afxRs 3.3

by Max Wawrzyniak

Contents

1 Introduction

afxRs is designed as a straight forward tool for resting-state functional connectivity calculations. It implements a preprocessing pipeline (which uses the standard SPM12 tools) and whole brain and network connectivity analyses as well as calculation of voxelwise mirrored homotopic connectivity (VMHC), hemodynamic lag (time shift analysis) and fractional amplitudes of low-frequency fluctuations (fALFF). Lesion network mapping has also been implemented. Special attention was payed to contemporary denoising strategies as described in the current literature, e.g. Varikuti et al. (2016). See Fig 1 for an overview of signal processing strategies used by the toolbox.

2 Folder organization

Table 1: folder organisation

`data`	Preprocessed data
`denoisingOptions`	Files which contain denoising options as described in 4.1
`doc`	Documentation
`masks`	Mask and TPM images are stored here (note that 'brainmask.nii' is used for whole brain functional connectivity calculation and must not be deleted or moved)
`results`	Results of connectivity analysis
`roi`	Regions of interest derived from atlases; atlases for automatic labeling
`scripts`	Internal scripts
`templates`	Anatomical templates (for visualization purposes)

3 Preprocecing

Preprocecing is implemented with realignment (functional data), coregistration (structural → functional data), segmentation (structural data), normalization (structural and functional data) and smoothing (functional data). Slice time correction can optionally be enabled or also omitted because of principal considerations regarding common repetition times (TR) and filter cut-off frequencies. It might be noted, that there is no consensus about this in the literature. Lesion masking is possible and

will internally be achieved by zero masking of the structural image. Preprocessing also saves motion and framewise displacement plots in the output directory. These graphs should be faithfully reviewed. The preprocessing function returns a Matlab struct which contains all important information on the subject and can be used later for the firstlevel. A struct array with all subjects is also saved to `\data\sampleName\subjects.mat`. Note, that hemodynamic lag analysis probably needs additional slice time correction and that VMHC needs a symmetric template for normalization.

Matlab example 1:

```
subject = afxPreproc(func,struc,options,conditions,sampleName,subjectName[,info]);
% func must contain all functional data according to conditions, e.g.:
func{1} = fullfile(dirFuncPre,preImgs)'
func{2} = fullfile(dirFuncPost,postImgs)'
conditions{1} = 'pre';
conditions{2} = 'post';
% struc must contain the structural image
% options:
options.resliceStruct = [1 1 1]; % reslice structural data to voxel size
options.resliceFunc = [3 3 3];   % reslice functional data
options.fwhm = [6 6 6]; % functional smoothing kernel (sholud be 2-3x resliceFunc)
options.dummyScans = 4; % number of dummyscans
options.TR = 2.0;       % repetition time
% options.lesion can optionally contain a binary lesion mask
% options.sliceorder can optionally contain the slice order for slice time
% correction
% sampleName must contain name for preproc folder in data\
% subjectName must contain name of subject
% info can contain any information related to the subject, e.g. sex, age ...
```

There is also an adapted ‘preprocessing’ script specifically for the use with the already preprocessed HCP data. This script performs smoothing, masking (to save disk space) and reformats the realignment parameters. A similar struct array with all subjects is also saved to `\data\sampleName\subjects.mat`. Because no individual tissue probability maps are available with the HCP data, all subjects will refer to the tissue probability maps shipped with SPM12. Resting-state data will be saved as 16 bit integer mat-file using a specific mask (masks\devMaskHCP.nii) to save disk space. These mat-files are uncompressed to improve performace. Importantly, no spatial or temporal preprocessing other than simple smoothing will be performed.

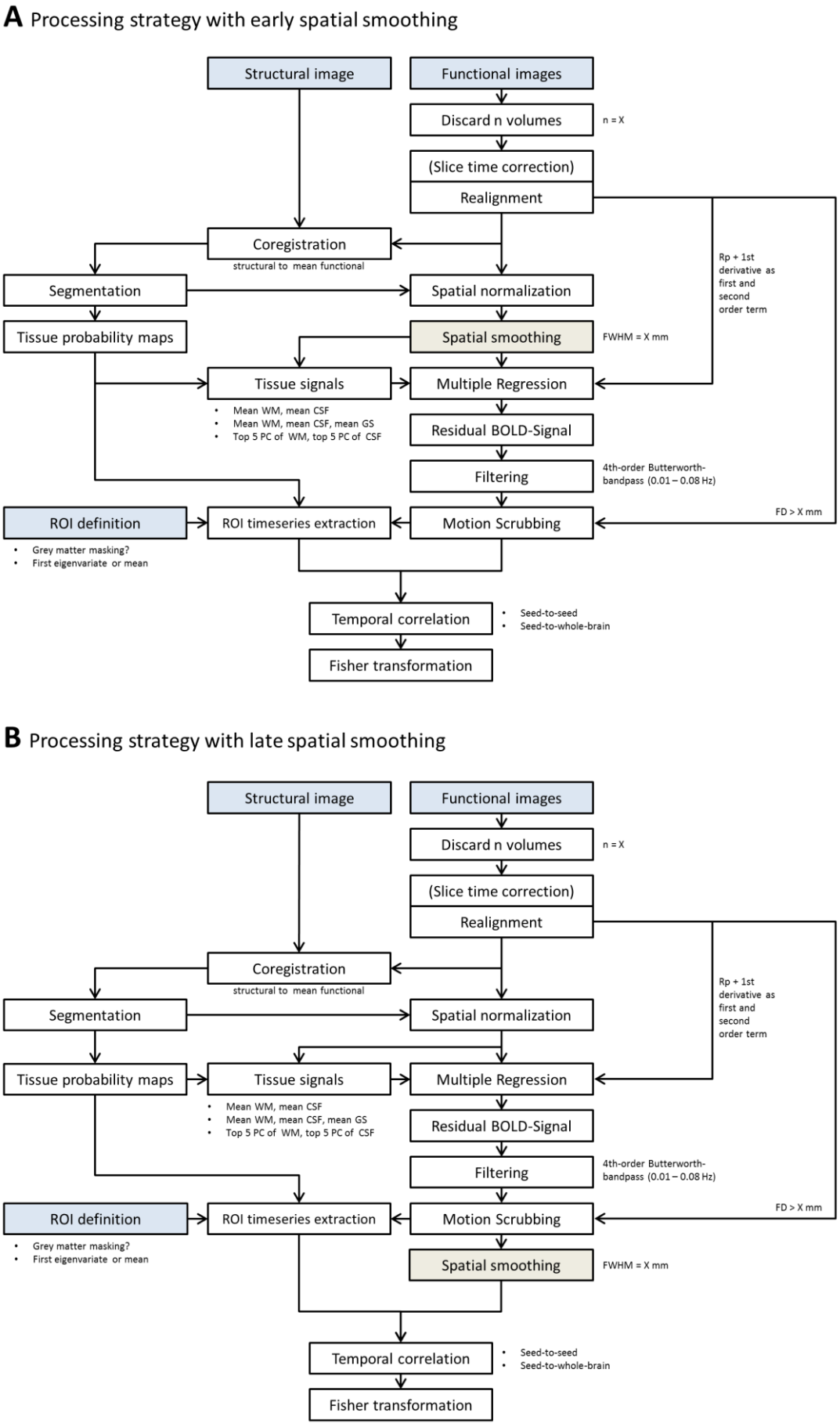

Figure 1. Processing pipeline overview with (A) early and (B) late spatial smoothing.

4 Firstlevel

Two principal strategies are implemented in the toolbox:

a. Late smoothing (Fig 1B)
    a. Load unsmoothed data
    b. Denoising
    c. Timeseries extraction (if applicable)
    d. Smoothing
    e. Calculation of functional connectivity (i.e. correlation coefficients) and other measures
b. Early smoothing (Fig 1A)
    a. Load smoothed data (smoothing performed in the preprocessing)
    b. Denoising
    c. Timeseries extraction (if applicable)
    d. Calculation of functional connectivity (i.e. correlation coefficients) and other measures

4.1 Denoising

Denoising consists of three steps:

1. Confound removal (multiple regression)
2. Bandpass filtering
3. Motion scrubbing

Varikuti et al. (2016) systematically investigated denoising strategies. Overall procedure and most thresholds are conducted from their publication.

Confound removal can remove variance explained by the following nuisance variables

- Motion parameters and their first derivative as first- and second order terms (motion parameters are obtained from the realignment procedure from the preprocessing pipeline and consist of six parameters (rigid body transformation with 3 translation and 3 rotation parameters)
- Weighted mean signal of any tissue class including global signal
- An arbitrary number of principle components of signals in any tissue class

Confound removal is achieved by multiple regression and further calculation with the residual time series. Mask thresholds for mean tissue signal or PCA calculation can be adjusted. Higher thresholds for the white matter mask are reasonable to prevent form accidental grey matter signal regression. For example, O'Muircheartaigh et al. (2016) used a white matter threshold of .75. I would suggest more conservative thresholds of .95 or .99 especially when using smoothed data. For further information on GSR and thresholds see e.g. Saad et al. (2012).

Filtering only preserves frequencies between some boundaries in the voxel time series. Common boundaries are 0.01 Hz for the high-pass filter and 0.08 or 0.1 for the low-pass filter. A 4^{th} order Butterworth bandpass filter is used to prevent from ringing artifacts near spikes.

Motion scrubbing discards volumes which are associated with high frame to frame motion as represented by framewise displacement (FD) as described by Power et al. (2012). FD represents maximum frame to frame movement within a sphere with a radius of 50 mm which represents the mean distance to the cortical surface. A common FD threshold is .5 mm. Although Power et al. (2013)

suggest a more conservative threshold of .2 mm. Resting-state sessions with less than 5 minutes (10 minutes for the HCP data) of data after motion scrubbing with a threshold of .5 mm are automatically excluded by the preprocessing script. Various precompiled denoising strategies can be found in the folder 'denoisingOptions'.

Matlab example 2:

```
denoisingOptions.regressRp = true;     % include motion parameters as explanatory
                                       % variable for the denoising
denoisingOptions.filter = [0.01 0.08]; % filter frequencies
denoisingOptions.TR = 2.0;             % TR in secs (necessary for the filtering)
denoisingOptions.regressTs(1).tpm = [0 1 1]; % tissue definition [gm wm csf]
denoisingOptions.regressTs(1).thresh = .95;  % threshold (individual probability)
denoisingOptions.regressTs(1).pca = 0;       % number of principle components
                                             % (0 -> mean; > 0 -> pca)
denoisingOptions.threshFD = .5;        % FD threshold for scrubbing
denoisingOptions.unsmoothed = true;    % operate on unsmoothed data and perform
                                       % smoothing after denoising and timeseries
                                      % extraction
denoisingOptions.sFWHM = 5;            % FWHM for late smoothing strategy

% note that the TR option is set automatically by afxFirstlevel()
```

4.2 Timeseries extraction

Definition of regions of interest

Regions of interest (ROI) can be defined as sphere or as mask image with and without individual grey matter masking. Grey matter masking is particularly important for spheres or non-functionally defined masks. Masking can be applied as median split approach or with a fixed threshold. Median split can be used to ensure identical ROI sizes in all subjects. See examples below.

Matlab example 3:

```
% sphere
rois(1).name = 'aIFG';
rois(1).type = 'sphere';
rois(1).coords = [-54 26 4];
rois(1).radius = 8;
rois(1).gmMask = .2;

% mask image
rois(2).name = 'hippCA';
rois(2).type = 'image';
rois(2).file = 'hippCA.nii';

% further gmMask options
rois(2).gmMask = 0;         % no gm masking
rois(2).gmMask = .2;        % fixed threshold for individual gm masking
rois(2).gmMask = 'split';   % median split gm masking, i.e. that half of voxels
                            % with highest gm probabilities

% note that denoisingOptions.unsmoothed determines wheter ROI timeseries
% extraction will be performed on smoothed or unsmoothed data
```

Timeseries extraction

Timeseries will then be extracted as first eigenvariate (for smoothed data) or mean (for unsmoothed data) of the signal in all voxels contained in the ROI. For information on mean signal vs. first eigenvariate, see Friston et al. (2006). Like tissue signal regression, ROI timeseries extraction can be

performed on smoothed and unsmoothed data. Extraction from unsmoothed data might be reasonable for irregularly shaped ROIs like the Hippocampus.

4.3 Functional connectivity calculation

Functional connectivity as expressed by Fisher transformed Pearson correlation coefficients will be calculated whole brain (between ROI timeseries and all voxel timeseries) and ROI to ROI between all possible pairs of ROIs. Results will be stored as nifti images (whole brain) and text/mat file (ROI to ROI) in `results\`. Furthermore, a Matlab struct containing all information about the firstlevel will be stored in `firstlevel_info.mat`.

4.4 Example

Matlab example 4:

```
afxConn(func,masks,rp,denoisingOptions,rois,firstlevelDir,conditionName,subjectNam
e, analyses)
% func              preproc functional data: - func.func (smoothed)
%                                            - func.func2 (unsmoothed, optionally)
% masks             individual gm, wm, and csf masks (optionally a lesion mask)
% rp                realignment parameter file
% denoisingOptions  see above (example 2)
% rois              see above (example 3)
% firstlevelDir     firstlevel directory relative to results\
% conditionName     name of condition
% subjectName       name of subject
% analyses          cell array of analyses methods, currently implemented:
                                                    - fc_wholebrain
                                                    - fc_network
                                                    - vmhc
                                                    - hemo_lag
                                                    - fALFF
afxFirstlevel(subjects,denoisingOptions,rois,projectName[,analyses])
% subjects         ... array of structs as returned by afxPreproc, see example 1
% denoisingOptions ... see example 2
% rois             ... see example 3
% projectName      ... firstlevel directory relative to results\
% analyses         ... see above, afxConn()
% note that subjects, denoisingOptions and rois can also be speciefied as
% filenames which contain the data, i.e. mat-files
```

5 Second level

Second level analysis is performed using SPM12. There is an interface script which automatically sets up and estimates (simple) second level evaluations as flexible factorial design and also some (basic) contrasts for whole brain connectivity maps. There is also a second level analysis for lesion network mapping, which calculates mean connectivity and a t-test representing the main effect across all conditions. The first level files will be deleted to save disk space. Finally, also afxStat (https://github.com/afx1337/afxStat) can be used for permutation testing.

Matlab example 5:

```
afxSecondlevel(firstlevelInfo,covariateFactors,groupFactor)
% firstlevelInfo    info file which contains all information about the firstlevel
%                   (generated by afxGroupStudy)
% covariateFactors  fields of info which shall be used as covariates
% groupFactor       field of 'info' which shall be used as grouping factor
```

6 Graphical user interface

All steps including preprocessing, first level analysis and second level analyses can alternatively be performed using the Matlab app 'afxRs.mlapp' (Fig 2). Lesion network mapping is also featured here. For lesion network symptom mapping, please refer to the afxStat toolbox (https://github.com/afx1337/afxStat). Results can be viewed, thresholded, resliced and saved with 'afxViewer.mlapp' (Fig 3). The viewer also features results tables and anatomical labeling with various atlases.

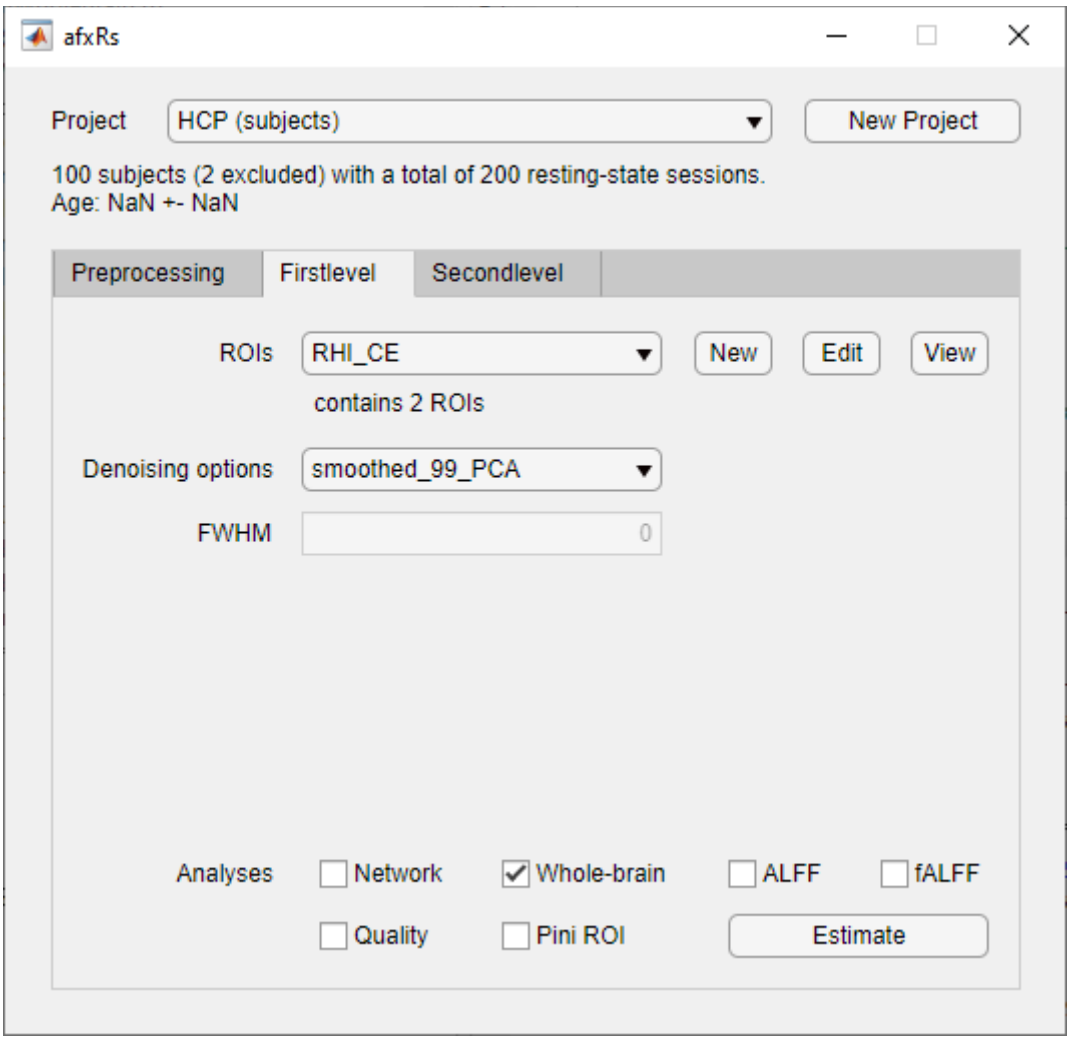

Figure 2. Graphical user interface.

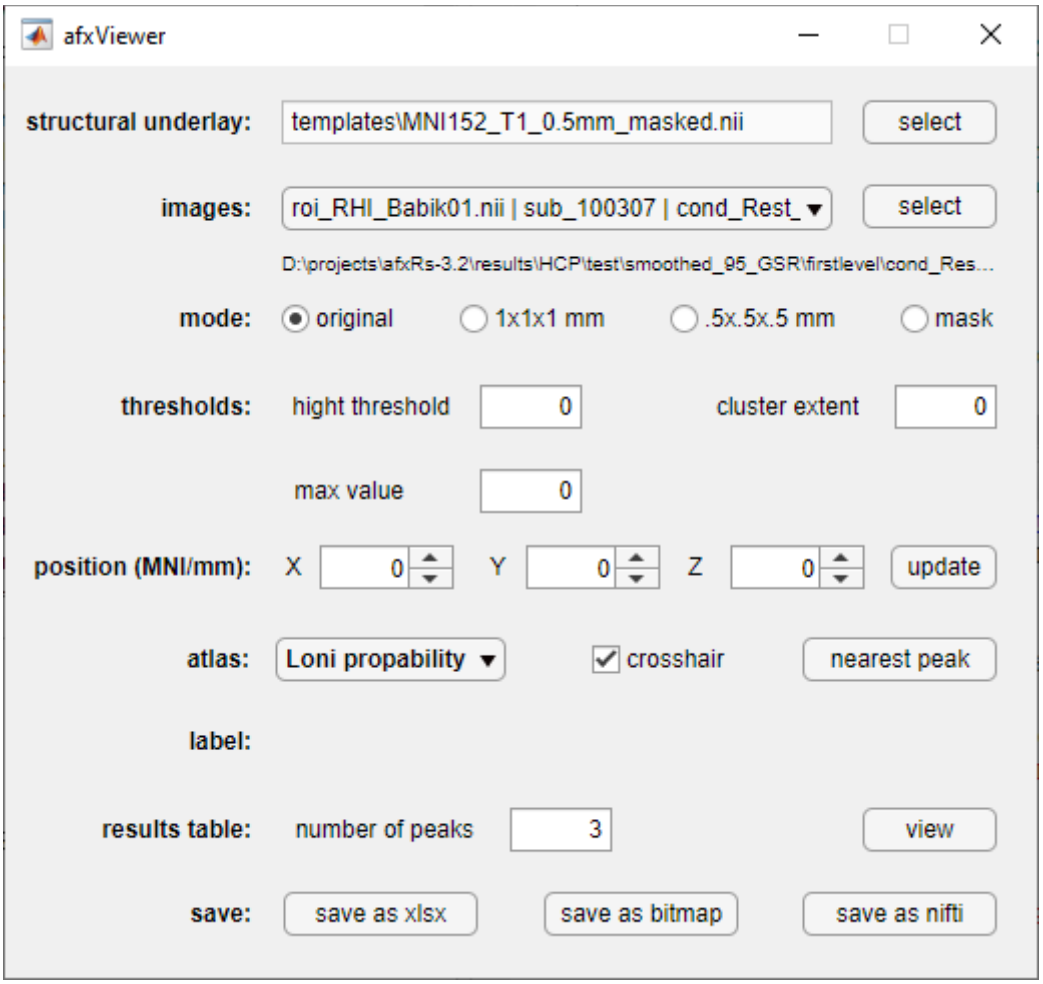

Figure 3. afxViewer interface.