

SPM12 fMRI and EEG Analysis Documentation

NMDA SS 2023

May 22, 2023

Contents

1	Introduction	2
2	fMRI Analysis	2
2.1	Preprocessing	2
2.1.1	Realignment	3
2.1.2	Slice Time Correction	5
2.1.3	Coregistration and Normalization	5
2.1.4	Smoothing	7
2.2	First Level General linear Model	9
2.2.1	Specification of Designmatrix	10
2.2.2	fMRI model estimation	12
2.2.3	Producing contrast images and SPMs	13

1 Introduction

The following document contains important aspects of the processing and analysing steps of fMRI and EEG data using the statistical analysis pipeline SPM12 based on matlab functions. The first part of this documentation will describe the steps of preprocessing and analyzing fMRI data. EEG data will be covered in the second part.

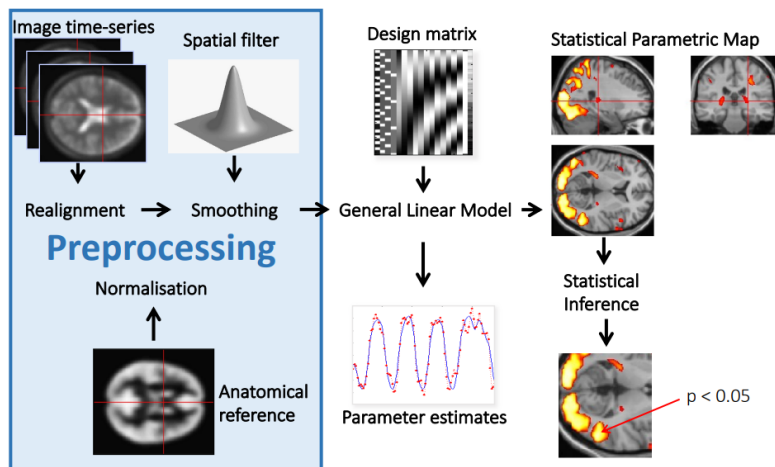
2 fMRI Analysis

2.1 Preprocessing

Preprocessing of fMRI images prior to statistical analysis allows to ensure that the data has sufficient quality and reduces the influence of disruptive factors to ensure reliable results.

Preprocessing of fMRI data consists of four different steps:

1. Realignment
2. Slice Time correction
3. Normalization
4. Smoothing

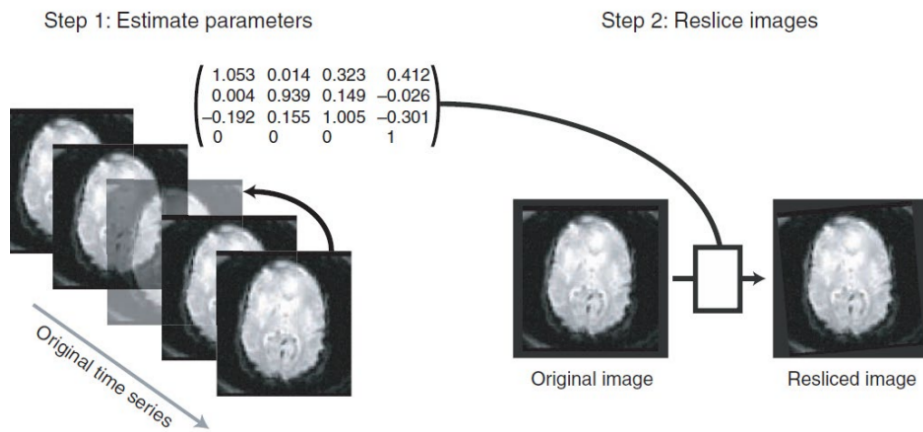


2.1.1 Realignment

Realignment in fMRI preprocessing is basically motion correction. Because it is not possible to completely avoid minimal motion in the scanner, some images of the time-series are tilted against each other due to motion artifacts. This will lead to a distortion in the recording of a particular neuronal structure.

In fMRI, especially task related motion is a huge problem. Example: If pushing a response-button induces head motion, a strong signal change will be artificially induced into the data. This signal change will be time-locked to the event of button-pressing and might overwrite any effect of interest.

To reduce the influence of these motion artifacts on statistical results, all fMRI data is realigned. For that, differences in brightness between the single images are used. As the shape of the head is not changing, an algorithm can determine optimal spinning/turning of the images to minimize these differences. The algorithmic calculations result in motion parameters, which are then used to interpolate and rewrite the images.



Important comments for the MatLab Code:

`nrun = x` → specifies how many times the realignment will run

`data_dir=strcat(data_dir, 'f*')` → specifies to only realign data files that start with f

`spm('defaults', 'FMRI')` → sets defaults modes for fMRI analysis in SPM12

Set SPM 12 Parameters

→ *matlabbatch1.spm.spatial.realign.estwrite.eoptions.[...] = x*, where x is one of the following arguments:

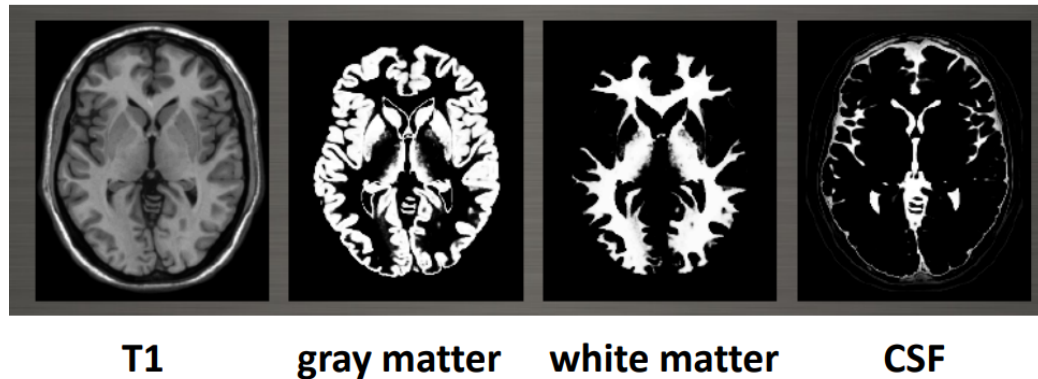
1. **quality** → Quality versus speed trade-off. Highest quality (1) gives most precise results, whereas lower qualities gives faster realignment
2. **fwhm** → The FWHM of the Gaussian smoothing kernel (mm) applied to the images before estimating the realignment parameters
3. **sep** → the default separation (mm) to sample the images
4. **rtm** → registers the images to the mean of the images after the first realignment
5. **wrap** → Directions in the volume whose values should wrap around in
6. **interp** → B-spline degree used for interpolation
7. **which** → values of 0, 1 or 2 are allowed [default: 2]
 0 - don't create any resliced images. Useful if you only want a mean resliced image.
 1 - don't reslice the first image. The first image is not actually moved, so it may not be necessary to resample it.
 2 - reslice all the images.
 If which is a 2-element vector, flags.mean will be set to flags.which(2).
8. **wrap** → three values of either 0 or 1, representing wrapping in each of the dimensions. For fMRI, [1 1 0] would be used. For PET, it would be [0 0 0]. [default: [0 0 0]]
9. **mask** → mask output images (true/false) [default: true]
 To avoid artifactual movement-related variance the realigned set of images can be internally masked, within the set (i.e. if any image has a zero value at a voxel than all images have zero values at that voxel). Zero values occur when regions 'outside' the image are moved 'inside' the image during realignment
10. **prefix** → prefix for resliced images [default: 'r']

2.1.2 Slice Time Correction

Preprocessing also includes slice time correction. FMRI volumes are typically acquired in a slice-wise manner and not all at once. Because of that, the time delay of image acquisition between the first and the last slice is almost $1 \times \text{TR}$. These slice-time acquisition differences can be compensated by temporal interpolation of the data (= Slice time correction).

2.1.3 Coregistration and Normalization

Normalization is based on the problem that every head (and brain) has a different shape. To allow the calculation of group statistics with spatial information, all individual brains need to be transformed to a standard space, for example the MNI space. For that the fMRI images need to be segmented into gray and white matter maps based on differences in bightness of different tissues in the T1 and probability maps.



After this step, so called Warping of the data is performed, where optimal transformation parameters are determined to map every individual brain to a standard-shape (= mapping onto the MNI standard template).

Coregistration: Important comments for the MatLab Code:

```
refimage_filename = strcat(data_dir_images, "meanfM00223-004.img,1") →  
get the name of the reference image file  
sourimage_filename = strcat(data_dir_field, "sM00223-002.img,1'") → get  
the name of the source image file
```

Set spm parameters

matlabbatch1.spm.spatial.coreg.estimate.ref = refimage_filename; → image that is assumed to remain stationary, moved image is matched to it
matlabbatch1.spm.spatial.coreg.estimate.source = sourimage_filename; → image that is jiggled around to best match the fixed image (reference image)
matlabbatch1.spm.spatial.coreg.estimate.other = '' → any images that need to remain in alignment with the moved image
matlabbatch1.spm.spatial.coreg.estimate.eoptions.cost_fun = 'nmi'; → sets NMI-brainspace as standard shape
matlabbatch1.spm.spatial.coreg.estimate.eoptions.sep = [42] → separation, mean distance between sample points, vector allows coarse registration followed by increasingly fine ones
matlabbatch1.spm.spatial.coreg.estimate.eoptions.tol = [...] → tolerance, the accuracy for each parameter, iterations stop when difference between successive estimates is less than required tolerance

Normalization: Important comments for the MatLab Code:

data_dir_images=strcat(data_dir_images, 'r')* → specify to only normalise data files that have been realigned
file_list_images=dir(data_dir_images); → get a structure of image files in directory
data_dir_field=strcat(data_dir_field, 'y');* → this will find the field file named according to the previous steps from manual
file_list_field=dir(data_dir_field); → get a structure with the field file

Set spm parameters

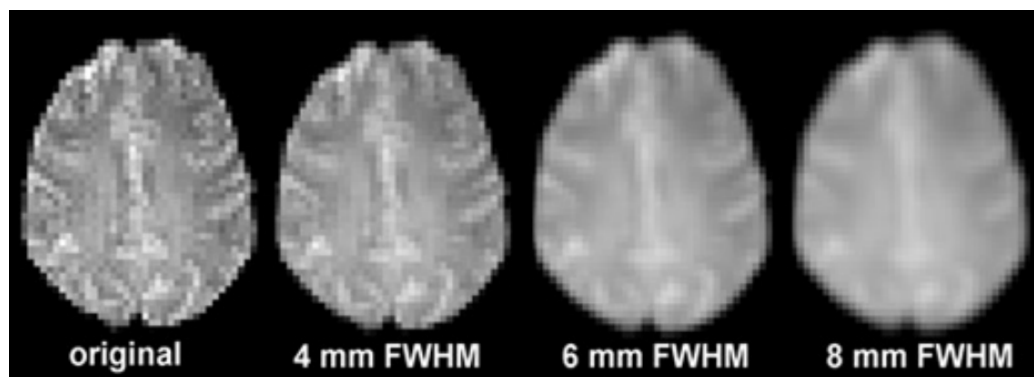
matlabbatch1.spm.spatial.normalise.write.subj.def = file_name_field1; → define the deformation field
matlabbatch1.spm.spatial.normalise.write.subj.resample = file_names_images; → define images to write, i. e. images for warping around to the estimated parameters
matlabbatch1.spm.spatial.normalise.write.woptions.bb = [-78-112-70787685]; → create the bounding box (in mm) of the volume which is to be written (relative to anterior commissure)
matlabbatch1.spm.spatial.normalise.write.woptions.vox = [333]; → voxel size (x,y and z direction) of the written normalised image in mm
matlabbatch1.spm.spatial.normalise.write.woptions.interp = 4; → method

of sampling during rewriting in different space: Nearest neighbour (not recommended), trilinear Interpolation (okay for realigned fMRI), B-spline interpolation (better quality but slower)

matlabbatch1.spm.spatial.normalise.write.woptions.prefix = 'w'; → adds prefix to file, default is w

2.1.4 Smoothing

But even after motion correction, normalization and transforming each brain into MNI-space, inter-individual spatial differences remain. To resolve these and to allow reasonable group-statistics, the data is smoothed using Gaussian Filter Windows. Smoothing eliminates small outliers, pronounces main effects that are present in multiple neighbouring voxels and affects the significance threshold of consecutive testing.



Important comments for the MatLab Code:

data_dir_norm = strcat(data_dir, 'w')* → specify to only use normalised functional images that start with w

file_list_norm = dir(data_dir_norm) → get a structure of image files in directory

Set SPM parameters

matlabbatch1.spm.spatial.smooth.data = file_names_norm; → Specify the file names of the normalized images to be smoothed

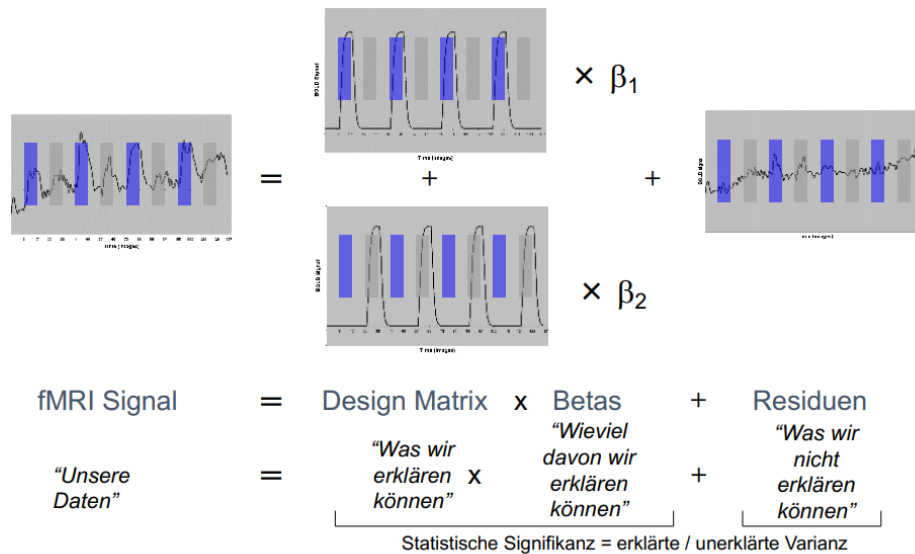
matlabbatch1.spm.spatial.smooth.fwhm = [888]; → specify the full width half maximum of smoothing kernel in mm in x,y and z direction
matlabbatch1.spm.spatial.smooth.dtype = 0; → data type of output images, 0 indicates same data type as original image
matlabbatch1.spm.spatial.smooth.im = 0; → adds implicit masking (mask implied by particulare voxel value) where 0 values are excluded
matlabbatch1.spm.spatial.smooth.prefix = ' s' → adds prefix to data files, default for smoothing is s

2.2 First Level General linear Model

The statistical analysis of fMRI data is usually split into two different levels:

1. First Level or subject level
2. Second level or Group level

The first or subject level focuses on the data of each participant individually. The analysis is based on the GLM which consists of the data, the regressors, one beta weight for each regressors and errors. The regressors are the variables we can explain, beta weights indicate how much each regressor explains variance between experimental condition and the error describes how much of the variance we cannot explain.

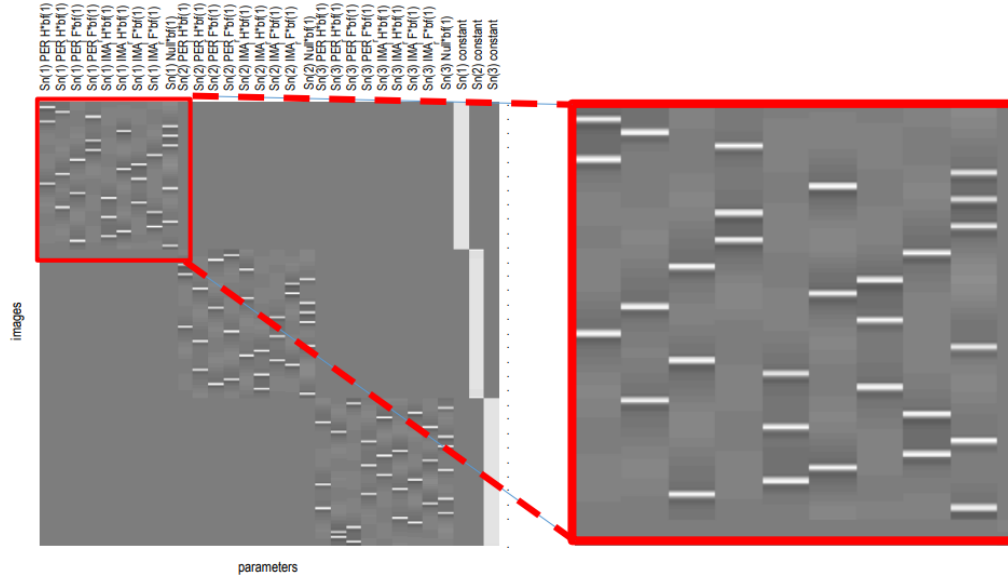


In the analysis each regressors is convolved with an HRF and a GLM is constructed that estimates beta-weights for each participant over each regressor. The beta weights are then taken to the second level analysis as observations. In SPM, the statistical analysis of fMRI data comprises three different steps:

1. Specification of the GLM design matrix, fMRI data files and filtering
2. estimation of GLM parameters using classical or bayesian approaches
3. interrogation of results using contrast vectors to produce statistical parametric maps (SPMs)

2.2.1 Specification of Designmatrix

Specification of the GLM Matrix consists of multiple important inputs. Timing parameters identify the onsets of events or blocks, interstimulus intervals and microtime resolution of the scans. For subject/session parameters, SPM requires input regarding scans, conditions and their names, onsets and durations of conditions types and names and values of the regressors. If you have a factorial design, additional inputs can allow SPM12 to generate the contrasts necessary to test for main effects and interactions.



Important comments for the MatLab Code:

Timing parameters

`matlabbatch1.spm.stats.fmri_spec.timing.units = 'scans';` → onset of events or blocks can be specified in either scans or seconds

`matlabbatch1.spm.stats.fmri_spec.timing.RT = 7;` → interscan interval, time between acquiring a plane of one volume and the same plane in the next volume

`matlabbatch1.spm.stats.fmri_spec.timing.fmri_t = 16;` → microtime resolution, number of time-bins per scan

`matlabbatch1.spm.stats.fmri_spec.timing.fmri_0 = 8;` → microtime onset, first time-bin at which the regressors are resampled to coincidence with data

acquisition

Data and Design

matlabbatch1.spm.stats.fmri_spec.ssess.scans = cellstr(files); → select all fMRI-scans for this session, all need to have same image dimensions, orientations and voxel size, etc.

matlabbatch1.spm.stats.fmri_spec.ssess.cond.name = 'listening'; → name of the experimental condition of interest

matlabbatch1.spm.stats.fmri_spec.ssess.cond.onset = [] → vector of onset times of condition type, usually takes place using variables that have been created before and loaded into MatLab

matlabbatch1.spm.stats.fmri_spec.ssess.cond.duration = 6; → specifies event duration, events are specified with a duration of 0. If a single number is entered, it will be assumed that all trials conform to this duration.

matlabbatch1.spm.stats.fmri_spec.ssess.cond.tmod = 0; → allows characterization of nonstationary responses

matlabbatch1.spm.stats.fmri_spec.ssess.cond.pmod = struct('name', 'param', 'poly',); → parametric variates that modulate stick function, modeling the interaction between trial and variate

matlabbatch1.spm.stats.fmri_spec.ssess.cond.orth = 1; → ?

matlabbatch1.spm.stats.fmri_spec.ssess.multi = ''; → option can be used to load data from multiple different conditions in one go. Necessary to create .mat file containing names, onset and durations of different conditions and select it using the SMP file selector

matlabbatch1.spm.stats.fmri_spec.ssess.regress = struct('name', 'val',); → additional columns included in the design matrix that may model effects and would not be convolved with haemodynamic response functions (e.g., movement parameters)

matlabbatch1.spm.stats.fmri_spec.ssess.hpf = 128; → high pass filter cut-off, default at 128 seconds, slow signal drifts with a period longer than hpf will be removed

Factorial design

matlabbatch1.spm.stats.fmri_spec.fact = struct('name', 'levels',); → add a new factor to your experimental design, name of factor and number of levels necessary to enter

Base Functions

matlabbatch1.spm.stats.fmri_spec.bases.hrf.derivs = [00]; → model HRF derivatives, canonical HRF combined with time and dispersion derivatives to comprise an informed basis set, allows variation in subject-to-subject and voxel-to-voxel responses. Time allows variance of the peak response by plus or minus seconds and dispersion allows the width of response to vary by similar amount.

matlabbatch1.spm.stats.fmri_spec.volt = 1; → Model interactions (Volterra), generalized convolution of inputs, U with basis set bf.

matlabbatch1.spm.stats.fmri_spec.global = 'None'; → Global Normalization, **None** leads to computation of grand mean value (g) for one session s and then multiplying each fMRI data point in s with 100/g. **Scaling** leads to multiplication of each fMRI value in scan n and session s by 100/g.

matlabbatch1.spm.stats.fmri_spec.mthresh = 0.8; → ?

matlabbatch1.spm.stats.fmri_spec.mask = '';m→ explicit mask, specifies an image for explicit masking the analysis. Optimal to use segmentation of structural images to specify a within brain-mask. If not available, leave field empty

matlabbatch1.spm.stats.fmri_spec.cvi = 'AR(1)'; → accounts for **serial correlations** due to aliased biorhythms and unmodelled neuronal activity using an autoregressive model. Can be ignored if NONE is chosen.

2.2.2 fMRI model estimation

Model parameters can be estimated using classical or Bayesian algorithms. For the classical algorithm, parameters are estimated using Restricted Maximizing Likelihood, which assumes that error correlation structure is the same over voxels. It should only be applied to spatially smoothed functional images.

In Bayesian 1st-level estimation, parameters are estimated using Variational Bayes that allows to specify spatial priors for regression coefficients. It does not require functional images to be spatially smoothed. Because it takes longer than the classical approach, it is not set as the SPM12 default mode. The Analysis Space-command allows to only analyse selected slices or clusters rather than whole volumes.

Important comments for the MatLab Code:

matlabbatch2.spm.stats.fmri_est.spmmat = fullfile(glm_dir,'SPM.mat');
→ select SPM file that contains the design specifications
matlabbatch2.spm.stats.fmri_est.write_residuals = 0; → ?
matlabbatch2.spm.stats.fmri_est.method.Classical = 1; → model parameters are estimated using Restricted Maximum likelihood, applied in this case onto the first level of analysis.

2.2.3 Producing contrast images and SPMs

This step is usually part of the second level of analysis. Only in the case of for example only one participant it can be performed on the participant level as well.

For that, the contrast images from the second step can be converted into statistical parametric maps (e.g., t-statistic maps) using a specific formula that divides the contrast image by its standard error. SPM usually does this automatically on the second level in addition to contrast images.