fMRI Basics: Spatial Pre-processing Workshop

Starting a VNC session:

Most of your fMRI analysis will be done on the central Linux machines accessed via a VNC (Virtual Network Computing) server. This is a program which runs on a remote computer (in this case the Linux machines) and provides you with a virtual desktop on that machine.

Most desktop machines will have a program called PuTTY installed – you can use this to log on to the central computing cluster:

- Double click the PuTTY icon.
- In the dialogue box, type “electra” in the box labelled “Host Name (or IP Address)”.
- Most of the other default settings should be OK for the time being, so click the “Open” button at the bottom of the dialogue box.
- In the window that opens, enter your user name and network password.
- The Linux machines are called l03 (lower case L followed by 03), l04 etc upto l52
- Pick a Linux machine to log onto and type `ssh` followed by the name of the machine you’ve chosen, e.g. `ssh l16`
- If you’re asked about permanently accepting the host’s key, just type yes.
- If you’ve not already done so, you might want to change the window manager used by VNC. The easiest way to get a reasonable window manager is simply to type `use_icewm` – this will set the ice window manager to be the default (Gnome and KDE are also available)
- To start a VNC server, type `vncserver`. There are various other options you can specify to control the properties of the server, e.g. `vncserver –geometry 1200x960` will give you a 1200x960 desktop.
- When you start the VNC server, you’ll get a message telling you the name of your new desktop (e.g. l16:1). Make a note of this and then log out of your putty session.
- Back on the Windows desktop, look for a shortcut to a VNC viewer program. This is just a program which allows you to connect to the VNC desktop you’ve started on one of the Linux machines. It’ll probably be called something like TightVNC or RealVNC. If there’s no desktop shortcut, try – start→programs→networking.
- When you start the program, you’ll be asked for the address of the VNC server you want to connect to – type the name you were given when you started your desktop (e.g. l16:1). After that, you’ll be asked for your password, and if all goes well, you’ll be connected to your VNC desktop.

NB – Matlab is also installed on most desktop machines, and if you’re doing lots of graphic intensive work (rendering results etc) it might be more efficient to do this locally rather than via VNC (although you will need to install SPM on your desktop in order to do this).

For more details of the computing resources see:
http://imaging.mrc-cbu.cam.ac.uk/imaging/ImagingComputing

Directory Structure

Now that you've logged onto the system, its worth having a quick look around to familiarise yourself with the directory structure.

Your network home space will be /home/(*username)*
Your imaging space will be /imaging/(*username)*

The /imaging tree also contains lots of other useful stuff such as the SPM installation (/imaging/local/spm).

When raw data is transferred from the scanner, it ends up in /mridata/cbu. Within this directory, each subject has their own directory (don't try to `ls /mridata/cbu` – it could take a *long* time!)

The old format for each subject's directory name was
/mridata/cbu/(*subject-ID_subject-ID)*

e.g.
/mridata/cbu/CBU080705_CBU080705

The new format is
/mridata/cbu/(*subject-ID_project-ID)*

e.g.
/mridata/cbu/CBU090250_MR09002

Within each subject's directory, there's usually a single session directory, the name of which is derived from the time the data set was collected, e.g.

e.g.
/mridata/cbu/CBU080705_CBU080705/20080826_152343

Within each session directory, there will be a number of directories containing the data from each of the scans you collected. These will be in the format (*scan-ID_scan-name)*, e.g. inside /mridata/cbu/CBU080705_CBU080705/ 20080826_152343/ you can see several folders including:

./Series_001_CBU_Localiser
./Series_002_CBU_MPRAGE
./Series_003_CS_MEP5_16SL
./Series_006_CBU_FieldMapping

Some of the most frequently used scan types are:

CBU_Localiser = a quick pilot scan taken at the beginning of each session. Collects 1 axial, 1 saggital, and 1 coronal slice, and is used to plan the acquisition matrix for the structural scan.

CBU_MPRAGE – the high resolution T1 weighted structural scan, usually run second and used to plan the functional scans.

CBU_FieldMapping – a map of the magnetic field strength within the scanner

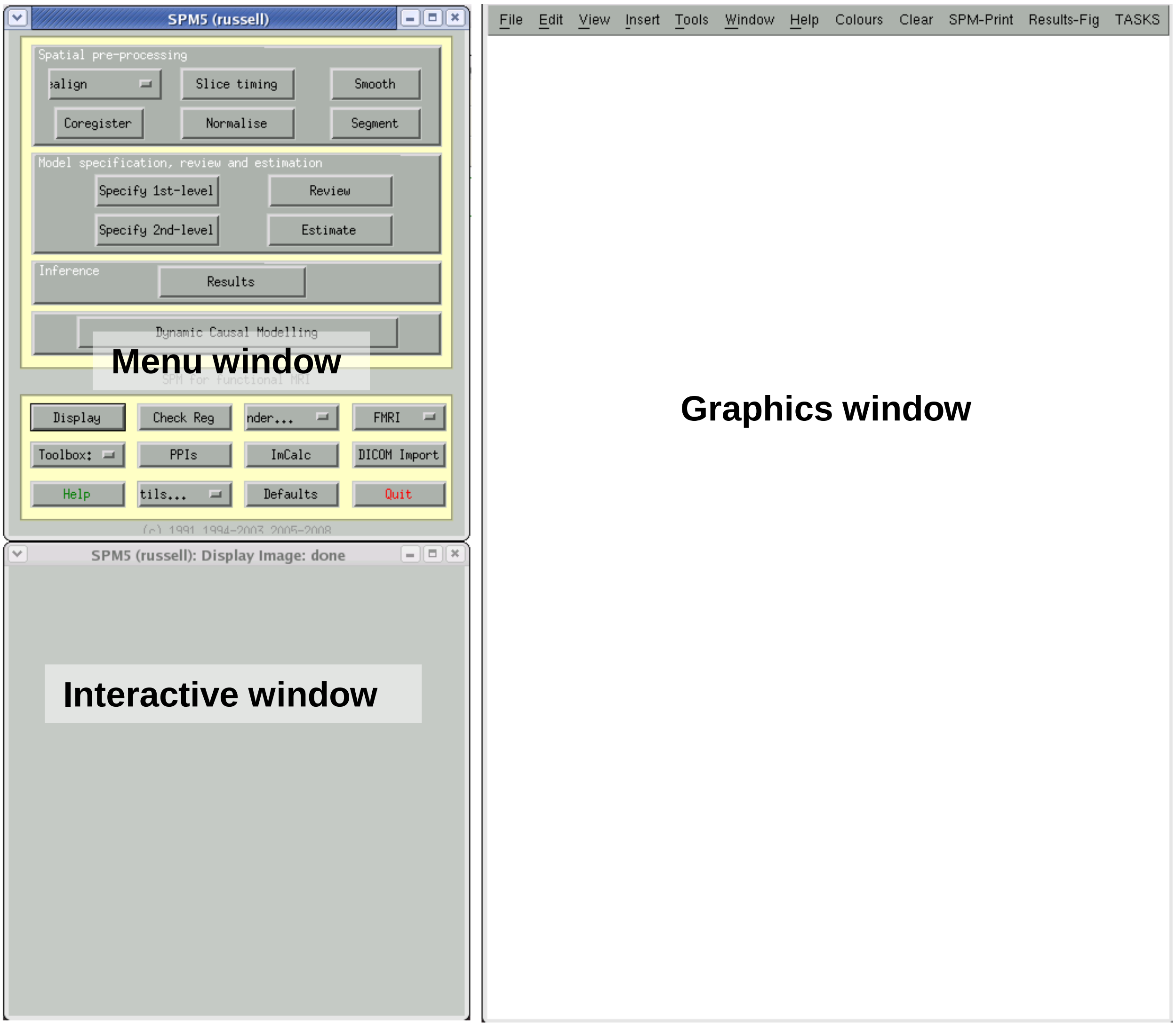
CBU_EPI – the default functional image scan, a T2* weighted gradient echo sequence.

Starting SPM

In a terminal window, type `spm 5`. This launches an SPM session on the processor which currently has the lowest load. (If you just type `spm` without any additional arguments you will start the default version of spm. Currently this is set to be SPM 8. The scripts for this workshop were written to work with SPM 5, and I've not tested them with the latest version.)

SPM will take anywhere from 10 seconds to a minute or so to load depending on how busy the machines are. Once you've started a session, you should see a Matlab command window and the 3 windows that make up the SPM gui (see figure below)

Most of the pre-processing steps described below can be run either through the SPM gui, or via batch scripts. When you start using SPM, you'll find that you hardly ever



use the gui for pre-processing or setting up statistical models, most of the time you'll rely on batch scripts.

If you don't know any Matlab, it's really worth learning at least the basics - it makes you much more independent, and allows you to do things that can't be done through the gui.

SPM is essentially a large collection of Matlab scripts, and you can integrate these into your own custom analyses as needed.

Pre Processing

All the pre-processing steps described below are included in a batch script you can find at:

/imaging/russell/MRI_workshop/pre_processing.m

If you've not already done so, make a directory in your imaging space for workshop related stuff. In either matlab or a terminal window type:

```
mkdir /imaging/(username)/MRI_workshop
```

Next, add this to your Matlab path by typing in the Matlab command window:

```
addpath /imaging/(username)/MRI_workshop
```

Then copy the script over to your local directory. In a terminal window, type:

```
cp /imaging/russell/MRI_workshop/pre_processing.m  /imaging/(username)/MRI_workshop
```

Once you've done this, open the script from the matlab command window by typing:

```
edit  pre_processing.m
```

The data set we're going to be working on comes from a localiser task where people saw blocks of face stimuli, blocks of scrambled face stimuli, and blocks of object stimuli. The ID of the subject we're going to be analysing is CBU071112, and the raw data are located in:

/mridata/cbu/CBU071112_CBU071112

The script is set up to create sub directories within your MRI_workshop folder as follows:

Main directory for this subject:
/imaging/*(username)*/MRI_workshop/CBU071112

Within that:
./Structurals – for the structural image
./Sess1, ./Sess2 and ./Sess3 – directories for functional data from each session

./Fieldmaps – fieldmap data

As the pre-processing progresses, you'll notice that various prefixes are added to the images. When the functional data is first imported, the images will have the prefix 'f'. The prefix won't change after realignment if you don't also reslice the images, the transformation will simply be stored in the header of the original files. If you do reslice the images, you end up with a new set of images with the prefix 'rf'. Slice timing adds 'a', normalisation 'w', and smoothing 's'. At the end of the pre-processing pipeline, you end up with a series of images with prefixes like 'swarf' (=smoothed, normalised, slice time corrected, resliced after realignment), 'warf', 'sarf' etc depending on exactly which pre-processing steps you carry out.

You'll also notice that the script creates a file called spm_preprocessing.ps in your MRI_workshop folder. At several points in the preprocessing, various graphs and other output generated by the procedures will be printed to this file.

As you analyse more and more data, the temptation becomes just to run the preprocessing automatically and not really check the results. Whilst the typical preprocessing pipeline is fairly robust, it's always worth spending some time checking your data. At the very least it's worth having a look at the mean and variance images for each session, the results from tsdiffana, the plot generated by the realignment stage, the mean normalised image, and the mean smoothed normalised image.

The workshop is based around the pre_processing.m script, which also includes more information about what's happening at each step. There are also instructions for using the SPM gui below. As well as the information in the script, you can get extra information about each of the spm routines by typing help *(function name)* in the Matlab workspace. A lot of the SPM functions have quite good help sections telling you about what that function is doing, what the various arguments that you pass into the function represent etc.

To get a better idea of what's going on at each line, you can run the script one line at a time. Put a breakpoint on the first line (click the "-" sign on the left of the Matlab editor window next to line 13. A red dot should appear). Press F5 to run the script, then press F10 to run each line in turn. To see the contents of variables at each step, hover the mouse cursor over the variable name. As you step through the script, read the comments (in green) to get a better idea of what's happening on each line.

The first few lines of the script define several variables that will be used, for example the paths of various directories etc. Before running anything, change line 13 to your user name.

```
usr_name = '(your user name)'
```

Now step through lines 13 to 71.

1. Copying the data from the central store and converting to nifti:

The data needs to be copied to the local directory and converted from the dicom format into the format used by SPM.

Script: Lines 81-143 in the script

Gui: click "Dicom Import" in the SPM menu window, and follow the prompts to select the dicom images and output directory.

(NB in the file selection dialogue box, you probably won't be able to navigate to the raw data directory by clicking on the directory links on the left hand side, you'll probably have to type the directory name directly in the "Dir" box at the top)

After this step, you should have 5 folders within your MRI_workshop folder – Structural, Fieldmaps, Sess1, Sess2 and Sess3. Structural should contain one image, and each session of the functional data should contain 120 images.

To get an idea of what each type of scan looks like, you can click "Display" in the SPM gui, and then select the image you want to display. To look at multiple images at the same time, click "Check Reg".

If you display an image, try getting a feel for the SPM display interface – click around on the orthogonal plots at the top, try comparing the voxel space and world space views etc.

2. Image Diagnostics with tsdiffana

Script: Lines 150 to 165

Gui: Type tsdiffana at the matlab prompt, and follow the prompts to select the image files.

Once this is finished, you'll have an image called raw_mean_img.nii and one called vmeanf*.nii in ./Sess1. Load these up and have a look at them. As the name suggests, raw_mean_img is simply an image created by taking the mean value at each voxel across all the functional images. Similarly, vmeanf shows the variance across all images, with brighter voxels showing higher variance. The variance image should look like a white outline around the edge of the brain and around the ventricles, as this is where you tend to get the highest variance (small head movements can change the tissue type drastically, e.g. changing the contents of a voxel from grey matter to air or csf). If the variance image just looks black, try going to the "colors" menu in the graphics window and choosing effects, brighten.

As well as the mean and variance images, tsdiffana should have printed a set of graphs to the first page of the printfile. For more details about what the graphs show, see http://imaging.mrc-cbu.cam.ac.uk/imaging/DataDiagnostics. The values plotted in these graphs are also saved to a mat file called tsdiff.mat saved in ./Sess1.

3. Realignment

Script: Lines 172 to 224

Gui:

- Click the Realign button on the menu window and select Realign.
- A new gui opens in the graphics window.
- In the top right hand panel, click New “Realign: Estimate & Reslice”
- Double click the “+Realign: Estimate & Reslice” subheading that appears underneath “Realign” in the top left hand panel.
- Click on the “Data” subheading that appears underneath “Realign: Estimate”
- In the upper right panel, click “New session”
- In the upper left panel, click on the “Session” subheading that appears underneath “Data”,
- Click “select files” from the top right panel.
- Once you’ve added as many sessions as you want to realign (NB this will realign all the images from all the sessions you’ve specified), double click on “Estimation Options” and “Reslice options” in the top left panel. Have a look through the options and their descriptions (these appear in the bottom panel when you click on each option). Most of the defaults are OK, but to speed things up a bit, you can change “Num Passes” to “Register to first” rather than “Register to mean”

The realignment process will output:

- A new page in the print file showing the translations and rotations applied to each volume (the realignment parameters).
- A file called “rp*.mat” in Sess1. This contains the realignment parameters.
- An image called “meanf*.nii” in Sess1. This is the mean realigned image.
- A new set of ‘rf*.nii’ images in each session directory. These are the resliced realigned images.
- It will also modify the header files of the ‘f*.nii’ images to incorporate the new transformations applied to those images.

Have a look at the graphs of the realignment parameters – especially look for big jumps in parameters. How does the position of any big changes compare to spikes in the tsdiffana graphs? Whereabouts in the time series are the biggest shifts? What’s the overall size of the movements?

4. Slice Timing

Script: Lines 230 to 257

Gui:

- From the menu window, select “Slice Timing”
- A new gui appears in the graphics window.
- In the top left panel, click “Data”, then in the top right panel click “new session”
- Click “session” that appears underneath “Data”, then click select files in the right panel
- Just do one session at a time (don’t want to interpolate across sessions)
- Click on each of the remaining options and read the description in the box at the bottom.

- To enter a value for each option, click on “specify text” in the top right hand panel, type the value in the box that appears, and press enter
- For this data, nslices = 32, TR =2, TA =2-(2/32), slice order = [32:-1:1], and reference slice = 32.

The output of slice timing will be a set of images with the prefix ‘arf’

5. Coregister structural to mean EPI

Script: Lines 263 to 292

Gui:

- From the menu window, select “Coregister”
- A new gui appears in the graphics window
- In the top right panel, click New “Coreg: Estimate”
- A new subheading “Coreg: Estimate” appears underneath “Coreg” in the top left panel. Double click this to open up the menu tree.
- Click “reference image”, then in the top right panel click “select files” – choose the image you want to register the other images to (e.g umeanf*.nii)
- In the top left panel, click “source image”, then in the top right image “select files”. This is the image that will be moved into registration (or “jiggled about” as its helpfully described in the help panel at the bottom…) with the reference image. This image will be used to determine the transformations necessary for registration
- In the top left panel, click “other images”, and “select files” in the top right. These images will be moved using the same transformations derived from the source image.
- Double click “estimation options” in the top left panel, click on each of the parameters and read the description in the help section in the bottom panel.

The output of coregistration includes:

- A new page in the print file showing the results of the coregistration
- A modification to the header of the structural image to incorporate the applied transformation.

Using Check Reg, bring up the mean undistorted EPI image (./Sess1/meanf*.nii) images and the structural image (./Structural/s*.nii). Click around the images and look at the results of the coregistration.

6. Normalise the coregistered structural

The original method of normalising was to trim and skull strip the structural (using the functions trim_img and bet), then to match the structural to the template using the whole image. The new method, which we're going to use here, is more automatic, and involves segmenting the image into white and grey matter first, then normalising these to tissue specific templates. This avoids the need for skull stripping.

The MPRAGE sequence used for structurals at the CBU often has a noticeable attenuation bias, however, with greater signal at the back of the brain than at the front. This can produce unpredictable results if the bias isn't corrected first. The solution to this is a 2 pass procedure – first correct the bias, then segment the bias corrected structural.

Script: Lines 297 to 389

Gui:

- First pass – bias correction only.
- From the menu window, click "Segment"
- Click "Data", then "select files" in the top right panel. Choose the structural image (./Structural/s*.nii), or trimmed structural if you've used the script.
- In the top left panel, double click "output files" to open the menu tree
- Click "grey matter", and in the top right panel, click "specify menu item". Select "none"
- Repeat this for "white matter" and "cerebro-spinal fluid".
- Repeat for "bias corrected", but select "save bias corrected"
- Repeat for "clean up any partitions", but select "do not clean up"
- Double click "custom" to open the menu tree.
- Click on each of the options and read the description of each parameter
- The only one that needs to be changed is "affine regularisation", click this and select "no affine regularisation"

- When this has finished, repeat the process with the following changes:
- When you select the structural, select the bias corrected image (./Structurals/ms*.nii or ./Structurals/mtb_s*.nii)
- For "grey matter" and "white matter" select "native and unmodulated normalised"
- For "affine regularisation" select "ICBM template – European Brains"

The procdure will output:

- A bias corrected version of your structural image (prefix 'm')
- Segmented grey and white matter images derived from the structural (grey matter = 'c1m*.nii', white matter = 'c2m*.nii')
- Normalised segmented grey and white matter images (grey matter = 'wc1m*.nii', white matter = 'wc2m*.nii')
- If you use the script, you'll also have a normalised version of the complete structural image ('wm*.nii')
- A mat file containing the parameters to transform the image from its native space to the template space ('m*seg_sn.mat')
- A mat file containing the parameters for the inverse transformation (i.e. from template space to individual subject space; 'm*seg_inv_sn.mat')

The gui steps described above won't output a normalised version of the complete structural image, but you can generate one by following the steps below. Usually you only need a fairly low resolution version of the normalised structural as its main use is in checking that the normalisation has worked well. The most useful outputs of the segment / normalise procedure are the normalisation parameters, and the grey / white matter images (these can be useful, for example if you want to analyse only grey matter voxels - you can use the grey matter image as a mask).

If you've generated a normalised structural (or if you've used the script), load it up alongside the T1 template (/imaging/local/spm/spm5/templates/T1.nii) using Check Reg to see how well the procedure has worked. Also compare the normalised structural and the normalised grey / white matter partitions using Check Reg. Finally, compare the un-normalised structural and un-normalised grey / white matter images.

7. Apply normalisation parameters to the EPI data

Script: Lines 395 to 414

Gui:

- In the menu window, select "Normalise"
- In the top left panel in the graphics window select New "Normalise: Write"
- In the top left panel, double click "Normalise: Write" to open the menu tree.
- Click "Data", then click "New Subject" in the top right panel.
- A new subheading "subject" appears underneath "Normalise: Write" in the top left panel. Double click this to open the menu
- Click "parameter file", then "select file" in the top right panel. Choose the normalisation parameters created in step 7 ('./Structurals/m*seg_sn.mat')
- In the top left panel, click "Images to write", then select file in the top right panel. Choose the files you want to normalise.
- Double click "Writing options" in the left panel, click on each of the options, and read a description in the bottom window. Most of the default values should be OK, although you might want to increase the voxel size from the default 2x2x2.

Output will be normalised versions of the files you've selected. These will have the prefix 'w' added to the beginning of the file name.

8. Smooth the EPI data

Script: Lines 420 to 445

Gui:

- In the menu window, select "Smooth"
- In the left panel, click "images to smooth", then click "select files" in the top right panel
- In the left panel, click "FWHM", then click "specify text" in the right panel – enter a vector of 3 values to specify the width of smoothing (in mm) in x, y and z directions
- In the left panel, click "Data Type" and select the data type for the output images ('Same' is usually OK)

Output will be smoothed versions of the files you've selected. These will have the prefix 's' added to the beginning of the file name.

Load up ./Sess1/wmeanf*.nii and swmeanf*.nii and compare them using check reg.

AA

In future, you'll probably want to investigate Rhodri's Automatic Analysis (aa) routines. These provide a much more powerful batching system than the simple script we've been using up to now with lots of useful features, for example they're restartable (i.e. they remember where they were up to if they get interrupted) and can handle parallel processing of multiple data sets.

For more information, have a look at :
http://imaging.mrc-cbu.cam.ac.uk/imaging/AutomaticAnalysisIntroduction

Being a good citizen…

Condiseration for other users is the general rule here… The computing resources are finite (at the moment…), and often under heavy demand.

As you can see, even a small data set like this can take up a large amount of disc space. This is a perennial problem as space on the physical discs is quite restricted. There's also a tape system, and older files are shunted off onto tape as the physical discs get full. This is invisible from the point of view of the file system, but files can take a *long* time to come back off tape. When the physical discs fill up, data is being shuffled on and off tape more frequently, the whole system slows down, and everyone's blood pressure goes up.

One obvious way to reduce disc usage is to clear up the images generated by intermediate stages of preprocessing. For univariate analyses you'll generally only need the smoothed normalised images (swarf*.nii), and for MVPA analyses the slice timed realigned images (arf*.nii). You might want to play with a few different smoothing widths (in which case you need the warf images), but once you've got something you're happy with, everything else can probably be safely deleted (and can be easily recreated if need be).

Specifying larger voxels when you write normalised EPI images can also save a huge amount of space. Upsampling the standard EPI resolution to 2mm^3 increases the size of the images 4 fold, and doesn't provide any extra information (it does make prettier pictures though…)

If you're running multiple sessions in parallel, think about how many sessions you'll really need. It's also good etiquette to leave large jobs going overnight, particularly if there's a high load on the Linux machines.

There are also a finite number of Matlab licences, so the more sessions you've got running, the fewer licences will be available for other people. Close any Matlab sessions you've finished with.

Multiple VNC sessions also take resources, particularly if you're using a graphics heavy windows manager like Gnome or KDE. Most window mangers allow multiple desktops, so you can usually get away with a single VNC session.