Pre-processing of Resting State fMRI (rsfMRI)

Akrem Sellami, and Sylvain Takerkart

January 2019

1 Proposed Pipeline

The main goal of the proposed pipeline is to process resting state fMRI data, in order to extract a set of features vectors, which can be represented as a correlation matrix between the voxels V_i and regions of interest (ROIs). The rsfMRI are available in the following directory:

/hpc/banco/sellami.a/InterTVA/rsfmri

These data contain 39 subjects: sub-03, ...sub-35, sub-37,...sub-42. For the sub-36 annotations files (which contain labels of all ROIs) not exist.
We used SPM software to process rsfMRI. All batch script are available in the following directory:

/hpc/banco/sellami.a/InterTVA/rsfmri/script_batch_rsfmri

The source code of the developed project is based on Python software. All files of the source code are available in:

/hpc/banco/sellami.a/Preprocessingrfmri/good_project

In fact, in order to execute it, there are two possibilities: interactive or no interactive mode of Frioul:

- Interactive mode: it aims to process all subjects in a sequential processing, i.e. subject per subject. In order to execute the program, the following syntax it should be used:
  python preprocessing_rsfmri sub-xx, e.g.: python preprocessing_rsfmri 3 4 5 6...42

- No Interactive mode: it seeks to process all subjects at the same time (parallel processing). Therefore, the following syntax can be used:

/hpc/soft/anaconda3/bin/python batch_frioul.py

The main goal of the proposed pipeline is to process all rsfMRI data, in order to compute the correlation matrix between all voxels and ROIs. Therefore, as input, we have a bold images for each subject S_i, and we apply a batch script on SPM using Matlab software to get a corrected image in Gifti format. Then, we compute the correlation matrix between voxels and ROIs. Therefore, Figure 2 reports different steps of the proposed pipeline for rsfMRI preprocessing. For this pipeline, we have used SPM on MATLAB and Python softwares. The

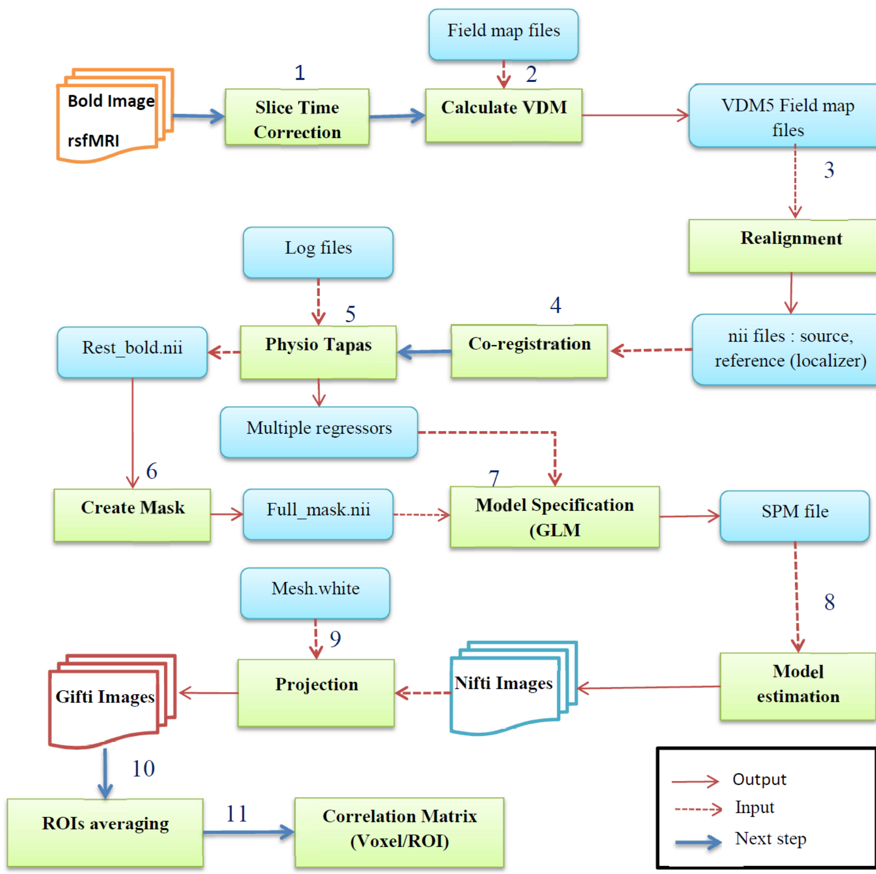

Figure 1: The proposed pipeline for resting state fMRI preprocessing

proposed batch script contains 8 steps: Slice time correction, calculate VDM, realignment, co-registration, Physio-tapas, model specification, and model estimation. For the rest of steps: create mask, projection, ROIs averaging, and correlation matrix, we used Python software. Figure, shows the organization

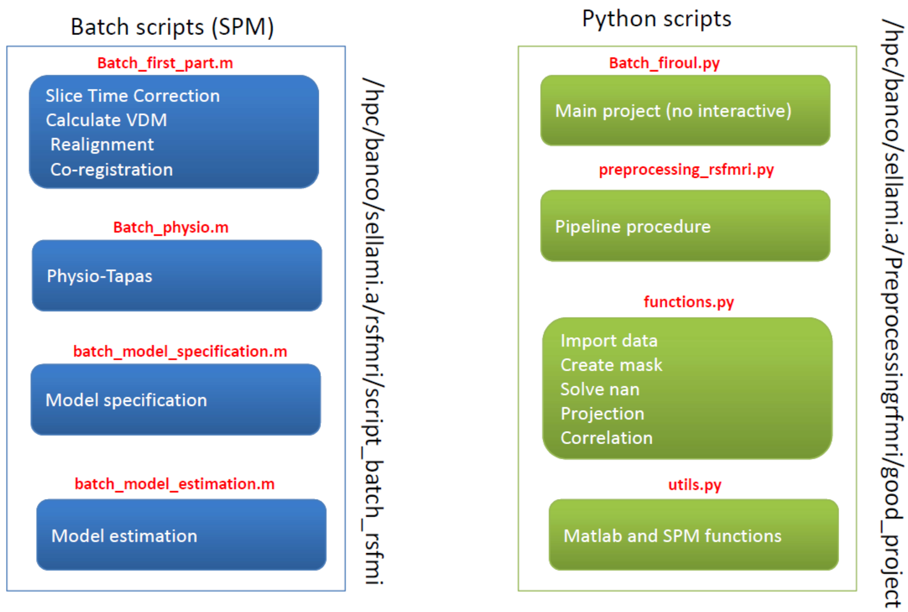

Figure 2: Description of source code files developed on SPM and Python softwares

Number of slice	60
TR	0.955
TA	0.939
Slice timing	0.875 0.7975 0.7175 0.6375 0.5575 0.4775 0.3975 0.32 0.24 0.16 0.08 0 ($\times$ 5)
Slice order	1 $\times$ 60

Table 1: Parameters settings of STC

2 Description of different steps of the proposed pipeline

2.1 Step 1: Slice Time Correction (STC)

Slice Time Correction (STC) generally improves our the statistical power of resting fMRI analyses. It aims to reduce the time gap between different slices. Table 1 shows the parameters settings for STC. To get an information about the echo times and time slicing, you can see the header files (*.json, *.txt).

2.2 Step 2: Realignment

The realignment is also known as 'motion correction'. The aim is to remove movement artifact in rsfMRI time-series. SPM does this by realigning a time-series of images acquired from the same subject using a least squares approach and a different parameters spatial transformation.

3 Step 3: Coregistration

A Process to overlay structural and functional images in a way that maximizes the mutual information. The mean functional image is coregistered to a high resolution anatomical image, and all of the other functional images are then resliced to align with the reference image.

3.1 Step 4: Physio-Tapas Toolbox

The general purpose of this Matlab toolbox is the model-based physiological noise correction of resting fMRI data using peripheral measures of respiration and cardiac pulsation.

3.2 Model specification (General linear Model)

General linear models (GLMs) are the most often used class of supervised methods for finding SPM of neural activation. For a given data set the target variable $y \in R^L$ (e.g. a single voxel time course of length L) is modeled as a linear combination of N given regressor time series x_i, each weighted by a coefficient stored in a vector $\beta \in R^N$, plus some gaussian, i.e., error $\epsilon \simeq (0, 1)$

$$\hat{y} = X\beta + \epsilon \tag{1}$$

where the $L \times N$ matrix $X = [x_1 x_2 ... x_N]$ contains the N regressors x_i of length L as column vectors and is called design matrix. A typical GLM analysis includes as regressors all experimentally controlled parameters and additionally so called nuisance regressors, which are not of interest in the analysis but explain considerable amounts of variance in the data.

3.3 Projection of nii files into gii files

In this step we use the following code to produce the gii files in LH and RH spaces: Example for sub-40 :

```
$FREESURFER_HOME/bin/mri_vol2surf --src {} --o {}_rh.gii --out_type gii --regheader sub-40 --hemi /rh --projfrac-avg 0 1
0.1 --surf-fwhm 0 --sd /hpc/banco/cagna.b/my_intertva/surf/data/sub-40/fs --trgsubject sub-40 \;

$FREESURFER_HOME/bin/mri_vol2surf --src {} --o {}_lh.gii --out_type gii --regheader sub-40 --hemi /lh --projfrac-avg 0 1
0.1 --surf-fwhm 0 --sd /hpc/banco/cagna.b/my_intertva/surf/data/sub-40/fs --trgsubject sub-40 \;
```

In order to visualize the obtained gifti files, we can use Anatomist software to combine the mesh lh.white or rh.white with the gifti file ($ctrl + f$). However, it's necessary to convert the mesh of surfer format to gifti format as follows :

```
freesurfer_setup

mris_convert /hpc/banco/cagna.b/my_intertva/surf/data/sub-40/fs/sub-40/surf/lh.white
/hpc/banco/sellami.a/InterTVA/rsfmri/sub-40/glm/noisefiltering/lh.white.gii

mris_convert /hpc/banco/cagna.b/my_intertva/surf/data/sub-40/fs/sub-40/surf/rh.white
/hpc/banco/sellami.a/InterTVA/rsfmri/sub-40/glm/noisefiltering/rh.white.gii
```

3.4 Correlation Matrix (V_i, ROI)

This step aims to compute the correlation matrix between two time series of voxels and ROIs. In fact, we have a set of ROIs obtained from annotations files, and a set of gii files extracted by the projection operation. Therefore, we compute the average matrix of All ROIS, and the correlation between the voxel and the mean of each ROI using a correlation coefficient such as Pearson coefficient or mutual information criterion.

3.5 Annotations files

ID ROI	Name	ID ROI	Name
-1	Unknown	40	Lat_Fis-ant-Vertical
1	G&S_frontomargin	41	Lat_Fis-post
2	G&S_occipital_inf	43	Medial_wall
3	G&S_paracentral	44	Pole_occipital
4	G&S_subcentral	45	Pole_temporal
5	G&S_transv_frontopol	46	S_calcarine
6	G&S_cingul-Ant	47	S_central
7	G&S_cingul-Mid-Ant	48	S_cingul-Marginalis
8	G&S_cingul-Mid-Post	49	S_circular_insula_ant
9	G_cingul-Post-dorsal	50	S_circular_insula$_inf$
10	G_cingul-Post-ventral	51	S_circular_insula$_sup$
11	G_cuneus	52	S_collat_transv$_ant$
12	G_front_inf-Opercular	53	S_collat_transv$_post$
13	G_front_inf-Orbital	54	S_front_inf
14	G_front_inf-Triangul	55	S_front_middle
15	G_front_middle	56	S_front_sup
16	G_front_sup	57	S_interm_prim-Jensen
17	G_InS_lg&S_cent$_ins$	58	S_intrapariet&P_trans
18	G_insular_short	59	S_oc_middle&Lunatus
19	G_occipital_middle	60	S_oc_sup&transversal
20	G_occipital_sup	61	S_occipital_ant
21	G_oc-temp_lat-fusifor	62	S_oc-temp_lat
22	G_oc-temp_med-Lingual	63	S_oc-temp_med&Lingual
23	G_oc-temp_med-Parahip	64	S_orbital_lateral
24	G_orbital	65	S_orbital_med-olfact
25	G_pariet_inf-Angular	66	S_orbital-H$_Shaped$
26	G_pariet_inf-Supramar	67	S_parieto_occipital
27	G_parietal_sup	68	S_pericallosal
28	G_postcentral	69	S_postcentral
29	G_precentral	70	S_precentral-inf-part
30	G_precuneus	71	S_precentral-sup-part
31	G_rectus	72	S_suborbital
32	G_subcallosal	73	S_subparietal
33	G_temp_sup-G_T_transv	74	S_temporal_inf
34	G_temp_sup-Lateral	75	S_temporal_sup
35	G_temp_sup-Plan_polar	76	S_temporal_transverse
36	G_temp_sup-Plan_tempo		
37	G_temporal_inf		
38	G_temporal_middle		
39	Lat_Fis-ant-Horizont		

Table 2: List of Id and names of all ROIs