

Manual for the `ica4spm` toolbox

Troels Bjerre[†], Jonas Henriksen[†], Carsten Haagen Nielsen[†]

Peter Mondrup Rasmussen^{††} Lars Kai Hansen^{††}, Kristoffer Hougaard Madsen[‡]

[†]*DTU Electrical Engineering, Technical University of Denmark,*

Ørsteds Plads Building 348, 2800 Kgs. Lyngby, Denmark

^{††}*DTU Informatics, Technical University of Denmark,*

Richard Petersens Plads Building 321, 2800 Kgs. Lyngby, Denmark

[‡]*DRCMR, Copenhagen University Hospital Hvidovre, Kettegaard Allé 3, 2650 Hvidovre, Denmark*

troels.bjerre@gmail.com, medicoing@gmail.com, chn@stanford.edu, peter.mondrup@gmail.com

lkh@imm.dtu.dk, kristoffer.madsen@gmail.com

January 1, 2009

1 Introduction

This tutorial is for the `ica4spm` toolbox described in [2]. `ica4spm` is a toolbox for exploratory analysis of functional magnetic resonance imaging (fMRI) data using independent component analysis (ICA) within the widely used SPM analysis pipeline. The toolbox enables dimensional reduction using principal component analysis (PCA), ICA using several different ICA algorithms, selection of the number of components using the Bayesian information criterion (BIC), visualization of ICA components, and extraction of components for subsequent analysis using the standard general linear model. The toolbox is capable of identifying activity and nuisance effects in fMRI experiments.

2 Installation and execution

MATLAB and SPM5 is required. Version 1.0 of the toolbox is developed for MATLAB R2008a. In addition, the Signal Processing toolbox for Matlab

is required. The `ica4spm` toolbox files should be unpacked to the subfolder `spm5/toolbox/ica4spm` in the SPM5 installation directory.

When running SPM5 under MATLAB, the `ica4spm` toolbox can be selected in the toolbox drop-down menu in the SPM5 interface. `ica4spm` can also be started from the Matlab prompt by typing `ica4spm`. This starts the `ica4spm` main user interface, shown in Figure 1.

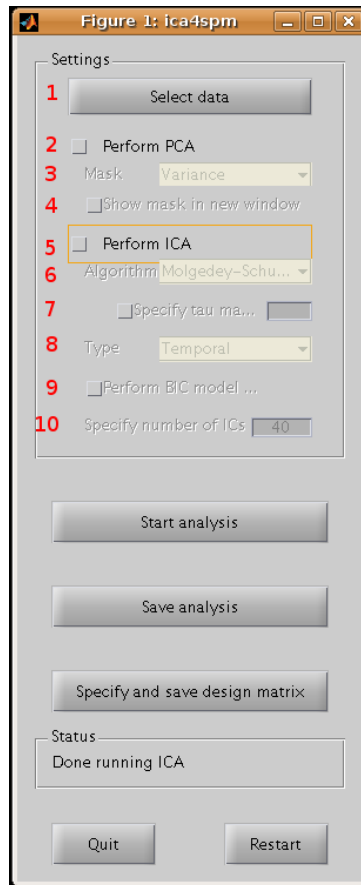


Figure 1: The main user interface of the `ica4spm` toolbox. The different options are numbered on the figure for description purposes, see section 3

3 Main user interface

From the main user interface, the user is able to specify the options wanted for the analysis. After specifying the options, the analysis is performed by pressing the button *Start Analysis*. The following paragraphs describe the functionality and implementation of the different settings in the main interface, with reference to the numbers assigned to the described boxes and fields in Figure 1:

1. fMRI data or a structure containing already analyzed data is loaded by pressing the button *Select data*. The SPM5 function `spm_select` is called, meaning that the data is selected in the same way as in SPM5.
2. The *Perform PCA* check box should be checked in order to perform principal component analysis of the data, using the function `runPCA`. The decomposition is performed with the MATLAB function `svd`. The PCA returns a dataset with reduced dimensionality which is needed in order to perform the ICA. When checked, the sub-controls to PCA will be enabled.
3. From the *Mask* drop-down box it is possible to choose how to mask the data before performing the PCA. This is needed in order to prevent MATLAB from running out of memory. Masking can be performed on basis of mean value or variance of the voxels across a time series. The data is thresholded so that either a fraction of voxels with highest mean values or a fraction with greatest variance are included in the analysis. The variance based masking is selected as default. The user can also load a user defined mask. After masking, a high-pass filter is applied to the dataset in order to remove low frequency drift. The filtering is performed by the SPM5 function `spm_filter` and the cut-off period is 60 s.
4. If it is a question whether the mask is applicable, the check box *Show mask in new window* can be used to display black and white images of the mask. The white areas mark the areas included in the analysis.
5. The *Perform ICA* check box should be checked in order to perform independent component analysis of the dataset with reduced dimensionality, using the function `runICA`. When checked, the subcontrols for the ICA will be enabled.
6. From the *Algorithm* drop-down box it is possible to choose which ICA separation method to use for the analysis. `ica4spm` offers the following algorithms:

- Maximum Likelihood (also known as Infomax) [1, 4]
 - Molgedey-Schuster [7, 4, 6]
 - Joint approximate diagonalization of eigenmatrices (JADE) [3]
 - Fast Fixed-Point Algorithm for Independent Component Analysis (FastICA) [5]
7. When the MS algorithm is chosen, it is possible to manually select a desired value of τ . As default, `icaMS` will estimate an optimal value.
 8. From the *Type* drop-down box it is possible to choose whether to perform a Spatial (sICA) or a Temporal (tICA) analysis.
 9. When the *Perform BIC model selection* check box is checked, the function `runBIC` estimates the optimal number of principal components for the data subspace on which the ICA is performed. After the estimation, the user is prompted to accept the proposed number of components or to manually specify another number. The function will plot the logarithmic likelihood of the number of components, given the data input, in order for the user to verify that the proposed value is correct.
 10. When it is selected to perform BIC model selection, it is possible to specify the maximum number of components for the `runBIC` function to investigate. When BIC model selection is unselected, it is possible to manually specify the wanted number of independent components.

When pressing the *Save analysis* button, it is possible to save the analyzed data which is stored in a MATLAB data structure called `ica`. When pressing the *Specify and save design matrix* button, the user will be prompted to specify wanted independent components, which will then be saved in a folder specified by the user. The *Restart* button clears the workspace, closes all windows and restarts the `ica4spm` GUI. This might be necessary after analyzing a large dataset, in order to prevent MATLAB from running out of memory. The *Quit* button terminates the GUI without saving the analysis data.

4 ICA visualization interface

After running the analysis, the result is displayed in the ICA visualization window shown in Figure 2. The following paragraphs describe the functionality and implementation of the different settings in this interface, with reference to the numbers assigned to the described boxes and fields in the figure:

1. The SPM5 orthogonal visualization function `spm_orthviews` is used to display the T1-weighted anatomical volume of the brain. The red blobs superimposed on the anatomical volume mark the voxels with greatest intensity in the selected independent component, while the blue blobs mark the voxels with the least intensity.
2. The percentage of greatest/least intensity voxels shown for the currently selected component, can be changed using the *Component visibility* slider.
3. The *Variance information* field give information about the total variance, the variance of the selected component and the relative variance of the selected component.
4. The cross-hair position in the brain can be changed by clicking in one of the three orthogonal images or specifying the coordinates in the fields. The coordinates are given in mm.
5. In the *IC component display* field, it is possible to choose which independent component to display.
6. In the *Time series* plot, the temporal development of the selected component is shown. In the *Time series plot settings* area, it can be chosen to view only the selected component (default) or all components. Checking the *Show paradigm* box plots the paradigm together with the time series. The data file containing the paradigm should be named `paradigm.mat` and contain two variables; `t`, `paradigm`, the time and some dummy variable for the paradigm. *It should be stored in the same directory as the acquisitions.* If no paradigm file is stored, the user will be notified by an error dialog box.
7. The *frequency* plot shows the Power Spectral Density estimate via Welch's method.

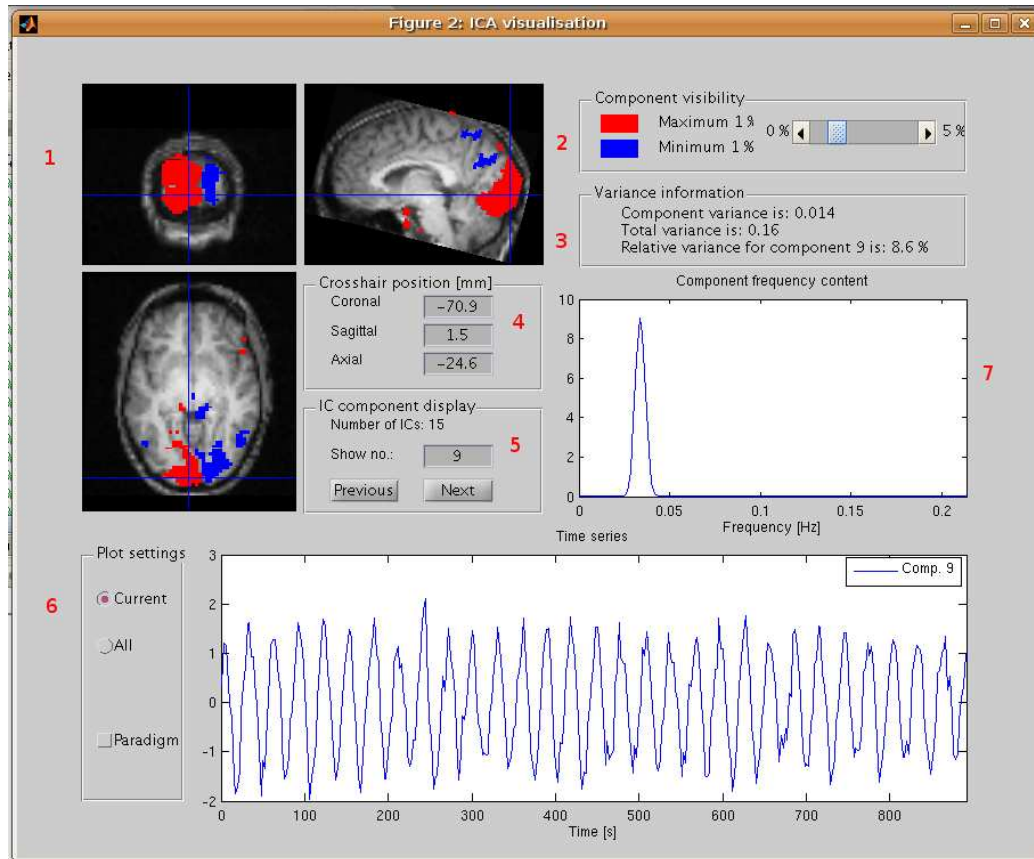


Figure 2: The ICA visualization interface of the `ica4spm` toolbox showing component 5 of the ICA performed on a training data set. Here the training data set is analyzed with a temporal Molgedey-Schuster ICA where the program chose the most optimal value of τ . The different options are numbered on the figure for description purposes, see section 4

References

- [1] A. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.
- [2] T. Bjerre, J. Henriksen, P.M. Rasmussen, C.H. Nielsen, L.K. Hansen, and K.H. Madsen. Unified ica-spm analysis of fMRI experiments - implementation of an ICA graphical user interface for the SPM pipeline. In *Proceedings of BIOSTEC - BIOSIGNALS 2009 Conference*, 2009.
- [3] Jean-François Cardoso. High-order contrasts for independent component analysis. *Neural Comput.*, 11(1):157–192, 1999.
- [4] LK Hansen, J. Larsen, and T. Kolenda. Blind detection of independent dynamic components. *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, 5, 2001.
- [5] Aapo Hyvärinen and Erkki Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.
- [6] T. Kolenda, L.K. Hansen, and J. Larsen. Signal detection using ICA: Application to chat room topic spotting. *Third International Conference on Independent Component Analysis and Blind Source Separation*, pages 540–545, 2001.
- [7] L. Molgedey and H. Schuster. Separation of independent signals using time-delayed correlations. *Physical Review Letters*, 72(23):3634–3637, 1994.