

Manual for ICA-AROMA

1 Introduction

ICA-AROMA (i.e. ‘ICA-based Automatic Removal Of Motion Artifacts’) attempts to identify and remove motion artifacts from fMRI data. To that end it exploits Independent Component Analysis (ICA) to decompose the data into a set of independent components. Subsequently, ICA-AROMA automatically identifies which of these components are related to head motion, by using four robust and standardized features. The identified components are then removed from the data through linear regression as implemented in `fsl_regfilt`. ICA-AROMA has to be applied after spatial smoothing, but prior to temporal filtering within the typical fMRI preprocessing pipeline. Two manuscripts provide a detailed description and evaluation of ICA-AROMA:

1. Pruim, R.H.R., Mennes, M., van Rooij, D., Llera, A., Buitelaar, J.K., Beckmann, C.F., 2015, ICA-AROMA: A robust ICA-based strategy for removing motion artifacts from fMRI data. *NeuroImage*
2. Pruim, R.H.R., Mennes, M., Buitelaar, J.K., Beckmann, C.F., 2015. Evaluation of ICA-AROMA and alternative strategies for motion artifact removal in resting-state fMRI. *NeuroImage*

2 General info

The ICA-AROMA package contains two python scripts; `ICA_AROMA.py` and `ICA_AROMA_functions.py`. The first is the main script to be called by the user. The second contains the functions used by the main script. The package furthermore contains three spatial maps (CSF, edge & out-of-brain masks) which are required to derive the spatial features used by ICA-AROMA. Check the ‘help’ function of `ICA_AROMA.py` for information on the argument options for running ICA-AROMA.

The scripting-examples provided in this manual are based on the bash shell.

3 Requirements

- FSL
- Python 2.7 (modules: os, argparse, commands, numpy, random)

4 Run ICA-AROMA - generic

For standard use, *ICA_AROMA.py* requires the following five inputs:

```
-i, -in      Input file name of fMRI data (.nii.gz)
-o, -out     Output directory name
-a, -affmat  File name of the mat-file describing the affine registration
             (e.g. FSL FLIRT) of the functional data to structural space
             (.mat file)
-w, -warp    File name of the warp-file describing the non-linear
             registration (e.g. FSL FNIRT) of the structural data to MNI152
             space (.nii.gz)
-mc          File name of the text file containing the six (column-wise)
             realignment parameters time-courses derived from
             volume-realignment (e.g. MCFLIRT)
```

Example:

```
python2.7 <path>ICA_AROMA.py -in func_smoothed.nii.gz -out ICA_AROMA
-affmat reg/func2highres.mat -warp reg/highres2standard_warp.nii.gz -mc
mc/rest_mcf.par
```

Of note, the registration files are required to transform the obtained ICA components to the MNI152 2mm template in order to derive standardized spatial feature scores. The fMRI data itself will not be subjected to any registration, transformation or reslicing!

4.1 Masking

Either the input fMRI data should be masked (i.e. brain-extracted) or a specific mask has to be specified (*-m*, *-mask*) when running ICA-AROMA.

Example:

```
python2.7 <path>/ICA_AROMA.py -in func_smoothed.nii.gz -out ICA_AROMA
-mc mc/rest_mcf.par -affmat reg/example_func2highres.mat -warp
reg/highres2standard_warp.nii.gz -m mask_aroma.nii.gz
```

We recommend not to use the mask determined by FEAT. This mask is optimized to be used for first-level analysis, as has been dilated to ensure that all ‘active’ voxels are included. We advise to create a mask using the Brain Extraction Tool of FSL (fractional intensity of 0.3), on a non-brain-extracted example or mean functional image (e.g. *example_func* within the FEAT directory).

Example to create an appropriate mask:

```
bet <input> <output> -f 0.3 -n -m -R
```

Of note, the specified mask will only be used at the first stage (ICA) of ICA-AROMA. The output fMRI data-file is not masked.

5 Run ICA-AROMA - after FEAT

ICA-AROMA is optimized for usage after preprocessing fMRI data with FSL FEAT, assuming the directory meets the standardized folder/file-structure, no temporal filtering has been applied and it was run including registration to the MNI152 template.

In this case, only the FEAT directory has to be specified (`-f`, `-feat`) next to an output directory. ICA-AROMA will automatically define the appropriate files, create an appropriate mask (see section 4.1) and use the ‘melodic.ica’ directory if available (in case ‘MELODIC ICA data exploration’ was checked in FEAT). We recommend not to run MELODIC within FEAT such that MELODIC will be run within ICA-AROMA using the appropriate mask. Moreover, this option in FEAT is meant for data exploration after full data preprocessing, as such it can be applied after ICA-AROMA, temporal high-pass filtering etc.

Example:

```
python2.7 <path>/ICA_AROMA.py -feat rest.feats -out
rest.feats/ICA_AROMA/
```

6 Output

<code>denoised_func_data</code>	Denoised fMRI data, suffixed with ‘_nonaggr.nii.gz’ or ‘_aggr.nii.gz’ depending on the requested type of denoising (see section 7).
<code>classification_overview.txt</code>	Complete overview of the classification results.
<code>classified_motion_ICs.txt</code>	List with the indices of the components classified as motion/noise.
<code>feature_scores.txt</code>	File containing the four feature scores of all components.
<code>melodic_IC_thr_MNI2mm.nii.gz</code>	Spatial maps resulting from MELODIC, after mixture modeling thresholding and registered to the MNI152 2mm template.
<code>mask.nii.gz</code>	Mask used for MELODIC.
<code>melodic.ica</code>	MELODIC output directory.

7 Additional options

7.1 Optional settings

<code>-tr</code>	TR in seconds. If this is not specified the TR will be extracted from the header of the fMRI file using ‘fslinfo’. In that case, make sure the TR in the header is correct!
<code>-d, -dim</code>	Dimensionality reduction into a defined number of dimensions when running MELODIC (default is 0; automatic estimation)
<code>-den</code>	Type of denoising strategy (default is <i>nonaggr</i>): <i>no</i> : only classification, no denoising <i>nonaggr</i> : non-aggressive denoising, i.e. partial component regression (default) <i>aggr</i> : aggressive denoising, i.e. full component regression <i>both</i> : both aggressive and non-aggressive denoising (two outputs)

7.2 MELODIC

When you have already run MELODIC you can specify the melodic directory as additional input (`-md`, `-meldir`; see example below) to avoid running MELODIC again. Note that MELODIC should have been run on fMRI data prior to temporal filtering and after spatial smoothing. Preferably, it has been run with the recommended mask (see section 4.1). Unless you have a good reason for doing otherwise, we advise to run MELODIC as part of ICA-AROMA so that it runs with optimal settings.

Example:

```
python2.7 <path>/ICA_AROMA.py -in filtered_func_data.nii.gz -out
ICA_AROMA -mc mc/rest_mcf.par -m mask_aroma.nii.gz -affmat
reg/func2highres.mat -warp reg/highres2standard_warp.nii.gz -md
filtered_func_data.ica
```

7.3 Registration

ICA-AROMA is designed (and validated) to run on data in native space, hence the requested ‘affmat’ and ‘warp’ files. However, ICA-AROMA can also be applied on data within structural or standard space. In these cases, just do not specify the ‘affmat’ and/or ‘warp’ file. Moreover, if you applied linear instead of non-linear registration of the functional data to standard space, you only have to specify the affmat file (e.g. `example_func2standard.mat`). In other words, depending on the which registration files you specify, ICA-AROMA assumes the data to be in native, structural or standard space and will run the specified registration. When you do not have a ‘affmat’ and/or ‘warp’ file available (e.g. fMRI performed with another software package then FSL), please create these files using FSL-FLIRT

(<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FLIRT/UserGuide>) and FSL-FNIRT (<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FNIRT/UserGuide>) respectively.

Example (for data in MNI152 space):

```
python2.7 <path>/ICA_AROMA.py -in filtered_func_data2standard.nii.gz
-out ICA_AROMA -mc mc/rest_mcf.par -m mask_aroma.nii.gz
```

Example (in case linear registration to MNI152 space should be applied):

```
python2.7 <path>/ICA_AROMA.py -in func_smoothed.nii.gz -out ICA_AROMA
-mc mc/rest_mcf.par -affmat reg/func2standard.mat -m mask_aroma.nii.gz
```