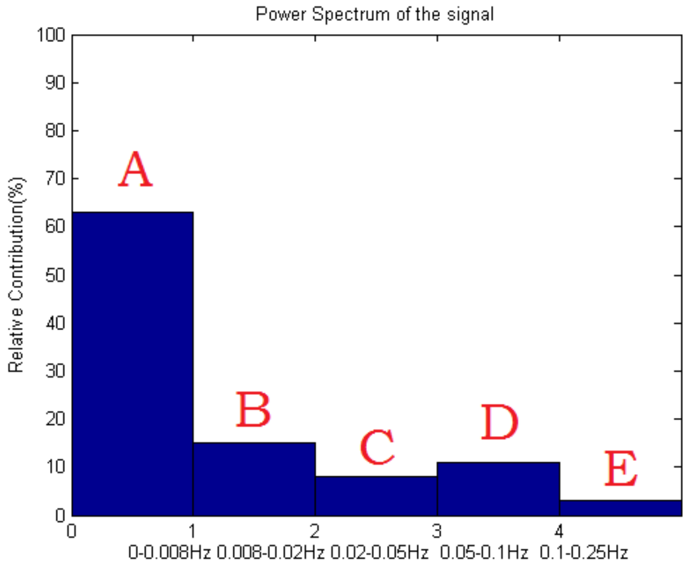

Figure 27: Typical power spectrum

The bars represent the contributions of the signal in each frequency band. The *neuronal coefficient*, that we denote η, is based on the values of A, B, C, D and E and Dr. A. Soddu proposed the following definition :

$$\eta = \frac{B + C}{A + D + E} \tag{56}$$

If η is greater than a certain value, the component can be considered to be neuronal. In fact, the different frequency bands can be associated to different phenomena :

- A is the low frequency band which, as explained in section 4.2.4 can be associated to the global signal.
- E is the high frequency band, typically corresponding to vascular noise.
- B, C and even D can be considered as the neuronal contributions even if B and D are at the boarders with A and E and thus also contain some contribution of the phenomena expressed in A and E. It seems that E is more affected than B which explains why B is at the numerator in equation (56) whereas D is at the denominator.

Let us now come back to the problem of removing the global signal. Here is the typical time course and power spectrum of a global signal :

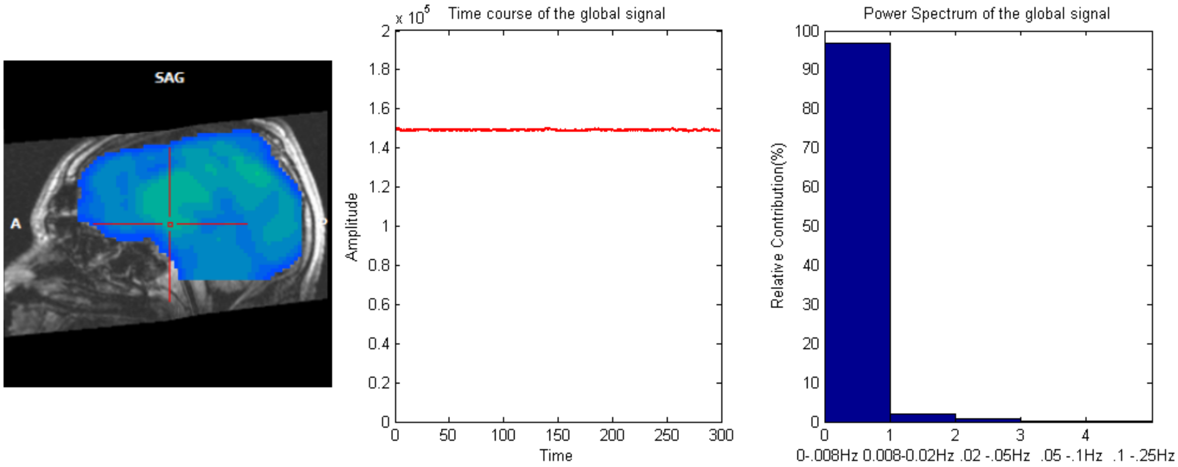

Figure 28: Typical global signal

The amplitude of the time course is huge and the signal is quite constant. Hence it is normal to get the power spectrum represented on the right part of the figure. Indeed if we suppose that the time course is constant : $x[n] = C$ and of length N, we get the following relation to express the power spectrum $\phi(k)$ of the time course :

$$\begin{aligned}
\phi(k) &= |\sum_{n=0}^{N-1} x[n] \exp(\frac{-2\pi i}{N}kn)|^2 \text{ for } 0 \leq k \leq N && (57)\\
&= \begin{cases} |\sum_{n=0}^{N-1} C|^2 & \text{if } k = 0 \\ C^2 |\sum_{n=0}^{N-1} \exp(\frac{-2\pi i}{N}kn)|^2 & \text{for } 1 \leq k \leq N \end{cases} && (58)\\
&= \begin{cases} N \cdot C^2 & \text{if } k = 0 \\ C^2 |\frac{1-\exp(\frac{-2\pi i}{N}kN)}{1-\exp(\frac{-2\pi i}{N}k)}|^2 = 0 & \text{for } 1 \leq k \leq N \end{cases} && (59)
\end{aligned}$$

The only non-zero value of the power spectrum is in $k = 0$ and that non-zero value is high ($N \approx 300$ and $C \approx 100000$). Hence the power spectrum has a very high contribution in the low frequency band. This contribution is so high that it perturbes the analysis of the time courses of all the components obtained with sPCA or ICA because the first band contribution is very high and the other contributions are nearly irrelevant. We give for example here the power spectrum of all the components in the case of a PCA followed by a sPCA treatment :

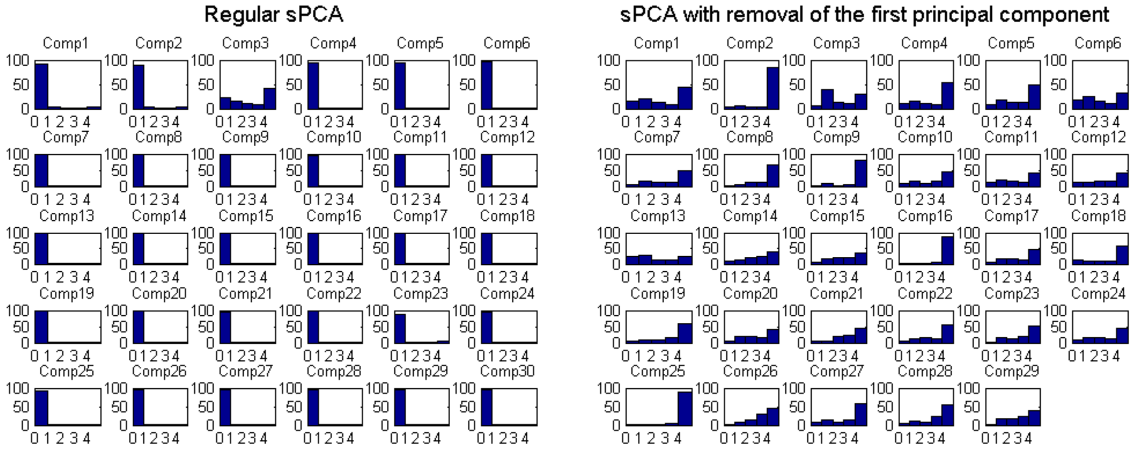

Figure 29: Impact of the removal of the global signal

We can see on the right part of this figure that nearly all the components are affected by the presence of the global signal and that the removal of this signal allows to highlight the other features of the components, making the computation of η more relevant. Appendix A.1 provides the time courses corresponding to those power spectra.

We mentioned in section 4.2.4 two ways to remove the global signal : removing the mean from the initial fMRI data or removing the first principal component once PCA has been performed [33]. Those methods experimentally showed to be more or less equivalent in the case of our model. However, experimentally the latter seems to better favour the expression of the neuronal frequency bands. We thus choose to get rid of the first principal component in order to remove the global signal. Concretely, here are the steps that we follow in order to derive the proper ICA and sPCA components and related time courses :

1. Removal of the first principal component (drop the first line of the principal components matrix).
2. Application of ICA and sPCA on that modified matrix.
3. As explained in section 3.3.2 the time courses are obtained by multiplying the initial data matrix with the inverse of the components matrix. In order to compute the correct time courses related to the ICA or sPCA components we have to remove

the contribution of the first principal component in the initial fMRI data. This contribution is equal to the product between the first principal component and its related time course. The computation of the modified total signal is mathematically expressed by :

$$S^*_{T,N_V} = S_{T,N_V} - TC^{PC_1}_{T,1} * C^1_{1,N_V} \tag{60}$$

where C^1 and TC^{PC_1} are the first principal component and its related time course, respectively.

4. Computation of the time courses matrix related to the ICA and sPCA components :

$$TC_{T,N_C-1} = S^*_{T,N_V} * C^{-1}_{N_C-1,N_V} \tag{61}$$

5.2 Analysis of a "half brain" patient

The only tool we have up to now to determine if a component is neuronal or not is the analysis of the power spectrum of the time course related to that component. However, we had the opportunity to get the data coming from a patient whose brain is active only in the right hemisphere. In that case we have another tool to determine whether a component is neuronal or not. Indeed, the neuronal components are those that show activity only in the right hemisphere. If a component shows activity all over the brain for example, that component is certainly related to a non-neuronal source. Here is an example of a neuronal component for this patient :

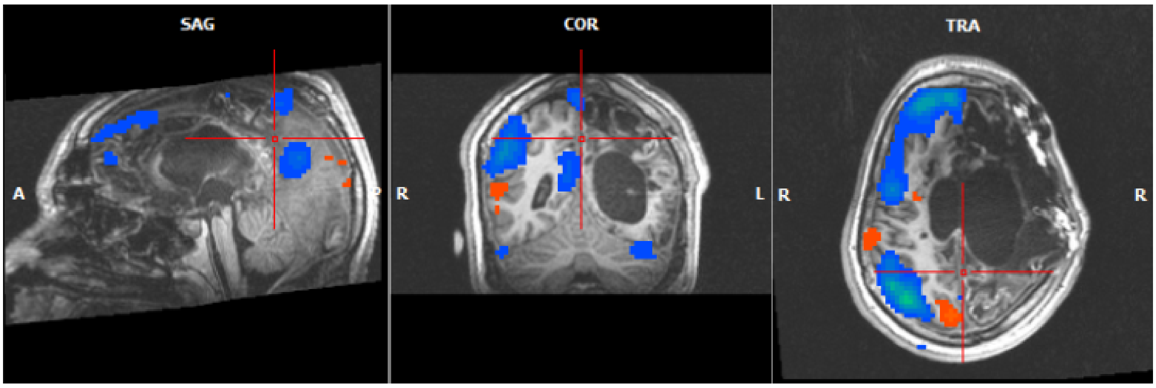

Figure 30: Neuronal component of the "half brain" patient

In this experiment we follow the scheme represented in figure 14 and the steps given in the previous methodology section to extract the components with ICA and sPCA. The idea is to underline some properties of the power spectrum of the time courses related to neuronal activity and in particular to check the validity of the neuronal coefficient η.

In order to quantify asymmetry of the components, we introduce the *asymmetric coefficient* χ defined by:

$$\chi = \frac{R - L}{R + L} \tag{62}$$

where R (L) denotes the sum of the absolute values of the voxels located in the right (left) hemisphere of the brain. Based on those two coefficients η and χ we can place each component on a map for ICA and for sPCA :

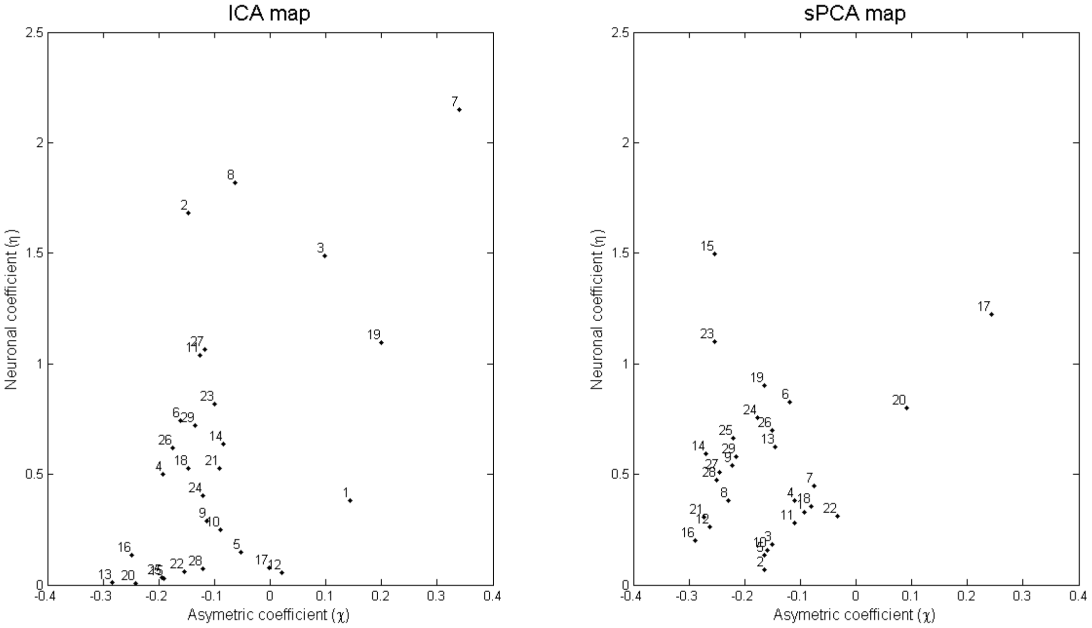

Figure 31: Neuronal maps of the components with ICA and sPCA

Those results are not very concluding. Indeed, we expected the neuronal components according to one criteria to be neuronal according to the other. Here it is obviously not the case : the components with the largest asymmetric coefficient χ does not have the largest neuronal coefficients η (see for example component 1 or 19 in the ICA map or 20 in the sPCA map).

By visual inspection of the power spectra of the components and the components themselves we had the idea to introduce a new version of the neuronal coefficient defined by :

$$\eta' = \frac{C}{A + B + D + E} \tag{63}$$

If we use this variant of the neuronal coefficient we get the following maps :

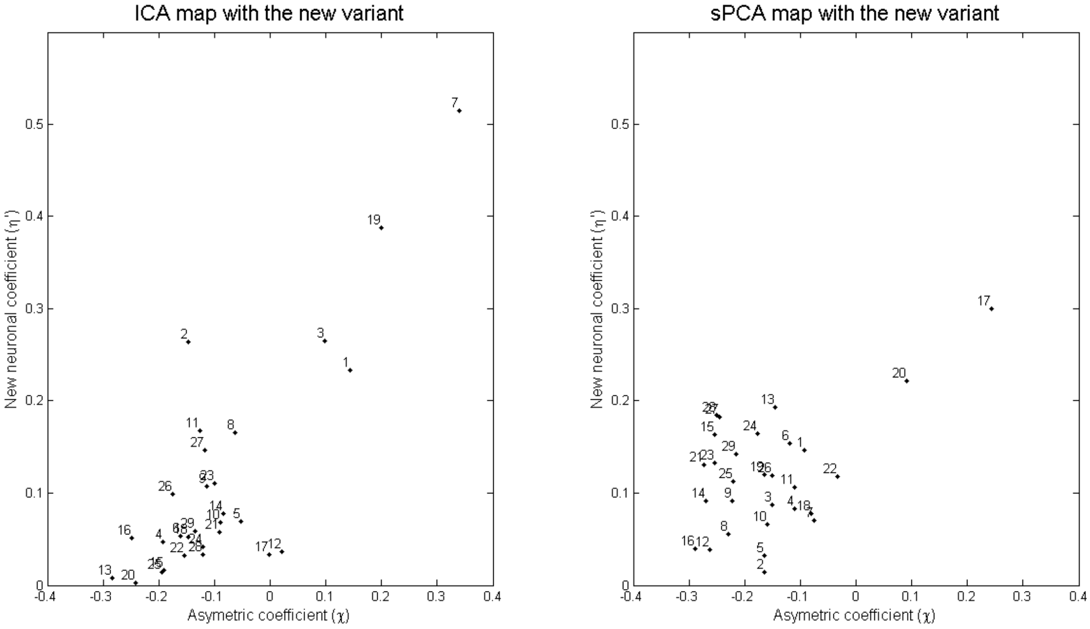

Figure 32: Neuronal maps of the components with ICA and sPCA and the new variant of the neuronal coefficient

The dispatching of the components on these maps are more satisfactory. The components with the highest neuronal coefficients are effectively the ones with the highest asymetric coefficients and η' thus seems to be a more reliable measure of the "neuronality" of a component. In this case components 7 and 19 with ICA and 17 and 20 with sPCA are neuronal. If we look only to η' which is the only criteria we can rely on to measure "neuronality" of the components in the case of regular brains, ICA seems to draw a better separation between neuronal and non-neuronal components. In particular, we can see this at the range of values taken by the neuronal coefficient. This range is significantly wider in the case of ICA than sPCA.

However the trap in introducing the new neuronal coefficient was to overfit the data in order to have a proper separation of the maps presented in figure 32. In fact when looking for a new version of η we imposed ourselves to find a very easy new formulation and the one presented in equation (63) appeared naturally. For example we did not try to change the bounds of the frequency bands which probably could have improved the separation between the neuronal and non-neuronal components.

In conclusion this new formulation of η gives good results in this case but we will have to test this new formulation in other configurations to confirm this.

5.3 Extracting the default mode

As in the toy example presented in the previous section, we will here try to extract a neuronal network from fMRI data. Following the scheme presented in figure 14 we want to extract the default mode (see figure 10) from some patients and evaluate the quality of the extracted network. The data used in this experiment consist of five control patients and two Locked-In Syndrom (LIS) patients' fMRI. The LIS patients moved a lot but still present a default mode and the interest is thus to see how both techniques deal with the motion of those patients.

The difficulty here is that we do not know what the real default mode *exactly* looks like even if this network was highlighted with the use of ICA in several studies (see [39] or [30]). We do not want here to compare the results obtained with sPCA to the ones obtained with ICA because we do not know a priori which decomposition technique is the best.

First, both techniques are used on the control data in order to build an average template of the default mode by averaging the five default modes that have been extracted. Afterwards, the default modes are extracted from the LIS-patients data and we compare those default modes to the average templates as shown here :

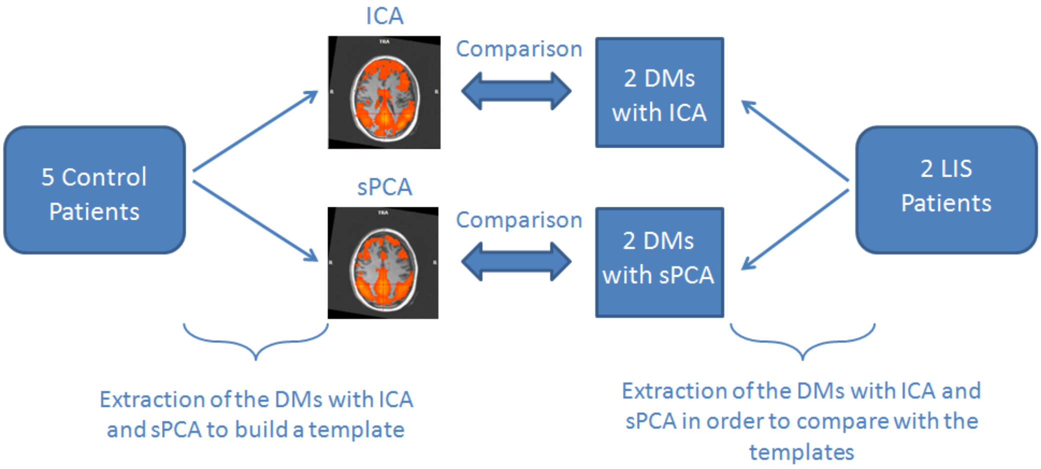
ICA
Comparison
2 DMs with ICA
5 Control Patients
2 LIS Patients
sPCA
Comparison
2 DMs with sPCA
Extraction of the DMs with ICA and sPCA to build a template
Extraction of the DMs with ICA and sPCA in order to compare with the templates

Figure 33: Scheme of the experiment

We have seen in section 2.2 that the networks extracted by ICA could be multiplied by -1. Then, we take the absolute values of the values in each voxel to derive the average. This is the reason why there is only one color on the resulting templates.

Taking the average[8] euclidean distance between the default mode templates and the LIS-patients default modes we have :

Algorithm \Patient	MJ-LIS	DD-LIS
sPCA	282	273
ICA	244	203

Table 3: Distances between the DM templates and the LIS-patients' DMs

The results are thus significantly better in the case of ICA than in the case of sPCA. Since the LIS patients moved a lot, we could explain this result at the light of figure 19 where we showed that sPCA seemed less robust than ICA to motion perturbations.

Using the average default mode just computed, we build a map similar to the ones presented on figures 31 and 32 with the neuronal coefficient η on the y-axis but on the x-axis instead of the assymetric coefficient χ we defined δ as the euclidean distance between a component and the average default mode template presented at the center of figure 33 for both ICA and sPCA. Plotting the 29 components of a control subject on such a map, we

[8] Average over different runs of the decomposition.

expect to find the neuronal components high on the graph and the default mode should be both high on the map (high η) and on the left of the map (low δ). Here is what we get for both ICA and sPCA in the case of a representative control :

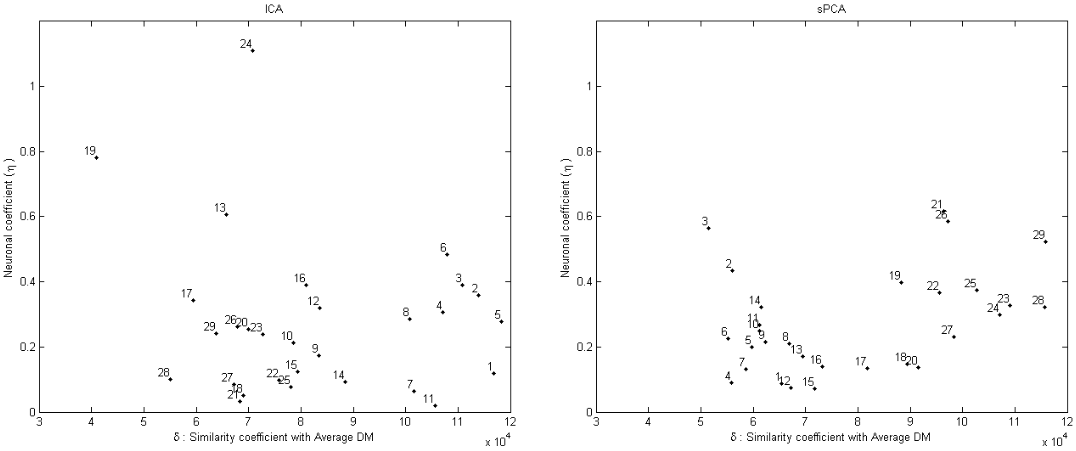

Figure 34: Dispatching map according to δ and η for ICA (left) and sPCA (right)

We can see that the separation along the horizontal axis is much better in the case of ICA. That means that the default mode is extracted in one component with ICA whereas in sPCA several components can be labelled as default mode according to δ. This seems to show that the spatial decomposition of ICA is better than the decomposition performed by sPCA. We can compare this result to the one presented on figure 20 in section 4.2.2 that concludes that sPCA deals worse with overdecomposition than ICA does. Moreover, the absolute value of η for the components supposed to be neuronal is much higher for ICA (at least for components 19 and 24), meaning that ICA is more able to extract neuronal information.

It could also be interesting to look at the power spectra (PS) of the time courses related to the default modes. As we explained in section 3.3.3 the neuronal components have higher contributions in the frequency band going from ≈ 0.01 Hz to ≈ 0.1 Hz. We compute the average power spectrum of the default modes extracted from the controls and we compare it to those of the LIS-patients and with the average power spectrum of the controls over all the components for ICA :

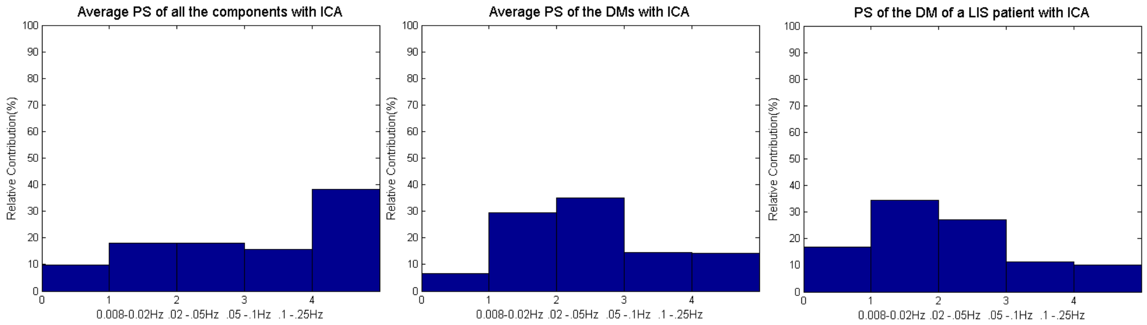

Figure 35: Comparison of the power spectra for ICA : average power spectrum (PS) over all the components (left), average PS of the default mode (center) and PS of the default mode of a LIS patient (right)

And for sPCA :

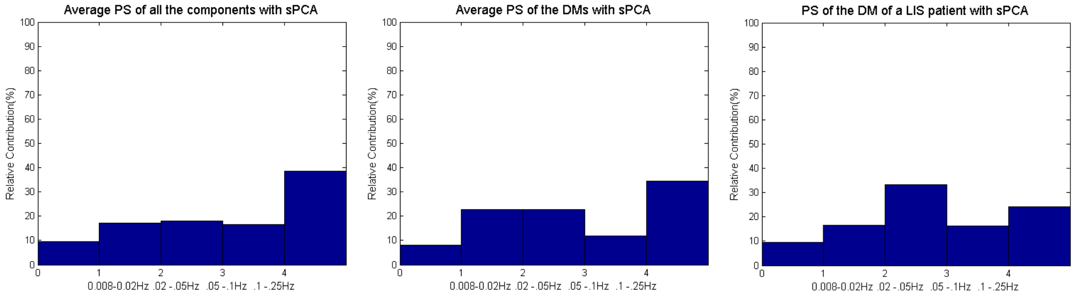

Figure 36: Comparison of the power spectra for sPCA : average power spectrum (PS) over all the components (left), average PS of the default mode (center) and PS of the default mode of a LIS patient (right)

First, we can see that the average PS over all the components is equivalent in both cases. Moreover, for both ICA and sPCA the second and third bars in the graphs, that account for neuronal activity, have higher contributions than in the average power spectrum. However, the average PS of the default modes show different features. In the case of sPCA, the highest frequency band corresponding to vascular noise is still present. As we explained in section 3.3.3, this band does not contain neuronal information and this is clearly a bad point for sPCA. That means that sPCA seems to be less able than ICA to extract neuronal information out of the components.

In this experiment we use five control patients. We will now use the nine subjects of the next experiment to compute more reliable statistical quantities about the default modes. As explained in section 5.4 below, the data of that section have not exactly the same features than in this experiment and that is the reason why we did not use those results in a comparison with the LIS patients of this experiment. However, the comparison of the ICA vs sPCA extraction of the default modes is relevant in this section. Here is thus the average and standard deviation of the default modes extracted by ICA and sPCA for the nine subjects used in the next experiment :

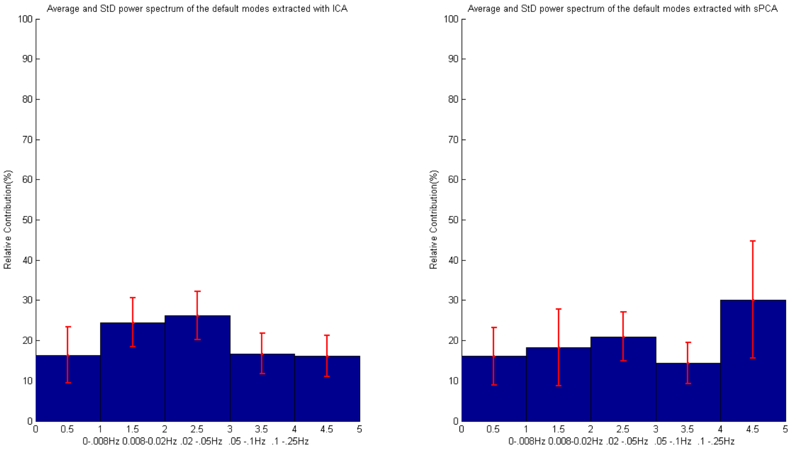

Figure 37: Comparison of the extraction of the DMs with ICA and sPCA

The observations concerning the means are the same as the ones done about figures 35 and 36. Moreover, the standard deviations of the contributions in each frequency band are significantly greater in the case of sPCA, showing that sPCA seems more unstable in extracting the default mode. This seems to confirm constancy of the extracted networks with ICA, which precisely was a reason of its success [38].

In conclusion we can note two trends outlined with this experiment :

1. ICA seems more robust to motion and environment changes than sPCA. In the case of motion and overdecomposition this point is supported by the simulation of a perturbation in the toy example.
2. Considering the contributions of each frequency band in all the power spectrum graphs suggests that ICA is more able to extract neuronal information whereas the neuronal components extracted by sPCA also include high frequency noise.

However we can have some reservations about the small number of patients used in this experiment that does not allow to verify the underlined trends.

5.4 From fMRI to PET

PET or *positron emission tomography* is a 3D nuclear medicine imaging technique that allows to study the functional processes in a specific region. When the brain is the organ that is studied, this method allows to localize the precise regions of the brain where glucose is consumed and thus reflects metabolic activity in a reliable way, more reliable than fMRI. However, the delay needed to get a whole PET map is quite important and does not allow to have the temporal precision requested to perform studies such as a power spectrum analysis of a time course.

This experiment requested to acquaint us with the *SPM* (Statistical Parametric Mapping) software. This is a powerful tool to analyze fMRI or PET maps but we had to transpose our codes in this new environment.

In this section we select the *neuronal* components in the classical decomposition performed either by ICA or sPCA. Then, starting from those components we build up a map, the *model map*, supposed to reflect the neuronal activity of the patient. We finally compare it to the PET map of the patient as explained on the following scheme :

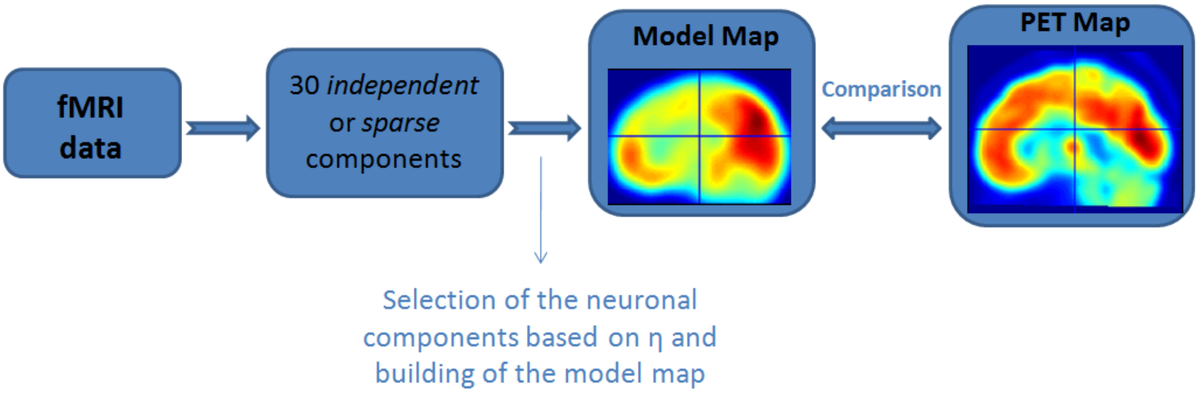
fMRI data
30 independent or sparse components
Model Map
Comparison
PET Map
Selection of the neuronal components based on η and building of the model map

Figure 38: Methodology to compare the model map and the PET map

The steps to build the model map are deliberately not detailed because it will be soon the object of a publication by Dr. A. Soddu. However, we can discuss some parameters of the building of the model map. We selected the neuronal components based on the analyses of their power spectra and using the neuronal coefficient η defined in equation (56).

If we sort the components according to η, the first one being the one with the highest neuronal coefficient, it could be interesting to see how good the model is correlated to the

PET map when the number of selected components varies. The following picture shows the link between number of components and quality of the model map :

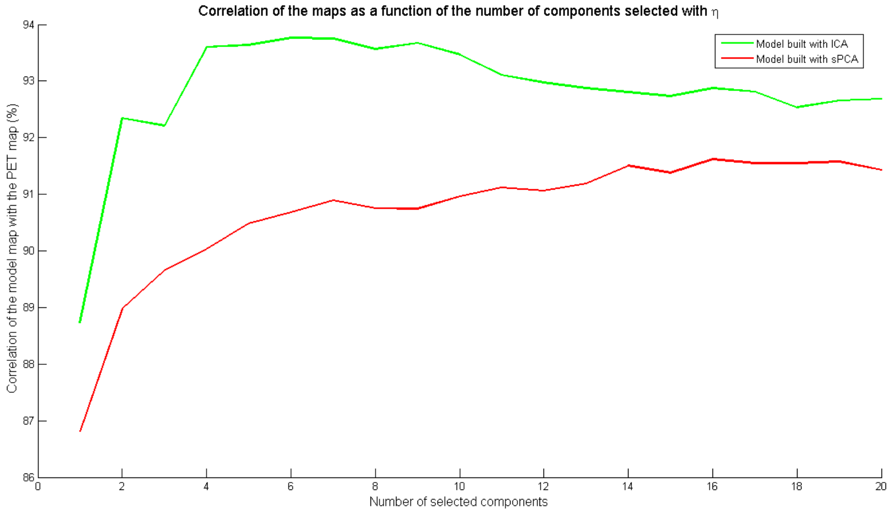

Figure 39: Correlation between the model map and the PET map for ICA and sPCA, as a function of the number of selected components

This graph gives a lot of information about the content of the different components. Considering the curve obtained using ICA, the fact that there is a maximum at around six[9] components is important. That means that up to that component, adding a component improves the model map and we can thus suppose that the neuronal information is concentrated in the first components, as expected. On the contrary the sPCA curve is a monotonous increasing function. That seems to show that there is still neuronal information in the further components and that the sPCA approach does not allow to isolate the neuronal information wheras ICA does. Appendix A.2 provides the same curve for two other subjects, leading to quite the same conclusions.

Here is a sagital view of the model map obtained using ICA compared to the PET map :

[9] This number varies between ≈ 6 and ≈ 12 for most of the nine subjects, see Appendix A.2.

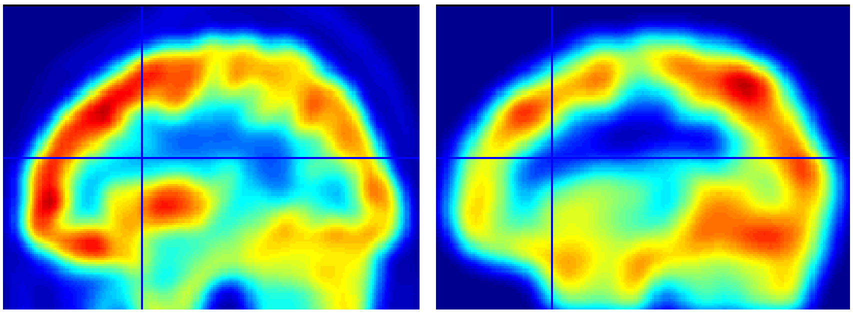

Figure 40: Comparison between the PET map (left) and the Model map (right)

The result is quite satisfactory and in this case we have a correlation[10] of 94%.

Let us now consider the other variant of the neuronal coefficient, η', to order the components. If we build the model map based on this ranking, we get the following results for the relation between the number of components used to build the model map and the quality of that map:

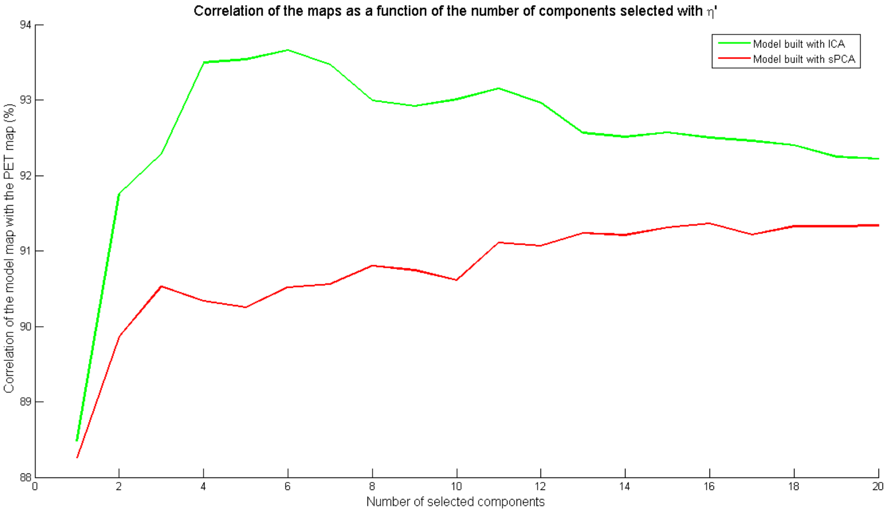

Figure 41: Correlation between the model map and the PET map for ICA and sPCA using η', as a function of the number of selected components

The global shape of the curve is the same as the one presented in figure 39 but one

[10] The correlation is computed as defined for the default configuration of the function *corr()* in Matlab, that is with the Pearson's correlation defined by $\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$ where σ denotes the standard variation of the random variables X and Y.

detail must retain our attention. The third component for the ICA construction and with the use of η is clearly not neuronal since the quality of the contruction decreases if we consider that component to build the model (see figure 39, upper curve). On the other hand the third component selected using η' seems to be neuronal since in that case the function increases until the sixth component and than decreases. Hence the separation between neuronal and non-neuronal components seems to be better when using η' rather than η. However this difference does not result in a difference of the maximum correlation using η or η', which is quite surprising.

This experiment tends once again to show that ICA is better than sPCA in extracting the neuronal information out of the total data. The argumentation holds on two observations :

- The curves presented on figures 39 and 41 both present a maximum in the case of ICA showing that the first components ordered according to η or η' account for neuronal activity and the last ones does not. The curves obtained with sPCA on the contrary are always increasing when adding components, meaning that each component adds neuronal information to the model and that the separation between neuronal and non-neuronal components does not allow to isolate neuronal information.
- Not very surprisingly with respect to the first point, the absolute value of the quality of the model map (or correlation with the PET map) is better for the ICA curves and for every number of components.

Here is finally a table representing the correlation between the model map and the PET map with the optimal number of components for ICA and the same number for sPCA, for eight other subjects and using η. This result confirms the last point of the argumentation that we just developed :

Algorithm \Subject	1	2	3	4	5	6	7	8
	Correlation between the maps (%)							
sPCA	82.1	90.4	91.6	92.0	93.0	91.8	91.4	91.8
ICA	84.5	91.8	93.4	93.7	93.6	92.9	92.2	93.9

Table 4: Maximum correlation between the maps for ICA and sPCA

6 Future Directions and questions

In the previous sections we presented simulations and experiments that allowed us to address the very topic of this master thesis : what are the fundamental features of sPCA in general and as a tool to analyze fMRI data. We also presented several results allowing to compare sPCA to ICA. This whole work raised questions and sometimes led to conceive new trails that could be interesting to explore.

In this section we develop some of those topics and possible improvements that we believe could be further studied.

6.1 Increasing the number of subjects

In our experiments we use from one to nine subjects. This allows us to give a good idea of the qualitative and quantitative behaviour of sPCA and ICA. However, increasing the number of subjects could improve the quality of some statistical results given in the application section. For example, the mean default mode represented in figures 35 and 36 is obtained by averaging over five default modes, making this template quite uncertain (high variance). In particular, we were not able to classify correctly the default modes among the components using K-means, a basic classification algorithm.

More generally, increasing the number of subjects might allow to use machine learning as a tool to select the neuronal components.

6.2 Improving the model

The model presented in section 4 can of course be improved. This could be done for example through simulating more realistic time courses rather than sawtooth or sinusoidal time courses. Realistic time courses would allow to analyze their power spectra, which was not relevant in our model.

However, it seems that realistic time courses (hence more complex) should be analyzed and compared with another technique than the one used in this master thesis (euclidean distance between the true and the extracted time course). Some authors presented other tools (see [8] or [17]) that might allow a richer interpretation of the results.

6.3 Distribution of the values of η

The results shown on figures 31 and 32 show a certain distribution of the values of η for each component, the highest values of η resulting in a high probability for the component to be neuronal. Using the nine subjects of the experiment presented in section 5.4, we compute the distribution of the values of η for either ICA or sPCA among all the subjects and components. Our expectation is here to be able to build a model, such as a *gaussian mixture model* for example (see [19]) of the distribution of the values of η for both the neuronal and the non-neuronal components. Here is the distribution we get for ICA[11] and η :

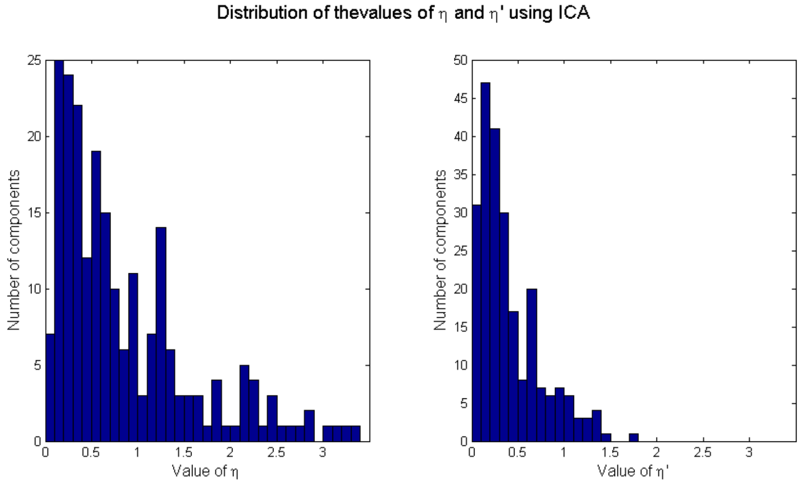

Figure 42: Distribution of the values of η and η'

Let us remark that the y-axes do not have the same scales because since the interval of variation of η is larger than the one of η' the interval of constant width in both graphs are proportionnaly wider in the case of η' and thus contain more elements. Besides, there does not seem to be a clear separation in any of the graphs.

If we suppose that it could be relevant to classify the components in two categories based on the value of the neuronal coefficient we can try to model those distributions by the mixture of two gaussians using an *expectation maximization* algorithm ([19] and Appendix B for the Matlab code). Here are the models in both cases :

[11] We choose ICA because as explained before the range of values of η is wider in the case of ICA compared to sPCA.

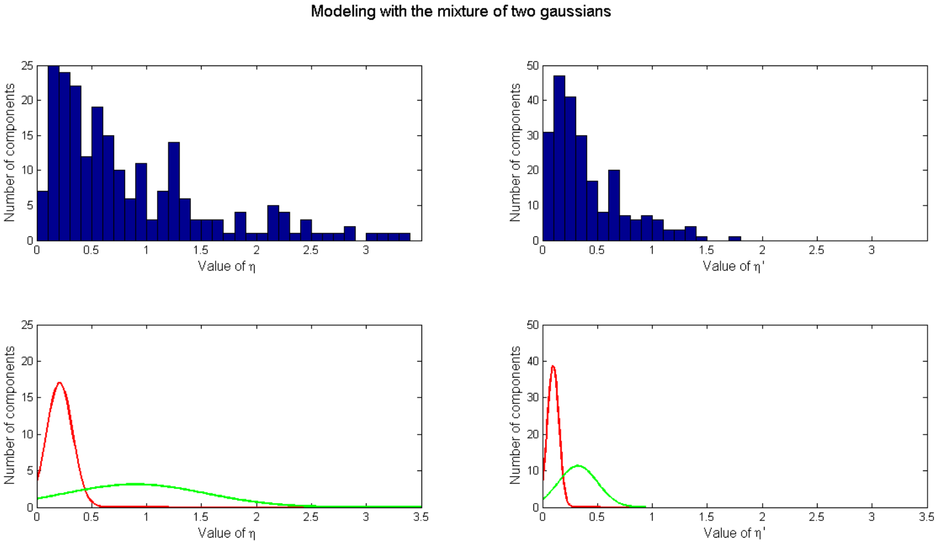

Figure 43: Modeling the distributions of the values of η and η' with two gaussians

Up to a scaling factor the results are quite the same. If we suppose that the gaussians correctly model the distributions of η for neuronal and non-neuronal components, we can remark a substantial overlapping between the gaussians that could explain why it is difficult to discern the neuronal component from the other based on η (or η'). Let us finally note that the algorithm used to compute the model did not find any result when looking for a model with three or four gaussians.

We can retain three questions and remarks out of those first elements :

- Is it reasonable to consider that η allows to split the data into only two categories? Indeed, even if η was built to highlight the "neuronality" of the components, this does not mean that *in fact* η does not separate the components according to some other criteria.
- Why did the algorithm [19] not find a solution for three or four gaussians? Is there a statistical property of the data that could explain this result?
- It could be interesting to perform the same kind of analysis with more subjects in order to have more reliable results.

In conclusion, if we take some precautions this approach could lead to a better understanding of the separation between neuronal and non-neuronal components performed by a specific decomposition technique.

6.4 Improving sPCA ?

One of the main drawbacks of sPCA compared to ICA seems to be its non-robustness to some perturbations. A first step to address this problem could be to characterize more precisely what kind of perturbation is the cause of this non-robustness. If it is possible to isolate some features that cause sPCA to be non robust, it might be possible to take this feature into account in the formulation of the sPCA optimization problem by modifying (41).

Moreover, we can expect a good choice of the sparsity parameter γ to improve the results of the sPCA algorithm. Our choice was to set values of the sparsity parameters lower than γ_{max} (see comments of equation (42)) in order to obtain non-zero solutions but we did not further tune the parameters. This choice was motivated by the fact that fastICA is also tunable through the function G for example and we did not want to promote a decomposition technique over the other. It could thus be interesting to compare the results of optimally tuned versions of both ICA and sPCA algorithms.

Another way to address the problem of non-robustness to overdecomposition of sPCA highlighted in figure 34 could also be to decrease the number of components selected in the dimensionality reduction step.

7 Conclusions

The conclusions of this master thesis are multiple. From a personal point of view I discovered a new way of addressing problems, of learning, of searching. The fact that we have a lot of time to work on our master thesis allows to deepen some topics and to have a better understanding of the underlying theories. This is important and allows us to make our own vision of the problems, to build the problems from their beginning instead of just trying to solve them.

After giving some ideas of future trails that could be explored in the previous section we now present the concrete results of the master thesis.

In the theoretical part we give a global view of component analysis, drawing parallels between the different techniques whereas the link between those concepts is often unclear because done too fast, without satisfying explanation. One key to do so was to adopt a constant and clear nomenclature in each case. Moreover, the exploratory theoretical section 2.2.5 gives a geometrical link between the statistical properties of the independent components and the structure of the fixed point iteration, perhaps allowing a better understanding of fastICA.

In the second part of this master thesis, our model of fMRI data presented in section 4 can be used to verify experimental results. In particular, the results presented in figure 19 seem to confirm the ones of section 5.3 : sPCA seems less robust to motion perturbations than ICA and the results of the simulation exposed in section 4.2.2 also seem to confirm the higher sensibility of sPCA to overdecomposition presented in the comments of figure 34. The first result of the simulations, with the default parameters (figure 18) confirms the expectations expressed in [17] for example : sPCA gives better results when the networks are very sparse. Hence, sPCA could give good results in the extraction of neuronal network in some ideal configurations (without perturbation). However, most of the results of the simulations including some kind of perturbation (motion, overdecomposition, presence of a global signal) tend to show that ICA is globally more robust than sPCA to those perturbations (figures 19, 20, and 22).

In the experiments of the last section, the formulation of a new neuronal coefficient in equation (63) seems to give better results and to be more relevant. Two elements support this argument : the separation of the components of the "half brain" patient presented in figure 32 using η' is clearly better than the separation using η presented at figure 31 and in the building of the model map of section 5.4, all the first components increase the quality of the model map when sorted with η' (figure 41) whereas this is not the case with η (figure 39).

In addition to the global non-rubustness of sPCA compared to ICA already mentioned, the other main result of this last section is that sPCA seems to be less able than ICA to isolate neuronal information. We can mention three results of the experimental section as arguments. First, when extracting the default modes in section 5.3, we saw that the average power spectrum of the default modes extracted with sPCA contains a much higher vascular noise contribution (frequency band E) than in the case of ICA (figures 35 and 36). The second result is obtained in the experiment with the "half brain" patient data (section 5.2) : on the figures 31 and 32 representing the dispatching of the components either using η or η' we can see that the level of threshold value that separates neuronal from non-neuronal components is much higher in the case of ICA than in the case of sPCA. That means that the neuronal components extracted with ICA contain more neuronal information, are more "neuronal" than the ones extracted with sPCA. The last result is the quality of the model map built using ICA or sPCA : as explained in section 5.4 in the comments of figures 39 and 41, the fact that for each sPCA component the add of this component improves the quality of the model map seems to show that there is neuronal information in most of the components, hence neuronal information is not gathered in few components as in the case of ICA.

As a final conclusion, we can say that even if conceptually sparsity is an interesting feature to incorporate to the decomposition techniques, sPCA seems to be non-robust to the perturbations inherent to experimental fMRI data and less able to extract and gather neuronal information in a few components, compared to ICA.

References

[1] G.K. Aguirre, E. Zarahn, and M. D'Esposito. The variability of human, bold hemodynamic responses. *Neuroimage*, 8:360–369, 1998.

[2] A.H. Andersen, D.M. Gash, and M.J. Avison. Principal component analysis of the dynamic response measured by fmri : a generalized linear systems framework. *Magnetic Resonance Imaging*, 17(6):795–815.

[3] J. Anemüller, J. Duann, T.J. Sejnowski, and S. Makeig. Spatio-temporal dynamics in fmri recordings revealed with complex independent component analysis. *Neurocomputing*, 69(13-15):1502–1512, August 2006.

[4] J. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7:1129–1159, November 1995.

[5] B. Biswal, F.Z. Yetkin, V.M. Haughton, and J.S. Hyde. Functional connectivity in the motor cortex of resting human brain using echo-planar mri. *Magnetic Resonance in Medicine*, 34(4):537–541, 1995.

[6] B.B. Biswal, J. Van Kylen, and J.S. Hyde. Simultaneous assessment of flow and bold signals in resting-state functional connectivity maps. *NMR IN BIOMEDICINE*, 10:165–170, 1997.

[7] J.-P. Cabannes. *L'Analyse en Composantes Principales*. Institut d'études politiques de Paris, Page destinée aux étudiants d'économétrie des mastères d'économie : "http://www.cabannes.net/acp.pdf", July 2009.

[8] V.D. Calhoun, T. Adali, and G.D. Pearlson. Independent component analysis applied to fmri data : a generative model for validating results. *Journal of VLSI Signal Processing*, 37:281–291, 2004.

[9] V.D. Calhoun, T. Adali, G.D. Pearlson, and J.J. Pekar. A method for making group inferences from functional mri data using independent component analysis. *Human Brain Mapping*, 14(3):140–151, 2001.

[10] V.D. Calhoun, T. Adali, G.D. Pearlson, and J.J. Pekar. Spatial and temporal independent component analysis of functional mri data containing a pair of task-related waveforms. *Human Brain Mapping*, 13:13–43, 2001.

[11] V.D. Calhoun and T.Adali. Unmixing fmri with independent component analysis. *Engineering in Medicine and Biology Magazine, IEEE*, 25(2):79–90, April 2006.

[12] V.C. Chen. Evaluation of bayes, ica, pca and svm methods for classification. *RTO SET Symposium on Target Identification and Recognition Using RF Systems*, 2004.

[13] D. Cordes, V. M. Haughtou, K. Arfanakis, J. D. Carew, and K. Maravilla. Hierarchical clustering to measure connectivity in fmri resting-state data. *Magnetic Resonance Imaging*, 20(4):305–317, 2002.

[14] D. Cordes and R.R. Nandy. Estimation of the intrinsic dimensionality of fmri data. *Neuroimage*, 29:145–154, 2006.

[15] D.A.Gusnard and M.E. Raichle. Searching for a baseline : functional imaging and the resting human brain. *Nature reviews neuroscience*, 2(10):685–694, 2001.

[16] A. D'Aspremont, L. El Ghaoui, M. Jordan, and G. Lanckriet. A direct formulation of sparse PCA using semidefinite programming. *SIAM Review*, 49(3), 2007.

[17] I. Daubechies, E. Roussos, S. Takerkarta, M. Benharrosh, C. Golden, K. DArdenne, W. Richter, J.D. Cohen, and J. Haxby. Independent component analysis for brain fmri does not select for independence. *Proceedings of the National Academy of Sciences*, 106(26):10415–10422, June 2009.

[18] K. Delac, M. Grgic, and S. Grgic. A comparative study of pca, ica and lda.

[19] I.D. Dinov. Expectation maximization and mixture modeling tutorial. *Statistics Online Computational Resource, UC Los Angeles*, September 2008.

[20] M.D. Fox, D. Zhang, A.Z. Snyder, and M.E. Raichle. The global signal and observed anticorrelated resting state brain networks. *Journal of Neurophysiology*, 101:3270–3283, 2009.

[21] P. Fransson. How default is the default mode of brain function ? further evidence from intrinsic bold signal fluctuations. *Neuropsychologia*, 44:2836–2845, 2006.

[22] C. Goutte, P. Toft, E. Rostrup, F.A. Nielsen, and L.K. Hansen. On clustering fmri time series. *Neuroimage*, 9:298–310, 1999.

[23] M. Hampson, B.S. Peterson, P. Skudlarski, J.C. Gatenby, and J.C. Gore. Detection of functional connectivity using temporal correlations in mr images. *Human brain mapping*, 15:247–262, 2002.

[24] A. Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. on Neural Networks*, 10(3):626–634, 1999.

[25] A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* Wiley-Interscience, 2001.

[26] A. Hyvärinen and E. Oja. Independent component analysis : Algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

[27] I.T. Jolliffe. *Principal Component Analysis.* Springer, second edition, 2002.

[28] M. Journée, Y. Nesterov, P. Richtarik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *CORE Discussion Paper 2008/70*, 2008.

[29] N. K. Logothetis, J. Pauls, M. Augath, T. Trinath, and A. Oeltermann. Neurophysiological investigation of the basis of the fmri signal. *Nature*, 412, 2001.

[30] X. Long, X. Zuo, V. Kiviniemi, Y. Yang, Q. Zou, C. Zhu, T. Jiang, H. Yang, Q. Gong, L. Wang, K. Li, S. Xie, and Y. Zang. Default mode network as revealed with multiple methods for resting-state functional mri analysis. *Journal of Neuroscience Methods*, 171:349–355, 2008.

[31] D. Luenberger. Optimization by vector space methods. *Wiley*, 1969.

[32] R.E. Madsen, L.K. Hansen, and O. Winther. Singular value decomposition and principal component analysis. *Neural Networks*, pages 1–5, 2004.

[33] S. Mahajan. Application of pca, s-pca and ica in fmri data exploratory analysis. *Internship report - University of Liege*, 2009.

[34] M.J. McKeown, L.K. Hansen, and T.J. Sejnowski. Independent component analysis of functional mri: what is signal and what is noise? *Current Opinion in Neurobiology*, 13:620–629, 2003.

[35] M.J. McKeown, S. Makeig, G.G. Brown, T. Jung, S.S. Kindermann, A.J. Bell, and T.J. Sejnowski. Analysis of fmri data by blind separation into independent spatial components. *Human Brain Mapping*, 6:160–188, 1998.

[36] M.J. McKeown and T.J. Sejnowski. Independent component analysis of fmri data : Examining the assumptions. *Human Brain Mapping*, 6:368–372, 1998.

[37] L. Pauling and C. D. Coryell. The magnetic properties and structure of hemoglobin, oxyhemoglobin and carbonmonoxyhemoglobin. *Proceedings of the National Academy of Sciences*, 22:210–216, march 1936.

[38] V. Perlbarg and G. Marrelec. Contribution of exploratory methods to the investigation of extended large-scale brain networks in functional mri : Methodologies, results, and challenges. *International Journal of Biomedical Imaging*, 2008(218519):14, 2008.

[39] M. E. Raichle, A. M. MacLeod, A. Z. Snyder, W. J. Powers, and D. A. Gusnard. A default mode of brain function. *Proceedings of the National Academy of Sciences*, 98(2):676–682, January 2001.

[40] M. E. Raichle and M. A. Mintun. Brainwork and brain imaging. *Annual Review of Neuroscience*, 26:449–476, 2006.

[41] M. E. Raichle and A. Z. Snyder. A default mode of brain function: A brief history of an evolving idea. *Neuroimage*, 37:1083–1090, 2007.

[42] C.S. Roy and C. S. Sherrington. On the regulation of the blood-supply of the brain. *Journal of Physiology*, 11(1-2):58–185, january 1890.

[43] J. Shlens. A tutorial on principal component analysis. *Systems Neurobiology Laboratory, Salk Insitute for Biological Studies*, 3.01, April 2009.

[44] A. Shmuel, M. Augath, A. Oeltermann, and N. K. Logothetis. Negative functional mri response correlates with decreases in neuronal activity in monkey visual area v1. *Nature neuroscience*, 9(4):569–577, 2006.

[45] K. Sjöstrand, T.E. Lund, K.H. Madsen, and R. Larsen. Sparse pca a new method for unsupervised analyses of fmri data. In *Proc. International Society of Magnetic Resonance In Medicine*. ISMRM, May 2006.

[46] K. Sjöstrand, B. Stegmann, and R. Larsen. Sparse principal component analysis in medical shape modeling. *Progress in biomedical optics and imaging*, 7(3), 2006.

[47] E. Snitkowska and W. Kasprzak. Independent component analysis in angiography images. *Warsaw University of Technology.*

[48] A. Soddu. Constrained connectivity ica graphs of default mode brain function in vegetative and locked-in state patients. In *JSMF Coma and Consciousness Consortium Meeting*. Weill Cornell Medical College, New York.

[49] A. Soddu, A. Vanhaudenhuyse, M. A. Bahri, M.-A. Bruno, M. Boly, A. Demertzi, J.-F. Tshibanda, C. Phillips, M. Stanziano, S. Ovadia-Caro, Y. Nir, P. Maquet, M. Papa, R. Malach, S. Laureys, and Q. Noirhomme. Identifying the default-mode component in spatial ic analyses of patients with disorders of consciousness. *Human Brain Mapping*, April 2011.

[50] H.N. Suma and S. Murali. Principal component analysis for analysis and classification of fmri activation maps. *International Journal of Computer Science and Network Security*, 7(11):235–242, 2007.

[51] G. Tononi, O. Sporns, and G. M. Edelman. A measure for brain complexity: Relating functional segregation and integration in the nervous system. *Proceedings of the National Academy of Sciences*, 91:5033–5037, 1994.

[52] M. Uzumcu, A.F. Frangi, J.H. Reiber, and B.P. Lelieveldt. Independent component analysis in statistical shape models. *Proceedings of SPIE*, 5032:375–383, 2003.

[53] R. Viviani, G. Grön, and M. Spitzer. Functional principal component analysis of fmri data. *Human Brain Mapping*, 24:109–129, 2005.

Appendices

A Additional figures

A.1 Time courses of the components

Figure 29 shows the effect of removing the first principal component on the power spectra of the time courses of the components for one subject : LA. Here are the time courses of the same components without the removal of the first principal component :

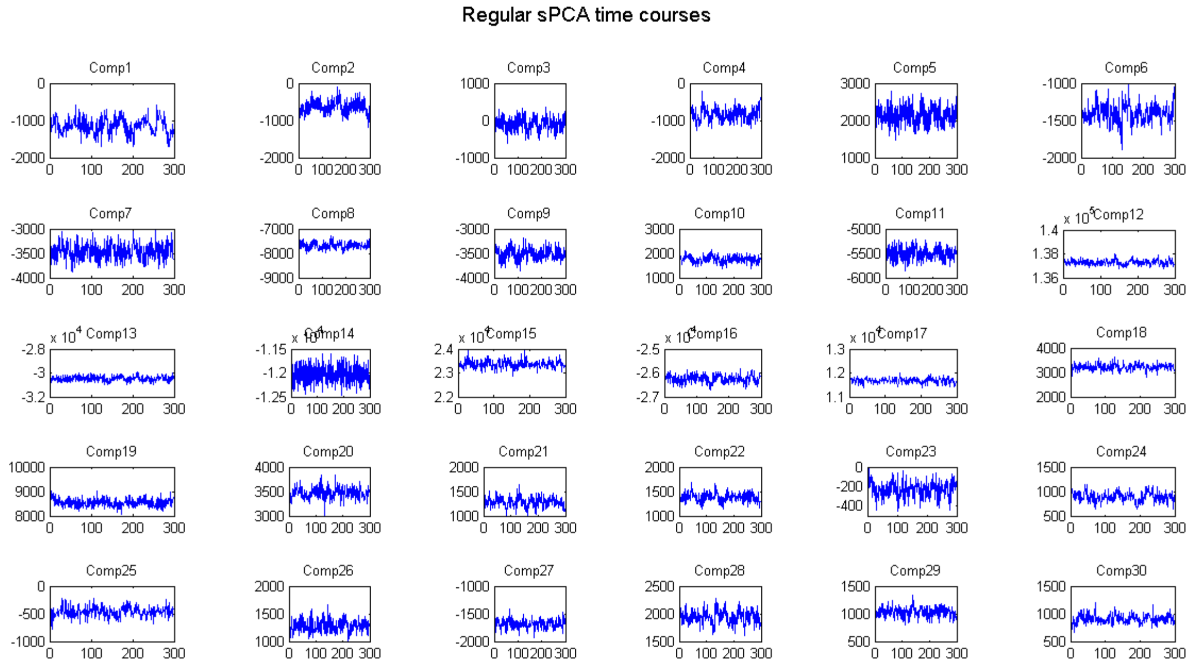

Figure 44: Time courses of the sPCA components

And with the removal of the first principal component :

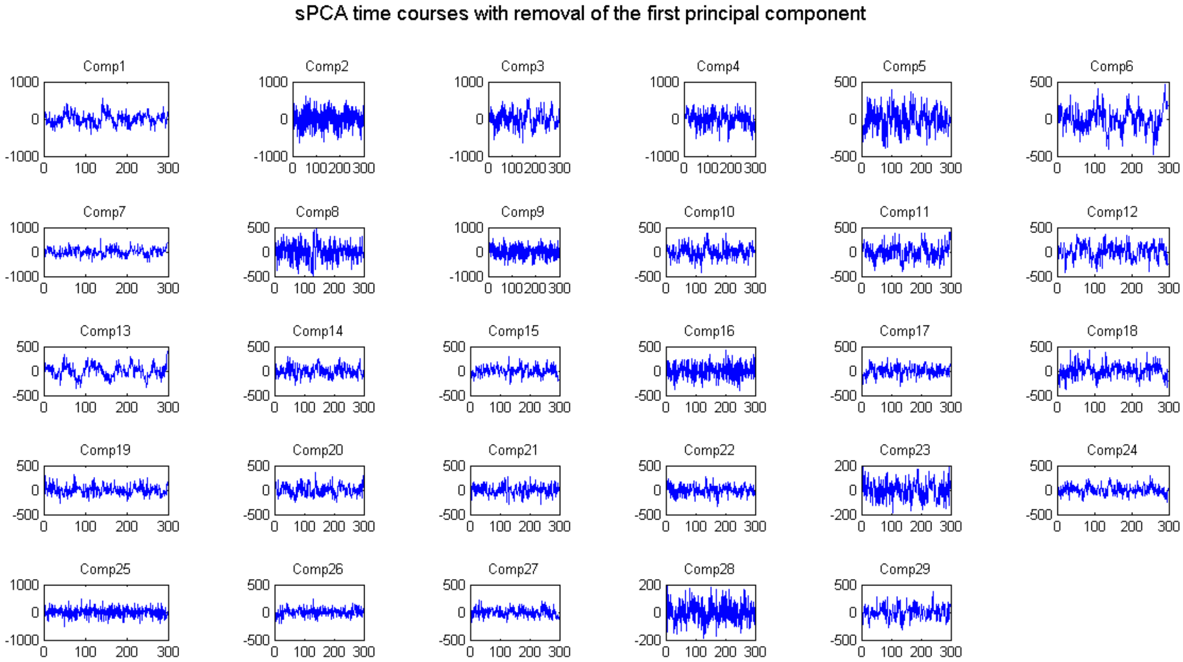

Figure 45: Time courses of the sPCA components after removal of the first principal component

Similar graphs can be obtained by running `Rem_TC_PS` on the other subjects.

A.2 Quality of the model map for other subjects

In the last experimental section we presented the quality of the model map as a function of the number of components used to build the model map (see figure 39). Here are the curves for two other subjects and using η.

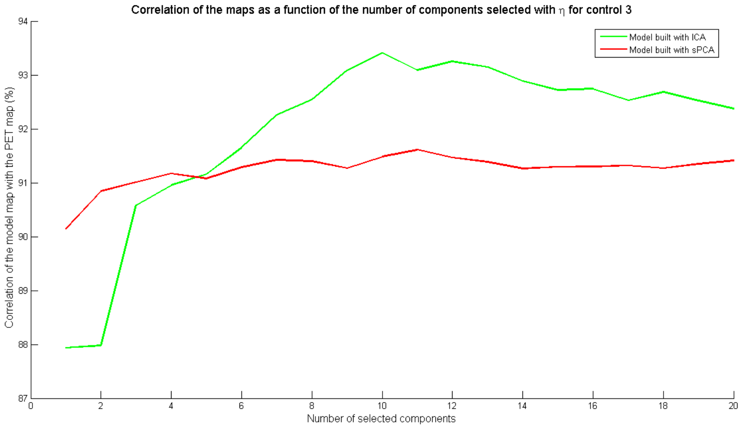

Figure 46: Correlation between the maps as a function of the number of selected components

We can see that the first sPCA components seem to contain more neuronal information than the ICA components. However, when more components are selected ICA gives a better model map than sPCA, as in all the other subjects. In this case there seems to be approximately ten neuronal components.

We finally show the case of a decomposition where the correlation for the ICA curve is oscillating, which seems to indicate a decomposition of lower quality :

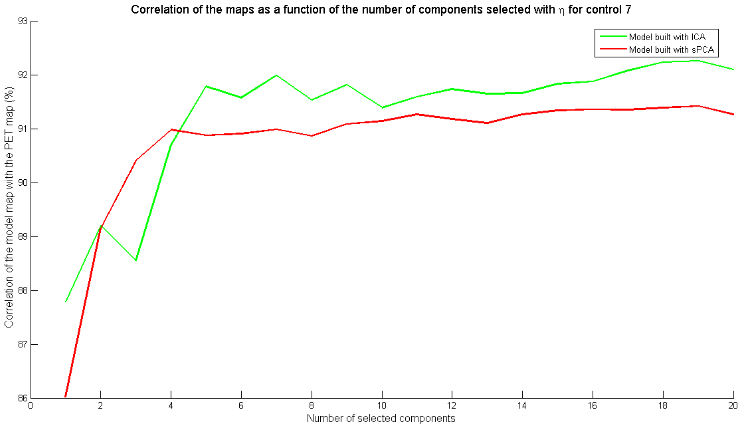

Figure 47: Correlation between the maps as a function of the number of selected components

B List of matlab codes and additional data

On the additional electronic support we furnish the data of one patient from each of the three experiments as well as the Matlab saves related to those patients. "LA_CONTROL" and "DD_LIS" are a control and a LIS patient of the second experiment (section 5.3), "MF_MCS" is the data related to the "half brain" patient (section 5.2) and "BL" contains the data related to one subject of the last experiment (section 5.4). Note that only the initials of the patients' names are available.

We now give a list of the Matlab codes used in each section and for each experiment. Those codes are also included in the additional electronic support.

Section or Experiment	Name	Description
Component Analysis	`GPower`	Computation of PCA and sPCA (author see [28]).
	`fastica`, `whitenv`, `fpica`, `remmean`, `pcamat`	Computation of fast ICA (author see [25]).
	`fpicaM`	Modified version of `fpica` in order to be able to follow the trajectory of the fixed point iteration, look for `w_rand` for the changes.
	`show_trajectory`	Shows the trajectory of the fixed point iteration. (Figures 5, 6, 7 and 8).
	`PCA_comprehension`	Application of PCA on a toy example and comparison with ICA. (Figure 4).
	`ICA_comprehension`	Example of ICA decomposition.
Toy Example	`toy_example`	Construction of the fMRI data model and computation of ICA and sPCA decompositions. (Figures 13, 15, 16 and 17).
	`t_e_analysis`	Analysis of the quality of a decomposition.
	`t_e_analysis_next`	Uses `toy_example1` and `toy_example_analysis` to compute the quality of the decomposition as a function of a varying parameter. (Figures 18 and all the figures of section 4.2).
	`t_e_remove_motion`	Attempts to extract the motion from the initial data. (Figure 24).
	`compute_kurtosis`	Computes the kurtosis.

Analysis of a "half brain" patient	`MCS`	Computes the ICA and sPCA analysis of the "half brain" patient.
	`Correlation_neur`	Computes the positions of the extracted components on a map as presented on figures 31 and 32.
Methodology	freq_analysis	Highly inspired from Mr. Mahajan's `publish5bandPS` code. Allows to compute the power spectrum of a given time course.
	Rem_TC_PS	Based on the results of `Extract_DM1` and `Extract_DM1_rem_1st` (see codes just below), gives the power spectra and time courses of each component for a subject. (Figures 29, 44 and 45).
Extracting the default mode	`Extract_DM1`	Computes ICA and sPCA decompositions on the five controls and two LIS patients and compares the default modes properties.
	`Ex_DM1_rem_1st`	Same as `Extract_DM1` with the first principal component that has been removed. (Figures 35 and 36).
	`Final_DM`	Computes the average and standard deviations of the DMs of the nine subjects presented in figure 37.
From fMRI to PET	`load_nii`, `make_nii`, `save_nii`	Allows to deal with nii files. Downloaded from MatlabCentral website, author : Jimmy Shen.
	`Final_init`	Loads the nii data (3D) and performs linearization.
	`Final_treatment`	Performs the ICA and sPCA decompositions on the data.
	`Final_for`, `Final_to_pet`, `Final_mask`	Not furnished because contains information about a future publication of Dr. A. Soddu. Computes the model map and the results of figures 39, 40 and 41.

Future directions	`em_1dim`	Computes a mixture of gaussians model based on the expectation maximization principle (see [19]). Downloaded from Aquaphoenix website, Matlab, Lecture 10.
	`Final_Gaussian`	Computation of the mixture of gaussians models for the results of figure 42. (Figure 43).
Codes written by Mr. S. Mahajan and used in this thesis	`vtc2matrix`, `vtcMoveToAHP`, `writeVMP`, `createvmpfromvector`, `inflateMatX3`	Allows to import and export data from Matlab to formats that are readable with BrainVoyager Viewer and conversely.
	`publish5bandPS`	Computes the power spectrum of a given time course.

Table 5: List of Matlab codes