fMRI Analysis Guide

MetaLab

March 17, 2017

1.1 Introduction

This guide is designed to step through different practical steps in analysing fMRI data, primarily using SPM 12 in conjunction with custom-written scripts available at `https://github.com/smfleming/MetaLabCore`. It assumes some basic familiarity with MATLAB. It's not exhaustive by any means, but should provide a starting point for more advanced analyses, and as the analysis techniques regularly used in the lab develop over time we'll update this document accordingly.

If you have any questions or comments, or want to contribute sections on other analyses (e.g. connectivity) please get in touch at stephen.fleming@ucl.ac.uk.

1.2 Installing software

1.2.1 SPM

Go to `www.fil.ion.ucl.ac.uk/spm/software` and download the latest version of SPM12 as a .zip file (make sure you're using the same version throughout a given analysis). Unzip into a convenient location and note the path to the folder. Start MATLAB and type:

```
addpath('~/your/spm/directory')
spm
```

If all is well you should see the SPM menu.

1.2.1.1 Setting SPM defaults

We want to change a couple of the SPM defaults to ensure that the scripts run smoothly. If you're using SPM12, create a new file called spm_my_defaults.m and save it in the SPM directory. In this file put the following lines:

```
global defaults
defaults.mask.thresh = -Inf;
```

Altering the masking threshold avoids areas of the brain being masked out automatically in first-level analyses, with possible consequences for group analyses (see `https://tinyurl.com/lo4p8o5` and `https://tinyurl.com/mg298j3`). For earlier versions of SPM you need to edit the spm_defaults.m directly instead; make sure you keep track of what has been changed.

To make things run a little faster we can also adjust the memory settings (this assumes you have at least 4GB of RAM):

```
defaults.stats.maxmem = 2^32;  % 4 GB
defaults.stats.resmem = true;
```

1.2.2 FSL

We sometimes use FSL for visualising data and other useful functions. Go to `http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FslInstallation` and follow the relevant instructions.

1.2.3 NYU CBI Data Quality tool

We use the data quality tools from NYU's CBI to check the functional runs for anomalies such as spikes. This toolbox can be downloaded from `http://cbi.nyu.edu/software/dataQuality.php` and should be placed in the MATLAB path.

1.3 Organising data

It's important to organize the data directories so each script can find the right folders. First create a new top-level project directory which we refer to below as dir_base e.g. /Documents/Data/MyProject. In the preprocessing scripts this path will be passed to each function as the variable 'path'. Inside this project directory make a new directory for each subject e.g. sub01, sub02, sub03, etc. How you name subjects is up to you but make sure it's consistent and you can link it up to the relevant behavioural datafiles. Inside each subject directory, create a folder 'raw'. This is the folder which will contain raw data downloaded from the scanner.

1.3.1 Importing data from NYU scanners

To access your data at NYU, first login to `cbi.nyu.edu` and navigate to the scheduler. Select the scan slot, select all scans and choose 'Save'. Enter a folder name for your data (e.g. yourname/sub01) and ensure Nifti single file output is selected. Click 'OK'. You will receive an email once the data are exported to the server.

To download the data to your local machine, connect to Tesla at the following address: `afp://tesla.cbi.fas.nyu.edu`. On a Mac you can do this inside the Finder menu by clicking Go and 'Connect to server'. Navigate to dawlab and find your scans for the relevant subject. Each run of the scanner is listed in a separate folder. For instance, here are some folder names for one subject from a recent project:

```
1 sub01/raw/03+Siemens-3x3x3-wholeBrain
2 sub01/raw/04+Siemens-3x3x3-wholeBrain
3 sub01/raw/05+Siemens-3x3x3-wholeBrain
4 sub01/raw/06+MotionLocalizer
5 sub01/raw/07+grefieldmapping
6 sub01/raw/08+grefieldmapping
7 sub01/raw/09+t1mprage
```

The first 4 folders contain functional data; the next two contain the fieldmaps; the final folder contains the T1 scan. Copy all the relevant data folders (you can ignore the circle localizer) to your local subject directory.

1.3.2 Importing data from FIL scanners

Thank you to Dan Bang and the Physics Wiki (`http://intranet.fil.ion.ucl.ac.uk/wiki/physicswiki/doku.php`) for the following FIL-specific guide.

All data acquired on the Trio/Quattro/Penta scanners will be transferred overnight to the FIL data server, called Charm, which can be accessed from within the FIL. You can usually find a shortcut to Charm on your desktop. Once you double click on this shortcut, there will be a folder for each scanner. Select the scanner from which you need to retrieve data and select the folder corresponding to the current year. In each scanner's respective folder data is stored according to date and scan ID (e.g. 20160523.MT04498_FIL). All data from the Trio/Quattro/Penta is archived on Charm in compressed format as *.tar files. Images are packed into a single file for each run which has the extension '.tar'. The Import Archive tool will allow you to import and dicom convert the archived data files in one step. The tool, which works with SPM5, SPM8 and SPM12, can be downloaded and used as described in the following steps:

- Open the File Explorer (e.g., by click on the Charm shortcut)
- Type `svn://bread.fil.ion.ucl.ac.uk/tools/Import_Archive` into the path; this will open the FIL Repository Browser
- On the left-hand side, identify Import_Archive, right-click and choose checkout, then checkout the toolbox into a subfolder of your SPM toolbox folder
- To run the tool, open Matlab, type SPM, select modality (e.g., fMRI), type Import_Archive and select the *.tar files to convert (note that each run of the scanner is listed in a separate folder)
- Select an Output directory (must be created beforehand) if it should be different from the current working directory
- The files for each run will be dicom-converted and written to a directory which is named MTXXXX.runnum (where XXXX is the subject ID and runnum is the run number)

1.4 Quality checks and data import

1.4.1 Initial quality check

Call the data_quality function by passing in dir_base, a cell array of the name(s) of the subjects you wish to check, and a cell array of directories corresponding to the raw functional scans. For instance:

```
data_quality('path', '\Users\sfleming\Documents\Data\
    MyProject', 'subjects', {'sub01','sub02','sub03'}, 'data'
    , {'03+Siemens-3x3x3-wholeBrain','04+Siemens-3x3x3-
    wholeBrain','05+Siemens-3x3x3-wholeBrain','06+
    MotionLocalizer'})
```

This will apply custom quality checks for data spikes (artifacts due to non-physiological noise). Check the command window; it should say 'No spike candidates found' for each run. If any spikes are flagged you will need to visualise that particular scan for tell-tale features of a spike (e.g. stripes in the image); often spike candidates turn out to be false alarms.

A highly recommended step is to view a movie of raw data runs using FSL. Note that scanner data at the FIL is saved as 3D data, so you would need to convert it into 4D to use movie mode in fslview, for example. This can be done by: opening SPM, select Batch, click SPM, click Util, select 3D to 4D file conversion, click on 3D Volumes and find the set of 3D files that you'd like to convert to 4D. This will generate a 4D file as specified by Output Filename. Note that you can do several batches at once by repeating the above procedure. You can also save the script for running the batches for future reference, or hacking to your needs.

Start FSLView by opening the Terminal and typing

```
fslview &
```

The & means that the program you asked for (fslview) runs in the background in the terminal (or shell), and you can keep typing other commands while it is running. If you hadn't done that then you would not be able to do anything else in the terminal until you killed fslview (or, alternatively, you could type control-z in the terminal and then bg to get fslview running in the background post-hoc). In fslview go to Open to select your 4D file for each scanning run, and click on the movie button to cycle through the images. Any spikes or sudden movements should be obvious as discontinuities in the image series.

1.4.2 Data import

To unzip the data, you can call fil_mri_unzip(filename,subject) where filename is the name of the tar-file that you retrieved from Charm and subject (e.g. subject number

6) specifies where the unzipped data will be stored (in this example, a folder called s6). You must specify the following inside the script: a) a directory containing the Import Archive Tool b) the directory containing the tar-file source data and c) the directory which stores the unzipped data. Note that only one subject can be unzipped at a time.

To organise the data into appropriate folders, you can call fil_mri_organise(which_subjects) where which_subjects specifies which subject folders should be organised (e.g. [1:6]). You must specify inside the script: a) directory contained the (unsorted) unzipped data, b) folder names for sorting (e.g. 'functional','fieldmap'), and c) number of dummy volumes per functional scan (deleted automatically during sorting). The script has some dependencies: getDropbox, which we use because the first part of the Dropbox path is different on Macs and PCs, and fil_subject_details, which specifies which scan run is which (e.g. your experiment might have 10 runs where 1 is the localizer, 2 is the structural, 3 to 7 are functional scans and so forth). Note: there has to be correspondence between folder names used for sorting (see above) and those specified inside fil_subject_details, and fil_subject_details must be executed before running fil_mri_organise. Finally, the script also automatically copies the fieldmaps such that there is a set for each functional scan (needed for fieldmap processing).

1.4.3 Set the origin to the anterior commissure

In order to ensure that subsequent normalisation routines work as they should, it helps to get the brains roughly into MNI coordinate space. To do this we're going to reorient each subject's high-resolution structural image, and apply the same transformation to ALL the scans (functional, structural and fieldmaps) to ensure that they all move together. To do this, load up SPM and select Display. Select the subject's structural image. Click around until you find the anterior commissure (see `http://imaging.mrc-cbu.cam.ac.uk/imaging/FindingCommissures`).

To set the origin to the AC you need to "undo" this transformation. For instance, if the coordinates of the AC are [-5 20 -3] then you would need to type [5 -20 3] into the x, y, and z boxes of the reorientation function. It might also help to rotate the brain along its X-axis (to more approximate the typical MNI template orientation). Once you're happy with the transformation, click "Reorient images..." and then select both the current structural image (this should already be selected for you) and all the functional scans (f*.imgs) in each session and fieldmaps.

Hit "done" to apply the reorientation to all the subject's images.

1.5 Preprocessing

All the preprocessing steps are handled by the custom scripts, although you should still understand what each step does :-)

1.5.1 Fieldmaps

To undistort EPI sequences using fieldmaps, use fil_fieldmap_preprocess(which_subjects) where which_subjects specifies whose data should be processed (e.g. [1:6]). You must specify inside the script: spm directory; directory containing the organized subject folders; scan parameters (e.g. number of functional scans and fieldmap settings).

This script calls SPM's fieldmap_preprocess routine for multi-subject, multi-session datasets. fieldmap_preprocess creates a single fieldmap file (fpm5_*), converts it to a voxel displacement file file (vdm5_*), and matches the vdm5 file to the first EPI image from each run. For multiple sessions, copies of original fieldmaps should be present in each of dir_fieldmap/session* directories. This allows each session to have individually matched voxel displacement maps.

See `http://intranet.fil.ion.ucl.ac.uk/wiki/physicswiki/doku.php?id=start:data_processing:using_field_maps` for further information.

1.5.2 EPI data

Preprocessing of functional (EPI) scans is handled by the script spatial_PreprocessingBatch.m. This function takes in string/value pairs to pass it information on the paths to the data, subjects to preprocess, and the preprocessing options you want. The main steps are a) slicetiming correction (optional, default is not to slice-time correct), b) realign and unwarp using the fieldmaps created in the previous step, c) coregistration of the structural image to the mean functional, d) segmentation and normalisation of the structural using SPM's Unified Segmentation routine, e) application of the warps generated in (d) to the EPI scans and e) smoothing of the normalised EPI scans. A typical call to spatialPreprocessingBatch would be as follows:

```
spatial_preprocessingBatch('dir_base', '/Users/sfleming/path/to/data/', 'dir_spm', '/Users/sfleming/path/to/spm/', 'subjects', {'sub1', 'sub2', 'sub3'}, 'n_sess', 4, 'TR', 2.34, 'nslices', nslices, 'sliceorder', [2:2:nslices 1:2:nslices-1])
```

where nslices = the number of slices in one functional volume.

1.5.3 Post-processing checks

After pre-processing has finished, it's now time to check that the data again to make sure nothing went wrong during the automated coregistration/normalisation routines, and to examine the realignment parameters for possible issues with scan to scan motion.

To check motion, use the script check_motion.m. Similar to data_quality.m this takes dir_base, subjects, and n_sess inputs and produces graphs of motion parameters and their derivatives, plus an initial assessment of whether the motion exceeds a pre-defined criterion. This criterion is rather arbitrary but a good starting point is whether the scan-to-scan motion exceeds 1/2 a voxel, which for most EPI sequences this is likely to be between 1 and 2mm (Poldrack, Mumford & Nichols, 2011). This script will write out a bad_scans.mat file to each subject's Functional directory which contains a 1xNscan*Nsess vector with 1's at each timepoint the motion exceeds the affine_cutoff parameter. This vector can be used to construct regressors to model out bad scans in the design matrix, assuming they are not too numerous. A similar and perhaps more straightforward approach is to include the first derivative of the 6 motion parameters as covariates in the design matrix as described below in Single subject model.

An alternative approach (which we haven't used much) is to use the ART Repair toolbox (`http://cibsr.stanford.edu/tools/human-brain-project/artrepair-software.html`), which will assess scan quality not only in terms of movement but also in terms of aberrant signal intensities (although the latter are usually correlated with sudden movements). This will also write out .mat files to the subject directory with regressors for modeling out bad timepoints.

To check normalisation, try using the check_warp script to visualise the normalised functional images for a group of subjects. All should look roughly (MNI) brain-shaped; gross warping errors will be clearly visible.

1.6 Single subject model

An example of scripting the creation and estimation of a first-level design matrix is found in the getonsetsModel.m script. An example of fitting a betaseries model (to obtain single-trial activity estimates) is found in the getonsets_singleTrial.m script. Once a design is estimated, the script 1stlevelModel.m is an example of how to code contrast estimates.

1.7 Group model

An example of scripting a second-level design matrix can be found in 2ndlevelModel.m.

1.7.1 Design efficiency

Excellent practical advice on design efficiency can be found on the CBU wiki, `http://imaging.mrc-cbu.cam.ac.uk/imaging/DesignEfficiency`.