

## Contents

1. Introduction/Disclaimer
2. Tools of the Trade and Basic Concepts
3. Analysis Pipelines
4. Copying the Data to a Local Directory
5. Data Sorting
6. Data Conversion
7. Slice-Timing Correction
8. Spatial Realignment
9. Co-registration
10. Normalisation
11. Smoothing
12. Pseudo-batch pre-processing with SPM
13. Model Definition and Statistical Analysis in SPM
14. Processing of Retinotopic Mapping Data
15. Visualisation of Anatomical Images Using mri3dX
16. Visualisation of Functional Data Using mri3dX
17. Generating Flat-Maps of Areas of the Cortex
18. Extraction of Time-Courses
19. Calculating % Signal Change
20. Brief Guide to File Formats
21. Data Management
22. Other Useful Resources

## 1. Introduction/Disclaimer

The following is a guide to some of the processes that have been developed for use with anatomical and functional MRI data obtained using the Royal Holloway scanner. It should definitely not be seen as exhaustive, and although I have tried to make it as accurate as possible, doubtless there are some errors. If anyone spots one, please let me know so it can be corrected. Any adverse consequences to your data/career/relationships/life which arise as a result of following the procedures in this document are very much not my fault. Have fun!

Matt Wall (May 2005)  
[matthew.wall@rhul.ac.uk](mailto:matthew.wall@rhul.ac.uk)

## 2. Tools of the Trade and Basic Concepts

### **2.1 Tools of the Trade**

You will need:

- A working (licensed) copy of Matlab.
- SPM2 (or '99, available from the SPM website)
- FSL (available from the FMRIB website)
- mri3dX (available from Krish's mri3dX website)
- Medcon (available from the Medcon website)
- MrGray and mrFlatMesh (Stanford Tools)
- Various bespoke Matlab and c-shell (unix) scripts (mentioned in the notes below as appropriate – available from Matt or Nathalie)

### **2.2 Basic Concepts**

There are a number of basic ideas that it is helpful to understand when starting out analysing fMRI data. If anything in the following section is unfamiliar, have a look at the relevant sources indicated.

#### **Matrices**

It is exceedingly helpful to have an understanding of what a matrix is in a mathematical sense, a basic understanding of matrix algebra, and how this relates to MRI data. Very briefly: MRI data-sets consist of sets of numbers arranged in a hypothetical 3D (or sometimes 4D) structure with the x and y dimensions usually representing the size of the slices (usually 64x64 or 128x128 for functional data) and the z dimension usually representing the number of slices. Performing the processing steps detailed below basically involves mathematical manipulations of these matrices, thus matrix algebra is important. There are thousands of web-sites which will introduce you to the wonders of matrix algebra but Matthew Brett's primer on the MRC-CBU website is particularly recommended:

<http://www.mrc-cbu.cam.ac.uk/Imaging/Common/matrices.pdf> (.pdf document).

#### **The ANALYZE file format**

The ANALYZE format is the data format used in several of the more popular imaging analysis software packages (e.g. SPM and FSL). A good introduction can be found here: [http://www.mrc-cbu.cam.ac.uk/Imaging/Common/analyze\\_fmt.shtml](http://www.mrc-cbu.cam.ac.uk/Imaging/Common/analyze_fmt.shtml)).

### Basic Statistics

Most of the statistical concepts (i.e. independent and dependent variables,  $t$ -statistics,  $z$ -scores) used in the analysis of fMRI data will probably be familiar to those from a psychology/science background. However, it is often the case that they are applied in a somewhat different manner to normal, and the terminology can sometimes be different when talking about imaging data. An excellent introduction to the theory behind SPM statistical analysis can also be found on the MRC-CBU webpages here: <http://www.mrc-cbu.cam.ac.uk/Imaging/Common/spmstats.shtml>.

### Matlab

Matlab is basically a high-level programming language that is based on manipulating matrices. It is also the language in which the majority of SPM is written. Since the SPM authors are generous enough to provide users with their source code knowledge of the Matlab language is helpful, particularly for performing non-standard analyses. There are many online Matlab tutorials available, but probably the best place to start is by typing 'helpdesk' from within Matlab itself. This opens up an HTML help system that contains a number of demos and basic explanations of key Matlab functions.

### 3. Analysis Pipelines

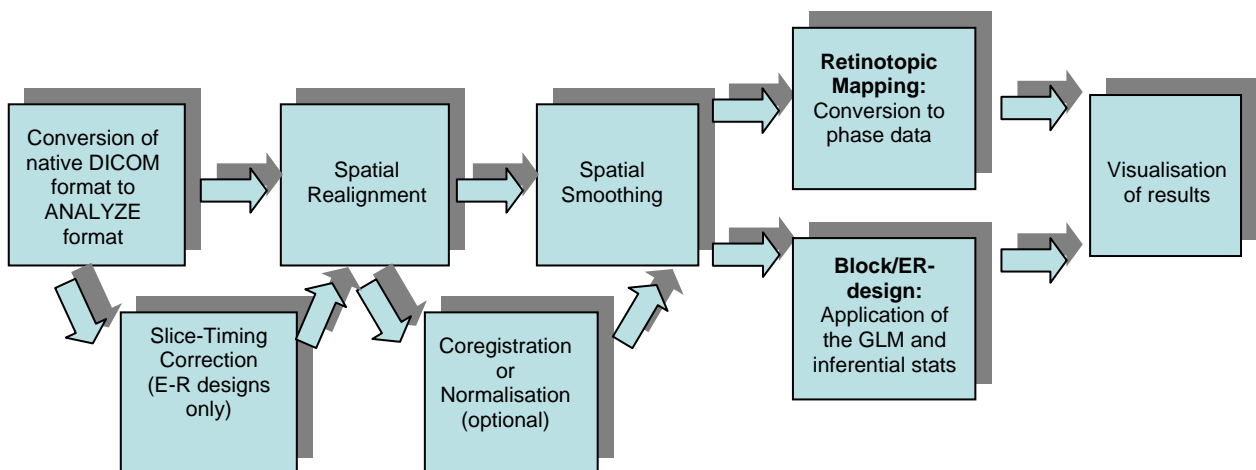


Figure 1. Canonical processing/analysis pipeline for functional MRI data.

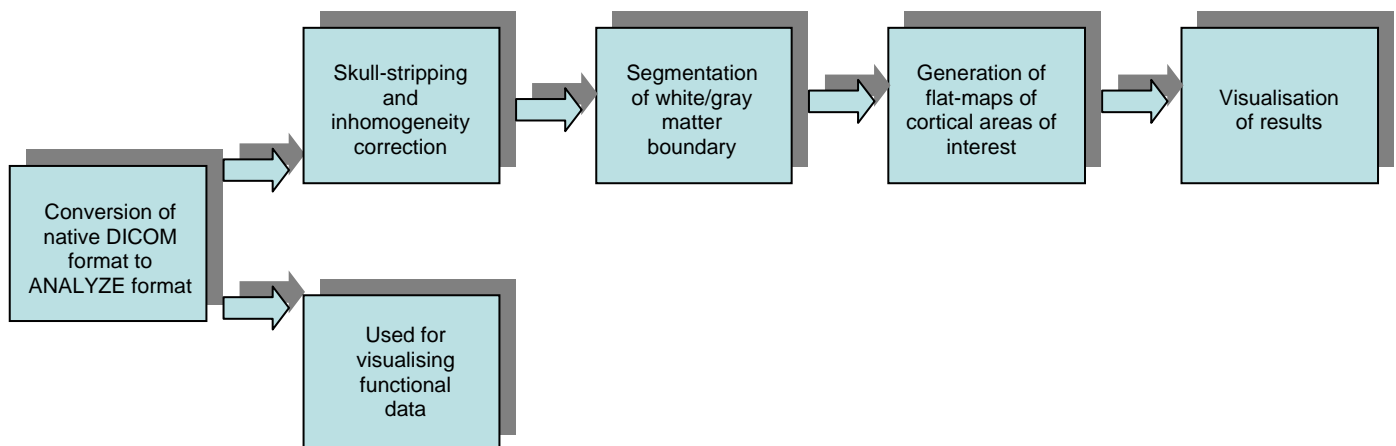


Figure 2. Canonical processing pipeline for anatomical MRI images.

#### 4. Copying the Data into a Local Directory

Open a terminal window, cd to the location where you want the data to end up on your machine and type:

```
ftp 134.219.32.32
```

Then login with your username and password to connect to thalamus. Next, type:

```
passive
```

then:

```
prompt
```

and then:

```
binary
```

Then cd to the directory on thalamus where the data you want is, and type:

```
mget *
```

The data will be copied onto your local machine into the directory you started out in. When it has finished type 'bye' to logout of thalamus and return to your local session.

#### 5. Data Sorting

Before anything can be done the data obtained from the scanner has to be sorted and converted from the DICOM format (.ima files) to the standard ANALYZE format (.img, and .hdr files).

To sort the data, cd to the directory where it is, and type:

```
sortDicom
```

This executes a c-shell script which will sort your data into directories based on the number of runs you performed on the particular subject. Note that if you acquired PACE+MOCO data you will have two directories for each run, the first one containing the PACE data and the second one the PACE+MOCO data. Directory '001' always contains the images from the localiser scan, and is thus ignored. Directory '002' often (though not always) contains the anatomical scans.

#### 6. Data Conversion

There are currently two options available for converting data to ANALYZE format. Both have their advantages, and the optimal method depends on various factors related to the kind of project you are engaged in.

##### **6.1 Option 1**

Conversion using the DICOM conversion toolbox in SPM. Within Matlab, cd to the directory where your data is located. Start SPM by typing 'spm fmri' at the Matlab prompt. Click the button marked 'toolbox' and then select the 'DICOM' option. Select

your data in the pop-up window, and click 'Done'. ANALYZE format files will be written to the current working directory.

## 6.2 Option 2

A new method has been developed by Krish Singh, which uses unix scripts and the Medcon software to convert the data. See Krish's supporting document for his scripts for details on this method.

There is currently some uncertainty about which method is superior for individual-subject analyses. Initially it was believed that the functional-anatomical coregistration was more accurate when option 2 was used. However some testing by Hiroshi has shown this is not always the case – possibly it may depend on the size and orientation of the particular EPI volume, or other factors. **Note that, if your design contains multiple participants which you intend to enter into the same design matrix (and you therefore intend to normalise the data) you *must* use option 1.**

## 7. Slice-Timing Correction

The slice timing correction function in SPM is necessary when an event-related design is used. This corrects for the fact that your functional data is acquired in slices, which are necessarily obtained at slightly different time-points within your TR period. Essentially, the correction shifts the phase of the time-courses of the voxels within each slice to provide data which is as if each slice had been acquired at exactly the same time.

The procedure for doing this is not particularly straightforward and requires some knowledge about how your data is structured. The Royal Holloway scanner typically uses an interleaved sequence (i.e. all odd numbered slices are acquired first, followed by all even-numbered slices) but the exact order differs in the following way:

- If you have an **odd** number of slices, the order goes: 1, 3, 5, 7... 2, 4, 6 etc.
- If you have an **even** number of slices, the order goes: 2, 4, 6, 8.... 1, 3, 5 etc.

There are two methods of doing the correction. One is easier, but not recommended by the SPM gurus (although Krish says doing it this way is fine). The other method is recommended but involves editing the SPM defaults before specifying your model. Both methods have been used at Royal Holloway – in practice the difference between the two methods is likely to be negligible.

To perform the correction the easy way:

1. Hit the 'Slice-timing' button in SPM.
2. Enter '1' for number of sessions.
3. Select your data and hit 'done'
4. Select your sequence type. If you have an odd number of slices you can select 'Interleaved 1, 3, 5, etc.' If you have an even number, you must select 'user specified' and enter a vector in the 'order of slices' box.
5. In the 'Reference slice' box put '1'.
6. Enter your TR in the 'Interscan Interval' box
7. You are then asked to enter your TA.  $TA = TR - (TR/N)$ , where N = the number of slices you have. The TA is calculated for you, but check it if you want and hit return.

To perform the correction the hard way:

1. Hit the 'Slice-timing' button in SPM.
2. Enter '1' for number of sessions.
3. Select your data and hit 'done'
4. Select your sequence type. If you have an odd number of slices you can select 'Interleaved 1, 3, 5, etc.' If you have an even number, you must select 'user specified' and enter a vector in the 'order of slices' box.
5. In the 'Reference slice' box, put the physical or temporal midpoint of your slice acquisitions (SPM inputs the physical midpoint automatically, so just accept this if you want).
6. Enter your TR in the 'Interscan Interval' box
7. You are then asked to enter your TA.  $TA = TR - (TR/N)$ , where N = the number of slices you have. The TA is calculated for you, but check it if you want and hit return.

If you use the second method you then have to change the SPM defaults before you specify your model (see below). To do this:

1. Hit the 'Defaults' button in SPM.
2. Select 'Statistics fMRI'.
3. Accept the default value for 'Upper tail F prob' (0.001)
4. Change 'Number of Bins' to your number of slices.
5. Change 'Sampled Bin' to your reference slice.

**IMPORTANT:** SPM automatically reverts to the normal defaults whenever it is initialised, so you have to change these values every time you shut down and start-up SPM.

Once the slice-timing correction algorithm has run, SPM saves a new set of files with the prefix 'a' – a\*.img.

## 8. Spatial Realignment

There are broadly two options for spatial realignment. Firstly, the data from an individual participant can be realigned one run at a time. In this case, the first image is taken as a reference point and all subsequent images in that run are co-registered to it. The second option is realigning multiple runs at once. In this case the first image of the first run is taken as a reference point, and the first images of all subsequent runs are co-registered to it. Then, the subsequent images in all the runs are co-registered to the first image in their individual runs.

The first option is preferable if you believe the participant may have moved significantly between runs. After performing the realignment the runs can then be normalised or co-registered (see below). **However, in the majority of cases the second option is preferable (more accurate, and also faster) and should therefore be adopted as the standard method.**

### 7.1 One Run at a Time

In SPM select the 'Realign' button and then the 'Realign' option. Press enter twice to input the number of subjects ('1') and number of sessions ('1'). Select your data files in the dialog box and click 'Done'. Then choose 'Co-register and re-slice' from the drop-down menu, and 'All images + mean image' from the second drop-down menu. The realign process will take place and you will see graphs of the movement

(translation or rotation) of the participant over time. Large movements on these graphs (more than 1-2mm or 1-2°) are cause for concern.

## 7.2 Multiple Runs

In SPM select the 'Realign' button and then the 'Realign' option. Press enter once to input the number of subjects ('1') and then enter the number of sessions (i.e. number of runs you want to realign together). Select your data files for the first run in the dialog box and click 'Done', then select the files for the second run and so on. When you have selected data for all your sessions, choose 'Co-register and re-slice' from the drop-down menu, and 'All images + mean image' from the second drop-down menu. The realign process will take place and you will see graphs of the movement (translation or rotation) of the participant over time. Large movements on these graphs (more than 1-2mm or 1-2°) are cause for concern.

In both cases SPM generates a new set of realigned files of the format r\*.img.

SPM2 also generates a text file of the format rp\_\*.img which contains a six column matrix, with a number of rows equal to the number of scans. The first three columns in the matrix are translation in the x, y, and z dimensions, and the last three columns are rotation around the x, y, and z axes in radians. These figures can later be entered into a SPM design matrix as confounding variables, which effectively mitigates the effect any remaining movement-related artefacts may have on the data. See Matt for details on how to do this if you require them.

## 9. Co-registration

Co-registration is used when multiple sessions from the same participant need to be compared, or incorporated into a model together. It is also used if a participant has left the scanner (i.e. to take a break) in the middle of a session. In this case, the later runs in the session would be co-registered to an earlier run.

Three sets of images have to be specified:

- The 'Target' volume is the image you are co-registering to.
- The 'Source' image is the first image in the run you want to co-register, or the mean image of that run (generated by the realignment process).
- 'Other Images' are all the individual images in the run you want to co-register.

Co-registration works best if all the images are the same modality (i.e. all EPI or all MP-RAGE) therefore, when co-registering EPI data the EPI run closest in time to the anatomy is normally used as the target volume.

Click the 'Co-register' button and enter the number of subjects you have (usually '1'), then select 'Co-register and reslice'. Select your target image and click 'Done', then select the source image and click 'Done' and finally select your individual files as 'Other Images' and click 'Done'. The co-registration process will execute and after a while the target image and the source image will be displayed, along with a couple of graphs depicting the details of the affine transformation.

SPM generates a new set of files after co-registration – also with the prefix 'r'. If your files were also re-aligned, they will now be of the format rr\*.img.

## 10. Normalisation

Normalisation is used to transform images from individual participants into a standard Talairach-type (note: not “Talairarch” but “Talairarch-type”) space in order that images from different participants can be directly compared. Often for this process the default settings in SPM need to be changed, so check with someone before you do it the first time.

Three sets of images have to be specified.

- The ‘Template’ is usually a standard brain template supplied with SPM, although others can also be used if required.
- The ‘Source’ image is usually a mean image created during the realignment process.
- ‘Images to Write’ are the individual images in the run you want to normalise.

Click ‘Normalise’ and then select ‘Determine parameters and write normalised’. Select the template you wish to use (EPI.mnc is the usual choice for normalising EPI images) and click ‘Done’. Then select the source image of the run you want to normalise (usually a mean image created during realignment) and click ‘Done’. Then select all the individual images of the run you want to process and again click ‘Done’. The dialog box will pop-up again asking you to select another source image (multiple runs can be normalised at the same time). When you have finished selecting all the runs you want to normalise for that subject, click ‘Done’ without selecting another source image and that will start the normalisation process.

SPM generates a new set of files with the prefix ‘w’ after normalisation. Your files will likely now be in the format `wrr*.img`. (SPM '99 prefixes an ‘n’ instead of a ‘w’.)

## 11. Smoothing

Spatial smoothing with a Gaussian kernel is used to reduce the effects of random noise on EPI data, and is almost always desirable. Various recommendations have been made regarding the size of the kernel to use, however the size of the voxels  $\times 2$  is usually a good value to use.

Click the ‘Smooth’ button and input the size of the Gaussian function you wish to use (2 x voxel size). Select the files you wish to smooth, and click ‘Done’.

**Top tip:** Multiple sets of data can be selected for smoothing at once – SPM smooths them all and dumps the smoothed files back in the correct directories. Leave the smoothing of all your data until last, select it all at once, then go for lunch!

SPM generates a new set of files with the prefix ‘s’ after smoothing. If you have performed all the steps above your files *may* now be in the format `swrra*.img`.

## 12. Pseudo-Batch Processing with SPM

All the pre-processing steps described in sections 6-10 above are often arduous and boring to perform on large datasets. To mitigate this, SPM allows processes to be initiated and run from the Matlab command-line, and therefore scripts can be written to automatically process large amounts of data at once and can be left running overnight, or while you slip off for a cheeky one in the student bar. This is SPM batch mode, although difficulties associated with it often lead people to replace the 'a' of batch with a different vowel when referring to it. Described below is a much less-powerful (but also much simpler) method of speeding up processing of data; hence pseudo-batch mode.

This method basically involves starting up multiple instances of Matlab, and multiple instances of SPM, and running processes in all of them at once. This is a very memory-demanding task, and Matlab's normal GUI (Graphical User Interface) is also quite memory-demanding and slow as it uses a Java Virtual Machine (JVM). Fortunately it's possible to start up Matlab without the normal GUI and JVM by typing:

```
matlab -nojvm -nosplash
```

at the normal unix prompt. (The '-nosplash' option just means the splash screen with the matlab logo is not displayed and matlab starts a bit faster.)

So...

1. In a unix terminal, cd to the directory where your first run of data resides.
2. Start up Matlab without the java interface as described above.
3. Start up SPM normally by typing 'spm fmri'.
4. Perform the steps necessary to initiate whatever process you require.
5. Once the process is running, open another terminal window, cd to the directory of your second run of data, and repeat the process.
6. Continue repeating until processes are running for all your data.

The number of different instances of Matlab and SPM you can start up will be dependent on the amount of physical and virtual memory available to you. My G5 with 1.5Gb of RAM runs ten instances absolutely fine. You should find that although each individual process is slower, you can perform a process on a set of data significantly faster than if you did all the processes individually on each run.

Note that you can use this technique for any pre-processing stage, but there's little point for smoothing as SPM allows you to select multiple runs at once anyway.

## 13. Model Definition and Statistical Analysis in SPM

### **13.1 Model Definition**

A detailed guide to model definition in SPM is outside the scope of this document, and in any case, any general information provided would not be sufficient to address the variety of designs that could be utilised.

However, there are several stages in the model definition and analysis that are common to all designs.

Models are defined by clicking the 'fMRI' button and then selecting 'Design'. You are then asked to enter various parameters and select various options including:

- The TR (in seconds).
- The number of scans you have in each session.
- The vector of onsets for the stimuli in all your conditions.
- Any other regressors you wish to include in the model.

**Top tip:** Sometimes vectors of onsets or definition of other regressors are complex and arduous to type in by hand. In such cases the vectors can be written into a plain text file and saved beforehand. When SPM requires the input of the onset vector, typing 'spm\_load' into the box will bring up a dialog box, allowing you to select the text file.

When the model has been defined you will see a set of figures in the display window. The top left figure illustrates the regressor of the first condition of the first session of the model you have defined. By clicking on 'Explore model' in the interactive window you will be able to see the subsequent sessions and trial conditions.

The bottom left figure illustrates the basis set you have selected (i.e. either the standard HRF, a fourier basis set, a FIR basis set, or whatever). If you are using the canonical HRF you will see it here. Along the bottom of the graph will also be a set of green dots, which represent the sampling of the HRF. For a block-design experiment this is not so important, however for an event-related design it is crucial to ensure that the HRF is adequately sampled, in order to prevent aliasing. If the HRF is not adequately sampled (i.e. the green dots are far apart) the design of the experiment needs to be adjusted to improve the sampling rate. This is achieved by adjusting the degree of 'jitter' of the stimulus presentations or trials compared to the TR; in practice this usually means adjusting the inter-trial intervals to be pseudo-random. What actually constitutes an optimal sampling rate is (yes, you guessed it... sigh...) a matter of debate. However, sampling at say, 3-6 Hz (i.e. every 333-166 ms) is almost certainly adequate to avoid aliasing of the HRF. Sampling at 10 Hz (i.e. every 100 ms) is more than sufficient for most experiments, and is not unrealistic in terms of experimental design either.

The model definition file is saved to the working directory, and is always called SPM.mat.

**Top tip:** If you are defining a number of identical models, the SPM.mat file can be copied into other directories at this stage. Thereafter, the model definition stage can be skipped for the subsequent models, and the same SPM.mat file can be used. It is good to get into the habit of saving a copy of the model in another location at this point. If you decide subsequently that you want to change some aspect of the analysis it is a simple matter to copy your saved SPM.mat file to another location and reassign the data. If you have not saved another copy of the model you will have to specify the model all over again – a potentially arduous task with a multi-subject event-related design experiment.

### 13.2 Assigning Data to the Model

After the model has been defined you have to assign data to it. To do this, click the 'fMRI' button again and select 'Data'. Select the SPM.mat file of the model you wish to assign data to and click 'Done'. Select the set of files you want for session 1 and click 'Done'. Carry on selecting files until you have assigned a set of data to all the sessions you have in your model.

**Important Note:** Once you have assigned your data to the model, the SPM.mat file and the data become inextricably linked i.e. the SPM.mat file contains pointers to the location of the data in your directory structure. If you subsequently move either the data or the SPM.mat file, or rename any of the super-ordinate directories in which they reside, estimating that model will be impossible, and you will either have to move the files back to their original location or start again by assigning the data in its new location to a new model.

### 13.3 High-Pass Filtering

After selecting your data SPM will ask you to select various options. The high-pass filtering option should always be applied. The optimal cut-off value for the high-pass filter is the time in seconds between a repetition of the same event or stimulus, multiplied by 2. In the case of a simple ABAB block design, this would be the SOA between event type A and event type and the repeat of event type A multiplied by 2. If each block lasts 15 seconds, there are 30 seconds between the onset of the two 'A' events, therefore the filter value should be 60.

If your design is more complex (i.e. is event-related, or has many blocked conditions) then the optimal value for the high-pass filter is more difficult to calculate. Imagine a blocked design with eight different event types. Each lasts 15 seconds and all are presented before the first one is repeated. In this case the SOA is 120 seconds, so following the rule above, the filter should be set at a value of 240. However, this may be too low a threshold to set, and still leave you vulnerable to the effects of low-frequency noise. In such cases a bit of empirical investigation may be required to determine which value of the filter most effectively attenuates the noise in your data.

Advice from Karl Friston on this topic: <http://www.jiscmail.ac.uk/cgi-bin/webadmin?A2=ind9910&L=spm&P=R586&I=-1>

After assigning all the data you will see the SPM design matrix displayed, with one column for every condition and regressor you have entered. If you have defined multiple sessions, you will also have one row for each session. There will also be a final set of columns (coloured plain white) that represent each session you have entered.

### 13.4 Performing the Stats

After assigning the data you are now ready to perform the real data-crunching part of the analysis. To do this, click the 'Estimate' button, and select the SPM.mat file of the model you wish to estimate.

If you have a large and complex model (multiple sessions and multiple subjects) this step may take some time (up to several hours, or even days) to complete.

The results of this process are saved into the working directory, and are known as beta files, as they have the format 'beta\_000\*.img'. You will have one beta file for every column of your design matrix, and 'beta\_0001.img' relates to column 1,

'beta\_0002.img' relates to column 2, and so on. Beta files are the raw intensity values related to the various conditions that were earlier specified in the design.

### 13.5 Defining Contrasts and Seeing the Results

As with section 13.1 above on model definition, the contrast stage of the analysis varies hugely depending on the experimental design, and therefore any general information is of limited usefulness. However...

Hit the 'Results' button and select the SPM.mat file of your model. This brings up the contrast manager. Hit 'define' to make a new contrast. Enter a name for your contrast and the contrast vector in the boxes on the left and hit 'submit'. You will see a graphic representation of your contrast superimposed over your design matrix on the right side of the box.

Points about contrasts:

- SPM allows you to define any contrasts you like, however meaningful  $t$ -contrasts always sum to 0.
- SPM adds zeros to the end of your contrast automatically, so for a six-column design matrix, SPM will interpret a contrast of the form '1 0 -1' as actually being: '1 0 -1 0 0 0'.
- $t$  contrasts require a vector,  $F$  contrasts can be either a vector or a matrix (for examining  $n$ -way interaction effects).

When you are happy with the contrast you have defined hit 'Done' to take you back to the contrast manager. Hit 'Done' again, and you then get a few other options presented in the interactive window. To display a set of basic results select the default or 'none' for all these options. You will then see the results of your analysis displayed on a glass brain, together with a miniature version of your design matrix.

There are various other ways available of visualising and interrogating your results. Results can be overlaid onto an appropriate anatomical image, time-courses at individual voxels can be plotted and the various statistics for any particular voxel can be tabulated. These functions are accessed through the buttons that appear in the interactive window when results are displayed. See the SPM manual for more details on these functions.

This process generates two new sets of files.

Con files (of the format 'con\_000\*.img') are linear combinations of the beta files, weighted by the contrast vector that is defined. Therefore, the con file of the contrast vector 1 -1 will be the result of the subtraction of the second beta file from the first. A con file of the contrast vector 1 0 will be identical to the first beta file. Con files are numbered according to the position of the contrast in the contrast manager.

Every contrast also generates a spmT file ('spmT\_000\*.img'). There is a direct one-to-one correspondence between the numbered con files and spmT files. These files contain the  $t$  statistics that resulted from the comparison of the beta files.

### 13.6 Specifying a Model Using a Text File

This section details an alternative method of specifying a model, by saving the required input in a text file and pasting it into the Matlab command line without using the SPM GUI. It requires a certain familiarity with the way SPM works and is not terribly straightforward so is probably not recommended for absolute beginners!

However, it is an absolute godsend if you have a large multi-subject model to specify, particularly if the design is event-related with true randomisation of the stimulus presentations. In such cases there may be several hundred onset vectors to input – doing this using the GUI is fairly torturous, and also much more subject to error.

Method:

1. Open Matlab and SPM normally and hit the 'defaults' button.
2. Select 'Miscellaneous Defaults'.
3. Say 'no' to the first question ('Log to File?').
4. Select 'GUI for files, Cmd line for input' from the next drop-down menu.
5. Say 'no' to the next question ('Allow multi-volume Analyze files?').
6. Accept the default value for the next option ('Grid-volume 1-10').

SPM will now accept input at the Matlab command line, but still display the file selection dialog box when required.

Next *without exiting SPM* (if you exit, the defaults you just changed are reset):

1. Click the 'fMRI' button.
2. Bring the main Matlab window to the front and you will see the question normally asked in the SPM interactive window ('Specify design or data') at the Matlab prompt with two options: design(1) or data(2).
3. Open a plain text file and put '1' on the first line. Also input '1' at the Matlab prompt.
4. You will then be asked for your interscan interval. Enter it on the second line of your text file and also enter it at the command prompt.
5. Do the same with the next question ('Scans per session'); put it on the third line of your text file and also into Matlab.
6. Carry on doing this until you have specified all the conditions in your first run.

Once you have performed this process for the first run you can then duplicate (copy-and-paste onto lower lines) the relevant parts of the text file relating to your later runs, changing the onset vectors where necessary.

You can then at some later point, get SPM into command-line-input-mode and copy-and-paste the contents of your text file at the prompt when it asks you 'Specify design or data?'. As long as you've prepared the text file correctly, it will whizz through and specify your model almost instantly. Of course, once you've performed this process once, you can save and re-use the text file with relevant changes to produce other models just as quickly.

If anyone wants to use this method and is unsure exactly how to go about it, go and see Matt Wall for a demonstration.

## 14. Processing of Retinotopic Mapping Data

**Important Note: For retinotopic mapping analysis with phase2fun (see below) the number of slices in your images must always be odd.**

Retinotopic data analysis follows a somewhat different process to that outlined above. After pre-processing in SPM (note: retinotopic data is *never* normalised) the individual scans are concatenated into a 4-D file, and then phase data is generated by a bespoke script written for that purpose.

#### 14.1 Data Concatenation

The FSL program 'avwmerge' transforms your set of 3-D image files into one, larger, 4-D file. To use this function type at the unix prompt:

```
avwmerge -t [newfilename.img] [s*.img]
```

Where [newfilename.img] = the name of the file you are creating and [s\*.img] = the set of 3-D files you are transforming.

#### 14.2 Phase2fun

Phase2fun is a bespoke script written by Krish Singh for generating a .fun file from Analyze-format retinotopic mapping data. Various variables have to be entered:

```
phase2fun <in> <out> <dir> <corr> <slices> <time> <cycles> <TR> <xy>
```

in : ANALYZE format file

out : name of new FUN file

dir : [0=CCW 1=CW] or [0=EXP 1=CON]

corr : [-0.8726643 for wedges] or [0.34906585 for rings]

slices : number of slices in the image

time : number of timepoints (or TR's)

cycles : number of times stimulus is repeated

TR : length of single TR in seconds

xy : dimension in pixels of the image (64, 128, or 256)

Most of these are straightforward, except perhaps the 'corr' variable, This variable models both the size of the wedge and its initial position or phase. The value you input should therefore be the offset of the initial starting position of the centre of the wedge from the 12 o'clock position, expressed in radians (proportions of  $\pi$ ). Here is how to calculate your 'corr' value for the two types of commonly used retinotopic mapping stimuli (wedges and rings):

##### Wedges

If the middle of the wedge is on the upper vertical meridian (12 o'clock) at the start of the cycle, then no correction is necessary. Otherwise, a correction is applied that relates to clockwise (CW) wedge rotation. For counter-clockwise (CCW), the correction is automatically adjusted by phase2fun. The sign of the correction is determined by whether the observed phase is advanced or retarded by the actual start position. For example, if the wedge starts centred at 3 o'clock, the phase observed in the data will be advanced by 90deg compared to the expected value. So the correction should be negative, to retard it. i.e. -90 deg or  $-\pi/2$  rad. When the start position is at 9 o'clock, the correction is -270deg. This is the same as +90deg and it doesn't matter which is specified. For simplicity, you could always specify a positive angle, measured going CCW from 12 o'clock, so the first example becomes +270 rather than -90.

## Rings

Here the principle is the same but it is more difficult to conceptualise. The correction is again specified in radians, so we have to think of steps of the ring in degrees of "rotation" (which should not be confused with degrees of visual angle). This is done by analogy with wedges. We typically use an 80 degree wedge. This fills 80/360 of the whole circle. So an "80 deg" ring is one whose width is 80/360 of the stimulus radius. If it makes 18 steps in a complete cycle then each step is "20 deg" i.e.  $360/18$ .

No correction is needed if the middle of the ring starts at the fovea. Because the ring turns into a disk at the fovea, the "middle" is more easily thought of in terms of time than space. With 18 steps/cycle and an "80 deg" ring, the fovea is covered by the stimulus during 4 consecutive ring positions and the "middle" (in time) is the transition between the 2nd and 3rd of the 4 time periods. So if that's where the stimulus cycle starts, the correction is zero. Otherwise, a correction of 20deg for each step away from this position is applied. The correction is considered in relation to a CONTRACTING ring and the expansion case is handled by phase2fun. For example, if the ring is contracting and the cycle starts when the ring has just left the fovea completely, the observed phase will be advanced by 40deg and so the correction needs to be -40deg to compensate.

The output of the 'phase2fun' program is a .fun file, which can be visualized on flatmaps in mri3dX (see section 15 below).

### 14.3 Averaging .fun Files

A unix script exists which combines .fun files containing phase information and produces an average .fun file. To use this type:

```
AvPhaseWrap file1.fun file2.fun fileN.fun... outputfilename.fun
```

at the unix prompt. Simply list all the fun files you want to average, followed by the file name of the new .fun file you want to generate.

## 15. Visualisation of Anatomical Images Using mri3dX

### 15.1 Generating .mtf Files

Mri3dX is a sophisticated piece of visualisation software written by Krish Singh.

In order to load anatomical volumes into mri3dX, .mtf files need to be created. These files perform a similar function to .mat files in SPM and contain information relating to the position and orientation of the volume.

There is a Matlab script called spmMTF.m which enables the creation of .mtf files from .mat files. To use this script, type:

```
spmMTF('* .mat',256,256,176,'* .mtf')
```

at the Matlab prompt, where '\* .mat' is the name of the .mat file, '256,256,176' are the x-dimensions, y-dimensions and number of slices of your image respectively, and '\* .mtf' is the name of the .mtf file you wish to create. This will create a .mtf file and dump it in the working directory.

## 15.2 Viewing Anatomies in mri3dX

You can now load your anatomical image in mri3dX. To do this, cd to the directory where the image is located, and at the unix prompt type:

```
mri3dX filename.img
```

A splash screen should appear for a couple of seconds, followed by a display of your anatomical image in three orthogonal slices in the main mri3dX window.

## 16. Visualisation of Functional Data Using mri3dX

### 16.1 Generating .mtf Files

In order to visualize the output of SPM analyses (con\_\*.img and spmT\_\*.img files) in mri3dX .mtf files also need to be created in a similar way. For functional data the script is called 'spmMTFfun'. To use it, type:

```
spmMTFfun('*.mat',64,64,28,'*.mtf')
```

at the Matlab prompt, where '\*.mat' is the name of the .mat file, '64,64,28' are the x-dimensions, y-dimensions and number of slices of your image respectively, and '\*.mtf' is the name of the .mtf file you wish to create (use the same file name root as your original file). This will create a .mtf file and dump it in the working directory.

### 16.2 Viewing SPM Output Files in mri3dX

To view your functional data set in mri3dX, first load an anatomical image as described in section 13 above. Then from the 'Fun' menu select 'Load Analyze Data'. Select your Analyze format file and click 'open'. If everything is correct, your functional dataset should appear overlaid on your anatomy in near-perfect registration. A new window also opens which gives you various options for changing the view of your Analyze data (opacity, thresholding, etc.)

### 16.3 Viewing .fun Files in mri3dX

To view your .fun data set in mri3dX, first load an anatomical image as described in section 13 above, except that the switch '-fun' should be inserted after 'mri3dX' when you load your anatomical image, so:

```
mri3dX -fun filename.img
```

This starts up a slightly different version of mri3dX optimised for viewing .fun files. First of all deactivate the 'Match Origins' option from the 'Fun' menu. Then, to view your .fun file, open the 'Fun' menu and select 'Load FUN File'. Select your file and then click 'open'.

For further information about mri3dX and what it can do see its website:

<http://www.aston.ac.uk/lhs/staff/singhkd/mri3dX/>

## 17. Generating Flat-Maps of Areas of the Cortex

For the purposes of clarity, it is often desirable to visualise functional datasets on a flattened 2-D representation of an area of the cortex. A set of tools developed by the Stanford imaging group enables this to be achieved.

Extensive documentation on mrGray and mrFlatMesh can be found on this website: <http://info-center.ccit.arizona.edu/~cni/mrgray.htm>. The steps below should be taken as an example only – generating good flat-maps seems to require a certain amount of practice and even a bit of intuition!

### 17.1 Skull-Stripping

The first stage is to extract the brain from your anatomical image. FSL tools provides an automatic method of doing this. To use it type:

```
bet [inputfilename] [outputfilename]
```

at the unix prompt. Note that just the roots of the filenames (i.e. without any extensions) are used. A file with the output file name and the suffix '\_brain.img' is created.

### 17.2 Inhomogeneity Correction

Images from our scanner are usually inhomogeneous in contrast – the back of the brain is lighter than the front. This makes segmentation and flat-mapping more difficult so to correct it the FSL inhomogeneity correction tool (called 'fast') can be used. To use this, type:

```
fast -or [filename.img]
```

at the unix prompt. Unfortunately, it's anything but fast – usually takes a few minutes. At the end a file with the suffix '\_restore.img' will be created.

### 17.3 Conversion to .dat format

The next stage is to convert your .img file to the .dat format used by mrGray. Another Matlab script performs this function. Type:

```
spm2dat ('filename_brain_restore.img')
```

at the Matlab prompt, and a .dat file will be created with the same file name as your .img file.

### 17.4 mrGray

The steps outlined below are intended as a reminder only – you really need someone to show you how it's done the first couple of times.

1. View > Set Grayscale. Some good figures are: 0.17 (brightness), 0.53 (contrast) and 1.2 (Gamma).
2. Set the boundaries of your volume of interest. Scroll through the slices to make sure it's all OK. Anterior boundary: Position at the posterior edge of the corpus callosum. Ventral boundary - Ventral edge of occipital lobe. Dorsal boundary - Dorsal edge of occipital lobe (parietal/occipital sulcus). Posterior boundary - posterior edge of occipital lobe.
3. Open the classification editor (set the crossbars to a nice area near your volume of interest first). Constant values: Noise = 22, Confidence = 0.9,

Smoothness = 2.

4. Generate values for the CSF, White matter and Gray matter by clicking on the sample and then previewing it. Adjust manually if necessary.
5. Select continuous white matter. Classification > delete > unselected white matter. Gets rid of cerebellum and contra-lateral hemisphere white matter.
6. Editing. Have to edit out the gray pixels around the ventricle (where no gray matter actually exists) and wall off the other hemisphere to prevent contralateral gray matter being included in the flat-map.
7. Grow the gray matter again, and select to check it looks OK. Use the Classification > Topology > Analysis box to check how well you've done.
8. Run 3D > build visualisation. Select 'white/gray boundary' under 'Interface to render'. Change voxel size to 1 x 1 x 1. Skip smoothing and decimation steps, change folder of output file. Produces .mrm file.
9. Pick a spot for the centre of your flat-mesh. Position the crossbars on a spot in the Gray matter (the middle of the calcarine sulcus is a good one for visual stuff) and note down the co-ordinates. Remember the order (Z Y X).

### 17.5 mrFlatMesh

1. Back at the Matlab prompt, type mrFlatMesh. Select the two input files (.gray and .mrm) that were generated in mrGray.
2. Set the coordinates for the centre of the flat map - remembering they're in a different order here (X Y Z).
3. Change the voxel size to 1 x 1 x 1.
4. Put in a filename for the output file (with a .amap extension).
5. Hit 'go'.
6. Check all the figures look OK.

Quite often mrFlatMesh fails to generate a map, or the figures it produces look strange, indicating that your flatmap is not that good. Usually this can be remedied by simply choosing another set of coordinates close to your original set, and/or by reducing the size of the flatmap slightly (the default is 50 mm). If the flatmap still looks rubbish, you may need to go back to mrGray and re-do the segmentation.

### 17.6 Viewing Flatmaps in mri3dX

Load your anatomical images and functional images, as described in sections 13 and 14 above. Then click the 'View' menu, followed by the 'Flatmap' option. A new window will open. Select the 'File' menu in the new window and then the 'Load ASCII Flat map' option. Select your flat-map and click 'open'. Your flatmap will be displayed. You will have to click 'File > Recalc' to see your functional data, and also do this every time you change the viewing options of your functional data, in order to see the new results on the flatmap.

## 18. Extraction of Time Courses

Currently two options exist for the extraction of time-course data from ANALYZE format files.

1. MarsBar. MarsBar is an extension-toolbox to SPM which allows average time-courses from an ROI to be computed. There is some documentation available online here: <http://marsbar.sourceforge.net/>, and Hiroshi has also written a brief guide to using MarsBar – see Matt for a copy if you need one.
2. Hiroshi has also written a bespoke script for dumping the average time-course from a Mri3dX flatmap-defined ROI into a simple text file. See Matt for instructions on how to use this.

## 19. Calculating % Signal Change

Although calculating percentages might seem to be a simple task, when trying to express the magnitude of an effect in terms of percent signal change it actually turns out to be less than straightforward. The problem is that of finding an appropriate baseline or denominator to divide the numerator by. Because the units in which the raw scanner data are expressed are essentially arbitrary and consist of interval data (i.e. data which has a fixed interval between each point of the scale, but not necessarily a true zero point) calculating an appropriate baseline is problematic.

Various solutions exist, and some have been instantiated into software packages such as MarsBar. However, it should be noted that because of the nature of the problem a truly exact solution is theoretically and practically impossible. See the MarsBar and SPM mailing lists for some fairly extensive discussions on the issue.

Note that when talking about the magnitude (in real numbers) of an effect it is **not** appropriate to use `spmT*.img` files. These files contain  $t$  values expressing the statistical significance of a difference between two (or a set of) images. These  $t$  values are affected by the degrees of freedom of the data matrix as well as the magnitude of the effect. Therefore, when talking about effect sizes in real numbers, the `beta*.img` and `con*.img` files should **always** be used.

One approach that has been used successfully in our lab is to take the mean of all voxels in the ROI that you are interested in, and use that as the baseline or denominator to calculate % signal change. This procedure uses the session-specific regressors that SPM automatically generates on the right-hand-side of a design matrix (the white bars). If you have six sessions in your model, with say, two conditions in each session, during the estimation process SPM will generate 18 `beta*.img` files; the first 12 represent the parameter estimates for your two conditions across the six sessions, whereas the last six represent the session-specific regressors.

Follow this procedure:

1. Generate a mean image from all your session-specific `beta*.img` files using `ImCalc` in SPM or `avwmaths` in FSL.
2. Load this image onto an anatomy/flatmap in `mri3dX`, load your ROI and use the 'calculate' button to get the mean of all voxels in your ROI. This is your denominator.
3. Load the contrast (i.e. `con*.img` file) you're actually interested in onto the same flatmap and ROI in `mri3dX` and hit 'calculate' again. This is your numerator.

4. Calculate the percentage signal change:  $(\text{numerator}/\text{denominator}) * 100$ .

## 20. Guide to File Formats

There are many different formats for MRI data currently in use. The following is a brief guide to some of the more commonly-used ones.

### **20.1 ANALYZE Format Files**

**.img files.** This is the format used by SPM and various other pieces of software, and are the files which contain the actual data from an image. Typically, they consist of a 3-dimensional matrix, and there will be one .img file for every volume that was acquired in any given scanning run. Sometimes they can exist as a 4-dimensional matrix, in which case there will be one large file for each scanning run, and individual time-points within a run will be represented within the file.

**.hdr files.** These are the companions to .img files. Every .img file must always have a .hdr file that has the same name, and they must exist in the same directory. The .hdr files contain information related to the size and orientation of the data matrix contained in the .img files.

**.mat files.** Matlab matrix files i.e. contains n-dimensional data matrices that can be manipulated using Matlab. .mat files are created by SPM during some of the pre-processing stages, and in this guise contain information about movement parameters (for realignment) or deformation parameters (for normalisation). SPM design files have the extension .mat, and .mat files can also be used for loading variables into the Matlab workspace when specifying a design. .mat files can therefore perform many functions, and the .mat extension does not necessarily signify a particular kind of data structure or function for the file.

### **20.2 BrainTools (i.e. mri3dX) Format Files**

**.fun files.** This is a format invented by Krish Singh for storing functional data. The main advantage of this format is that whereas .img files only store one floating-point value in each cell of the matrix, .fun files can store two. This makes them useful for procedures like retinotopic mapping, where it is desirable to threshold images on a correlation value, but actually view other data.

**.mtf files.** These are the BrainTools equivalent of .hdr files i.e. they contain information about the position, size and orientation of the data matrix stored in .fun and .img files. Required for viewing functional data of all formats in mri3dX.

**.kmr files.** This is a format for anatomical images, which is now largely defunct.

**.flatroi files.** These are files created when you save a ROI definition on a flatmap in mri3dX. They can be saved and later loaded back onto the same flatmap.

### 20.3 Miscellaneous Formats

**.JMA files.** This is the DICOM format in which data comes off the scanner. It can be transformed into ANALYZE format using the SPM DICOM toolbox, or Medcon. Can also be viewed in their 'raw' state by using a DICOM viewer – various viewers are available for various computing platforms. The best one for Mac OS X is called 'Osirix' – available for free download from the authors.

**.dcm files.** These files are an alternative extension for the DICOM format.

**.dat files.** Amongst other things, this is the format for anatomical images used by mrGray. ANALYZE images are transformed to .dat format using Adrian's 'spm2dat' script.

**.MrM, .Gray, .Class, .mrp files.** These are all formats used by mrGray for generating a 3-D model of a part of the cortex. .mrm and .Gray files are both required for generating a flatmap using mrFlatMesh.

**.amap files.** Flatmaps generated by mrFlatMesh. Basically they are ASCII text files containing a list of x-y-z co-ordinates. Can be viewed as flatmaps in mri3dX.

**.m files.** These are Matlab program files – containing code written in the Matlab programming language. The majority of the SPM application exists as an inter-linked set of .m functions. .m scripts can be anything from a two-line throw-away function for automating a tedious task (e.g. triggering a series of other scripts to run in sequence overnight) to thousands of lines of code which instantiate a major application (e.g. SPM).

**.nii files.** These are the NIFTI file format, which the new version (v. 3.2) of FSL can use (although it can still also use ANALYZE format data). Annoyingly, this version of FSL (including the useful avw-utilities like avwswapdim) now generates .nii files by default, instead of .img. However, this setting can be changed in one of the FSL preference files – see the FSL website for instructions.

This list is definitely not exhaustive! Quite a few other formats exist which are less commonly used. If there's a glaring omission from this list, please let me know...

## 21. Data Management

Data management presents a significant challenge in MRI research, simply because of the size of the datasets that need to be handled. It would not be unusual for a single group study with say, 12 participants to require > 20 Gb of hard-drive space. Even though processing power and storage capacity of computers is still increasing at an accelerated (if not quite exponential) rate this is likely to pose a significant problem for many years to come. A related issue is the policy adopted relating to backing-up and archiving data.

When a policy regarding the making of back-up copies of data is adopted two things need to be considered; a) the likelihood of the primary data being lost, and b) the time spent by the user in carrying out the policy. Clearly, if people are spending a

disproportionate amount of time backing up data which is regarded as being already 'safe' the policy is too strict. If people spend no time backing up data which is not securely held, the policy is too lax.

A standard back-up policy requires three copies of important data to be made. The first copy is the primary (i.e. working) data. The second is a back-up which is stored on-site. The third copy is a back-up which is regularly transported off-site. Such an 'ideal' policy may not be appropriate for MRI data, simply because of the time that would be required to copy such large datasets to external locations on a regular basis.

Some points to remember:

1. Hard drives can, and do, fail.
2. HARD DRIVES CAN, AND DO, FAIL!
3. High-specification computers/laptops like the ones we use are an attractive target for thieves.
4. 'rm \*' is a very easy command to type accidentally in a unix terminal.

**It is up to individual users to ensure their data is secure.** The following policy is simply a suggestion.

1. All data currently being worked on should be stored on the user's hard-drive.
2. The Cortex file-server and the external hard-drives available in the lab should be used for making temporary back-ups of data currently being worked on. Backing-up to these sources once or twice a week would seem to be an acceptable level.
3. Once a project is completed, the data should be archived to DVD-R. The raw (DICOM) data should always be archived, and it's probably worth archiving the final-stage processed data. It may or may not be worth archiving all the inter-mediate stages (i.e. re-aligned, normalised etc.) of the processed data depending on the size of the data-set. If the data is to be deleted off the hard-drives after archiving, two copies on DVD-R should be made and one transported off-site.

If anyone finds that they are having trouble managing the volume of their data (i.e. their hard-drive is too small, backing-up to external drives takes an inordinate amount of their time) please talk to Andy or Matt and another solution can probably be found.

## 22. Other Useful Resources

There is now a wealth of information available on the internet about functional imaging. The most useful sites are listed below, but this list is definitely not exhaustive. Sometimes Google can provide the answer to a question surprisingly readily!

The SPM website:

<http://www.fil.ion.ucl.ac.uk/spm/>

The SPM e-mail list archive:

<http://www.jiscmail.ac.uk/lists/spm.html>

The MRC-CBU imagers homepage – loads of good resources on SPM and imaging in general here:

<http://www.mrc-cbu.cam.ac.uk/Imaging/Common/>

The FSL website:

<http://www.fmrib.ox.ac.uk/fsl/>

The mri3dX website:

<http://www.aston.ac.uk/lhs/staff/singhkd/mri3dX/>

mrGray and mrFlatMesh information and documentation:

<http://info-center.ccit.arizona.edu/~cni/mrgray.htm>

Excellent glossary of imaging terms available here:

<http://info-center.ccit.arizona.edu/~cni/glossary.htm>

And most importantly: if you get stuck, just ask someone!