FMRI ICA Classify: ICA-based BOLD fMRI Artifact Remover under Matlab - version 1.1 (beta). Documentation

Jussi Tohka

January 23, 2008

1 INTRODUCTION

Blood oxygenation level dependent (BOLD) signals in functional magnetic resonance imaging (fMRI) are often small compared to the level of noise in the data. The sources of noise are numerous including different kinds of motion artifacts and physiological noise with complex patterns. It has been suggested that the contributions of some of these artifacts to fMRI timeseries can be detected using independent component analysis (ICA). If an independent component reflecting an obvious artifact can be automatically detected, then its removal from the subsequent statistical analysis of the data becomes straightforward. This package of the Matlab functions implements an automatic classifier trained to distinguish between independent components (ICs) reflecting an obvious artifact and ICs possibly related to true BOLD signals. The method has been described in [2].

Please read carefully this document and the article [2] before using this software. For licensing conditions, please see the m-files in this package.

This is still considered to be a beta release.

2 REQUIREMENTS

- FMRI ICA classify software runs under Matlab (Mathworks, Natick, MA). Hence, Matlab is needed. The Matlab versions 5.x, 6.x, and 7.x should be OK although the versions 5.x have not been tested. The software not relying on Matlab will be made available later on.
- FMRI ICA classify software is designed to work with the Melodic ICA-software from the FSL. The FSL software is freely available for research purposes. Note that the software currently relies on the naming conventions etc. of the Melodic 2.0 while the current version is 3.0. The version of this software which will be compatible with Melodic 3.0 will be made available later on. The component identification method is independent from the ICA algorithm used, but the training of the classifiers was performed based on the output of the Melodic 2.0. This may have an effect on the classifiers.
- FMRI ICA classify software also requires NIfTI tools by Jimmy Shen. These can be downloaded at `http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=8797&objectType=FILE`
- FMRI ICA classify software has been tested under Linux and Unix.

3 INSTALLATION

Just copy the zip-package into the same directory where the NifTI tools have been placed and unzip.

ica_directory	the directory where Melodic output lies.
classifier	the classifier with the desired Neyman Pearson threshold. See section classifiers for more information. Here you can give either a classifier struct or the name of the mat file containing the classifier in the variable 'gdt'. Whether a classifier or a file name is given is deduced automatically
blocked	0 if event related design (target frequency assumed to be [0.01 Hz - 0.1 Hz]) 1 if blocked design with the target frequency of 10 cycles n if blocked design with the target frequency of n cycles
denoise	'false' (default) this function only returns the desired string 'true' this function executes the desired melodic filtering command 'file' produces a shell script named ica_denoise.sh that, when run, executes the filtering command. The file is written in the working directory.
odir(optional)	sets the output directory of the melodic filtering. If you want to set the tr and flimits parameters but keep the default melodic filtering
tr(optional)	The repetition time (TR) in ms. Defaults to 2000. If you're using a considerably different TR than 2000, the classifier may not work well for the event related case.
flimits(optional)	A 3 component vector which sets 1) low frequency limit, 2) the lower limit of the target frequency and 3) the upper limit of the target frequency. All should be given in Hertzs. 2) and 3) are ignored for the blocked designs.

Table 1: Input arguments

4 INTENDED USAGE

FMRI ICA Classify can be used to remove obviously artifactual components for subsequent GLM-analysis. The motivation is that artifacts that have complex (spatial and/or temporal) patterns will complicate the estimation of the GLM parameters as well as the variance required to build statistical parametric maps. In this case, the 'pipeline' is as follows: 1) Pre-processing (e.g. motion correction, band-pass filtering), 2) ICA, 3) denoising by detecting the artifactual ICs and removing them from the data, and 4) the GLM-based analysis of the denoised data.

As indicated in [2], the denoising might be beneficial to the ICA analysis itself if there are strong artifacts. There can be also other applications as the software needs to make very few assumptions about the behavioral paradigm under study.

5 USING THE SOFTWARE

5.1 Basic usage

The main Matlab function is `fmri_ica_classify`. Use it as

```
[filter_str,rejected,accepted,cls] =
 fmri_ica_classify(ica_directory,classifier,blocked,denoise,
 [odir],[tr],[flimits]);
```

The details about the input arguments are given in Table 1 and the outputs are specified in Table 2.

File types: The 4-D timeseries files are expected to be either in 1) Analyze 7.5 or 2) NiFTI format. The program deduces automatically which type of files you use. The program supports the use of gzip compressed files under Unix-based operating systems.

filter_str	string containing the melodic filtering command. Note that this works for Melodic 2.0.
rejected	the indexes of the rejected components
accepted	the indexes of the accepted components
cls	0 for each accepted component and 1 for each rejected component

Table 2: Output

5.2 Example

For example

```
[fs,rej,acc] =
fmri_ica_classify('/data/fmri/subj1/EPI004/S004_4D_mcf_brain.ica',
                  '/worktmp/matlab/fmri/gdt5_blocked_tr005',
                   1,'false')
```

produces

```
>> disp(fs)
melodic -i /data/fmri/subj1/EPI004/S004_4D_mcf_brain
-v -o /data/fmri/subj1/EPI004/filtered_data.ica.auto
--mix=/data/fmri/subj1/EPI004/S004_4D_mcf_brain.ica/melodic_mix
-f 2,5,6,8,9,10,18,25,27,34,35,39,41
>> disp(rej')
     2     5     6     8     9    10    18    25    27    34    35    39    41
>> disp(acc')
  Columns 1 through 13

     1     3     4     7    11    12    13    14    15    16    17    19    20

  Columns 14 through 26

    21    22    23    24    26    28    29    30    31    32    33    36    37

  Columns 27 through 30

    38    40    42    43
```

When the command in 'fs' has been run, the denoised data can be found in directory `/data/fmri/subj1/EPI004/filtered_data.ica.auto`.

5.3 Using the software to denoise multiple timeseries

There are several options. One possibility is to collect all ICA directories to the variable (say) `icadirs`:

```
icadirs{1} = '/data/fmri/subj1/EPI004/S004_4D_mcf_brain.ica';
icadirs{2} = '/data/fmri/subj1/EPI005/S005_4D_mcf_brain.ica';
icadirs{3} = '/data/fmri/subj1/EPI006/S006_4D_mcf_brain.ica';
```

And then run following loop:

```
for i = 1:3
  [fs,rej,acc] =
  fmri_ica_classify(icadirs{i},’/worktmp/matlab/fmri/gdt5_blocked_tr005’,
                    1,’false’)
  !(fs)
end
```

Other way would be to collect all the filtering commands into a single cell array and then write these into a shell script file.

6 FILE NAMING CONVENTIONS

The FMRI ICA classify software assumes the following file naming conventions, which follow the default settings of the Melodic 2.0. You may need to edit the source code, if the conventions are not equal of the desired settings. The following conventions are in effect:

- The `ica_directory` is expected to contain following files: 1) `melodic_IC.nii` (or `melodic_IC.img` and `melodic_IC.hdr`) 2) `mask.nii` (or `mask.img` and `mask.hdr`), 3) `melodic_mix`, 4) `melodic_FTmix`; These file names can be easily adapted by editing the function `fmri_readicanii` in the file `fmri_ica_classify.m`.
- The `ica_directory` name is assumed to be composed by changing the extension of the original timeseries filename to `.ica`. You’ll need to edit either the call to `fmri_writeicafilter` in `fmri_ica_classify` or `fmri_writeicafilter` in the file `fmri_ica_classify.m`. There are some more help in the preamble of `fmri_writeicafilter`.
- The denoising output (by `melodic -i`) is written to the sub-directory `filtered_data.ica.auto` of the directory that contains the original Melodic output directory. For example, if the `ica_directory` is `/longpathname/S004_mcf.ica` then the denoised timeseries is written into `/longpathname/filtered_data.ica.auto`. If you want to change this, please edit the call to `fmri_writeicafilter` in `fmri_ica_classify`. See the premeable of `fmri_writeicafilter` for more help.

7 THE CLASSIFIERS

There are separate classifiers for the event related and block design cases. Event related classifiers should always be used for event related data, and block design classifiers should be used for block design data. The classifiers are based on the data from 20 subjects. The ICs of these data have been manually classified into ’obviously artifactual’ and ’possibly task related’. Global decision tree classifiers with Neyman-Pearson thresholds of $0.05, 0.075, 0.1, 0.125, 0.15, 0.175$ and 0.2. has been trained with these data. For example, the filename for block design classifier with the NP threshold of 0.05 is `gdt5_blocked005.mat` and the filename for event related design classifier with the NP threshold of 0.05 is `gdt5_event020.mat`.

The data for the classifier design came from a category learning task (for a complete description of the behavioral paradigm see [1]). Other important characteristics of the training data were as follows: Imaging was performed with a 3T Siemens Allegra head-only MR scanner. The images were collected using a gradient-echo echo-planar pulse sequence with interleaved acquisition (TR=2000ms, TE=30ms, 64×64 matrix, 3.125mm × 3.125mm pixel size, 25 slices, 4mm slice thickness/1mm gap, 200mm FOV). Four images at the beginning of each run were discarded to allow T1 equilibration. The target frequency for the blocked design was 10 cycles. One image series consisted 180 time points for the blocked design and 238 time points for the event related design.

8 IMPORTANT NOTES

- If your data is acquired using a considerably behavioral paradigm, equipment, or equipment settings we strongly suggest to perform validation of the rejected components before using the software.
- Particularly, the fact that the acquisition is interleaved is utilized by the classifier. So proceed with extra care if the acquisition was not interleaved.
- If your data has a considerably different number of time points, or different TR than the training data, there can be some effects on the features 1 and 2, especially in the block design case. We have incorporated a heuristic correction factor to these features, but we have not carefully evaluated its performance. Thus, extra attention should be given to these cases.

9 FUTURE EXTENSIONS

The following extensions are likely to be made available in near future:

- The Matlab-based ICA classification for Melodic 3.0.
- The global decision tree training algorithm to train your own classifiers.
- The C-code for the classification/classifier training to be able to train/apply the classifier without Matlab.

References

[1] A.R. Aron, M.A. Gluck, and R.A. Poldrack. Long-term test-retest reliability of functional mri in a classification learning task. *NeuroImage*, 29(3):1000 – 1006, 2006.

[2] J. Tohka, K. Foerde, A.R. Aron, S.M. Tom, A.W. Toga, and R.A. Poldrack. Automatic independent component labeling for artifact removal in fmri. *NeuroImage*, 39:1227 –1245, 2008.