



IDL_SPM99 USER GUIDE

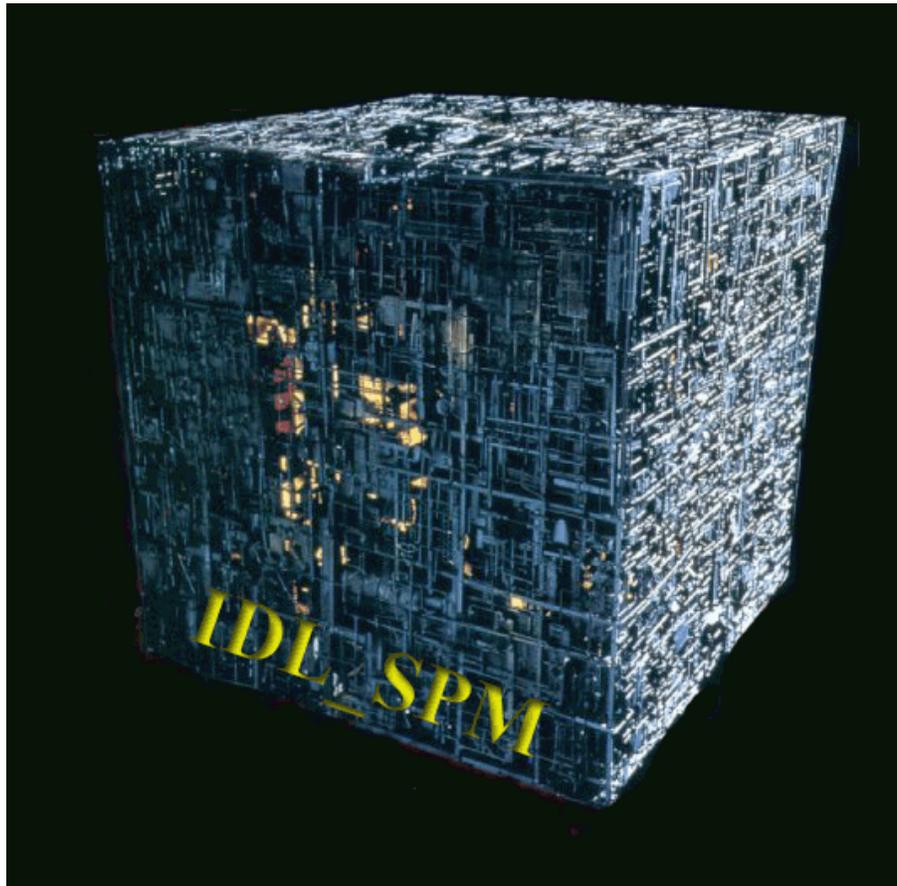




TABLE OF CONTENTS

IDL_SPM99.....	4
I. GETTING STARTED.....	4
II. QUICK START.....	4
III. SETTING UP THE REFERENCE PARADIGMS.....	5
BLOCK DESIGNS.....	6
EVENT-RELATED DESIGNS.....	6
FIXED ISI.....	7
PSEUDO-RANDOM DESIGNS.....	7
Straight vector definition.....	8
MIXED DESIGNS.....	8
Mixed epoch-event.....	8
IV. ANALYSIS OVERVIEW.....	9
A. LOAD AND RECONSTRUCT DATA.....	9
B. SETTING UP AND RUNNING ANALYSIS.....	10
Defining the reference paradigm.....	11
Setting up analysis parameters.....	12
Filtering options:.....	13
Adding Confounds/Regressors.....	13
Realignment Parameters.....	13
Smoothing Parameters.....	14
Running the Data.....	14
C. BATCH CONTROL.....	15
Recovering from an error.....	15
Killing Batch jobs.....	16
Batch Diagnostics.....	16
V. GROUP ANALYSES.....	17
SETTING UP GROUP ANALYSIS PARAMETERS.....	17
VI. VIEWING THE DATA.....	19
Cluster level correction.....	24
Map level correction.....	24
ROI Info.....	24
3D DISPLAY.....	25
MULTIPLANAR DISPLAY.....	25
MINISLICER.....	26
SAVING IMAGES.....	27
Snapshot.....	27
Print.....	27
Masking the data.....	28
Loading a mask.....	28
Shortcut.....	28
Radiologic Viewing.....	28
VII. AUTOMATED ROI EXTRACTION AND ANALYSIS.....	28



VIII. VOXEL BASED MORPHETRY (VBM) ANALYSIS.....	29
IX. INDEPENDENT COMPONENT ANALYSIS (ICA)	30
X. FUNCTIONAL CONNECTIVITY ANALYSIS.....	30
XI. MNI SPACE, TALAIRACH SPACE, and BRODMANN AREAS	31
XII. ATLAS DATA	31
XIII. WHAT IF I DON'T HAVE GE DATA.....	31
Viewing.....	31
Running the analysis with DICOM data.....	31
Running the analysis on non-GE data without DICOM data	33
XIV. POWER USER	34
Some Header Routines.....	34
Some Registration Routines.....	34
Some image conversion and viewing routines.....	35
APPENDIX I (INSTALLATION).....	37
System Requirements.....	37
Software Download Site	37
Some Preliminaries for a smooth install before you start	37
Configuring the main program directory	37
Upgrading to the latest release.....	38
Creating the ref_functions directory, cpufile, and job_control files.....	38
Configuring Users, IDL, and Matlab	39
IDL_SPM wizard.....	39
Wizard user configuration (create_user).....	39
Manually configuring the .idlspm site directory.....	40
Manually configuring user .idlspm directories	40
Testing your installation	41
Configuring Additional Users.....	42
Tuning up the batch processing	42
Upgrading to Matlab6, IDL 5.X and Beyond	42
Changing the Location of the appl folder	43
SPM96.....	43
APPENDIX II (Data Acquisition and Archival)	45
Archiving fMRI Data.....	45
Running fmriepi:.....	45
Archiving the anatomic images	47
APPENDIX III (Directory Structure and File Naming Conventions).....	48
DIRECTORY STRUCTURE	48
FOLDER CONTENTS	49
FILE NAMING CONVENTION	49
APPENDIX IV (Archiving to CD).....	50
Program Files.....	50
CD archiving information.....	50
Some useful scripts	52
SOLARIS/SUN issues	53
APPENDIX V (Describing the methods for a paper).....	54
APPENDIX VI (FAQ).....	56



IDL_SPM99 J. Maldjian, MD

This version incorporates the full release of SPM99, LX compatibility, batch mode with multiprocessor support and job queing, an interactive anatomic atlas with MNI to true Talairach conversion, Brodmann Area labels, automated roi extraction, multiplanar viewing, and cross-platform compatibility (Linux/Sparc).

I. GETTING STARTED

In order to use IDL_SPM, certain environment variables, and paths have to be set properly. You will need to setup IDL, Matlab, and the location for the IDL_SPM programs. This should be done by the system administrator (see Appendix I). This release makes extensive use of batch mode in SPM99. It will utilize multiple cpus on the network simultaneously to dramatically decrease overall processing time, especially for multisubject jobs. This is accomplished by breaking up each analysis into smaller batch jobs, and sending these small batch jobs to available cpus. Proper configuration is critical for this to work smoothly. This manual describes *some* of the capabilities of IDL_SPM. It is not meant to be exhaustive, but it should give the user sufficient knowledge to run a wide variety of analyses.

II. QUICK START

An analysis can be setup within IDL_SPM in less than 30 seconds (assuming you were running fmriarch while you were scanning):

1. Open a command window and enter:

idlspm

Or you can launch from within IDL:

Start idl and enter:

@bootfmri

@bootspm

2. Click on SETUP SUBJECTS

click on ADD

select one of the .im files from your subject DATA directory

click on DONE

click on CYCLE ALL SUBJECT REFS

select the appropriate reference function for each paradigm

click on DONE

Enter some information for each RUN in the paradigm text box

Click on DONE

Supply the patients last name when prompted for a filename

3. When the SPM99 ANALYSIS widget comes up, if you are happy with the default settings, click on DONE (you may optionally want to click on REALIGN_OPTIONS and set spm internormalization to allow a direct epi to epi talairach normalization)

4. You can leave at this point, however it is better if you exit IDL after the batch scripts have been created so that you don't waste a cpu.



III. SETTING UP THE REFERENCE PARADIGMS

One of the new features in IDL_SPM is robust paradigm definitions, including both epoch (block) and event-related designs. The reference files are text files which can be edited using any text editor. Sample reference files are located in the /appl/IDL_SPM97/spm97/sample_ref subdirectory. The structure of the reference file is simple. At WFU, the reference subdirectory is /fmri2/fmri/ref_functions . The reference functions are composed of a short text header (defining the TR, number of images, number of conditions, type of paradigm) and the actual paradigm definition (indicated by begin and end). Semicolons before any text are comments (or commented out). Long lines of text can be continued onto subsequent lines by using the \$ symbol. All reference files have the following syntax:

```
;comments
;comments

paradigm_name :      paradigm name (right hand motor)
tr             :      repeat time (in seconds)
images        :      total number of images per run
conditions    :      number of conditions (baseline always gets condition # 1!)
ER            :      event-related flag (1 or 0)
fixed         :      fixed isi (1 or 0) - only if event-related
mixed        :      mixed flag (0 or 1) for mixed epoch and event paradigms
names         :      condition names
contrast_names :      contrast names (for your output)
contrast_types :      (T or F)
contrast_values :      contrast vectors for your contrast_names

begin
paradigm definition
end
```



BLOCK DESIGNS

For a blocked 30 second on, 30 second off paradigm, with the start off, a tr of 3 seconds, and 100 images (5 minutes of imaging), the reference file would look like:

```
tr:          3
conditions:  2
images :     100
ER:         0
fixed:      0
names :      baseline,finger_tap
contrast_names: motor_on,motor_off
contrast_types :      T,T
contrast_values: [-1,1],[1,-1]
```

```
begin
  blocks      ,      1,2,1,2,1,2,1,2,1,2
  time        ,      30
```

end

Since the block time is the same for the on and off periods, the shorthand of entering only 1 value for the time condition is OK. The paradigm portion of the file could have been equivalently written:

```
begin
  blocks      ,      1, 2, 1, 2, 1, 2, 1, 2, 1, 2
  time        ,      30,30,30,30,30,30,30,30,30,30
```

end

If there were multiple conditions, let's say swallow, smell, and baseline, with swallow for 20 seconds, followed by baseline for 20 seconds, followed by smell for 15 seconds followed by baseline for 25 seconds, and a tr of 2 (160 images, 5 minutes of imaging) the header would look like:

```
tr:          2
images: 160
conditions:  3
ER:         0
fixed:      0
names :      baseline,swallow,smell
contrast_names: swallow-baseline,smell-baseline,$
                swallow - smell
contrast_types :      T,T,T
contrast_values: [-1,1,0],[-1,0,1],[0,1,-1]
```

```
begin
  blocks      ,      3, 1, 2, 1, 3, 1, 2, 1
  time        ,      20, 20, 15, 25, 20, 20, 15, 25
```

end

Note that the baseline condition **ALWAYS** gets assigned the number 1, and is the first condition name listed after **names**, no matter when it occurs in the paradigm. In this example, the baseline condition occurred after the smell condition (look at the blocks definition: 3,1,...).

EVENT-RELATED DESIGNS

In an event-related design, the stimulus is delivered for a very brief period of time (ie ≤ 1 TR). The stimulus can be repeated at fixed intervals (the ISI, or inter-stimulus interval), or can be delivered in a pseudo-random order (pseudo because obviously the experimenter must know the order).



FIXED ISI

Let's say our experiment is a simple sensory stimulus, with the buzzer on for 1 second each time the stimulus is delivered. Again, the TR = 2, images = 160. For this experiment, let's say the first time the stimulus is delivered is 14 seconds into the paradigm, and it is then repeated, at fixed intervals every 20 seconds. The reference function would look like:

```

tr:          2
images:     160
conditions: 1
ER:         1
fixed:      1
names:      buzzer
contrast_names: buzzer_on
contrast_types: T
contrast_values: [1]

begin
  1      ,      14
  isi    ,      20
end

```

Note that we have set the ER flag (for event-related) in the header, as well as the fixed flag, indicating a fixed inter-stimulus interval. Now let's say we have a buzzer which occurs 14 seconds into the paradigm, and is repeated every 20 seconds, and a light that goes on at 5 seconds into the experiment, and is repeated every 13 seconds. The reference function would look like:

```

tr:          2
images:     160
conditions:  2
ER:         1
fixed:      1
names:      buzzer,light
contrast_names: buzzer_on,light_on,buzzer-light
contrast_types: T,T,T
contrast_values: [1,0],[0,1],[1,-1]

begin
  1      ,      14
  isi    ,      20
  2      ,      5
  isi    ,      13
end

```

Note that the ISI for each stimulus must follow the stimulus definition, but we can have as many conditions and stimuli as we like for each experiment.

PSEUDO-RANDOM DESIGNS

In an event-related design the use of a pseudo-random order is a powerful method to control for confounds within the experiment related to the subject's expectations or anticipation of an upcoming stimulus. Let's say we have a buzzer which goes on at the following times in our experiment: 3 seconds, 15,21, 40, 100, 200,210, and a light which goes on at the following intervals: 10,20,40,80,140,190

```

tr:          2
images:     160
conditions:  2
ER:         1
fixed:      0
names:      buzzer,light

```



```

contrast_names: buzzer_on,light_on,buzzer-light
contrast_types: T,T,T
contrast_values: [1,0], [0,1],[1,-1]

```

```

begin
  1      ,      3, 15,21, 40, 100, 200,210
  2      ,      10,20,40,80,140,190
end

```

Straight vector definition

The final way of defining the paradigm can be used for setting up bizarre paradigms which are a mix of epoch and event-related designs. The paradigm definition portion of the file just lists the condition, and when it occurs in the paradigm. This method is not terribly useful for defining paradigms, however, this can be used to define confounds (not fully implemented in this release). If you are interested in confound/regressor definitions, check the section on adding confounds/regressors in the SETTING UP AND RUNNING ANALYSIS section.

MIXED DESIGNS

Mixed designs are comprised of epoch conditions and event conditions. This is a new feature of SPM99. Mixed designs include the mixed field in the header definition. Epoch conditions MUST come first, then the event conditions.

Mixed epoch-event

In this example, the blocks will be 30 seconds on, 30 seconds off, with the start off, a tr of 3 seconds, and 100 images (5 minutes of imaging). Interspersed with this will be a third condition, a single-event type condition, in which a buzzer goes off at time points of 12, 40, and 55 seconds. The reference file would look like:

```

tr:          3
conditions:  3
images :     100
ER:          1
fixed:       0
mixed:       1
names :      baseline,finger_tap,buzzer
contrast_names: motor_on,motor_off
contrast_types :      T,T
contrast_values: [-1,1],[1,-1]

begin
  blocks      ,      1,2,1,2,1,2,1,2,1,2
  time        ,      30
  3           ,      12, 40, 55
end

```

Note that we have set the ER flag (for event-related) in the header, as well as the mixed flag. When setting up the analysis for this data set, both the epoch widget box selections, and the single-event widget box selections will be available.



IV. ANALYSIS OVERVIEW

The steps to be followed for a single subject analysis are summarized below:

1. Load and reconstruct data
2. SETUP SUBJECT and reference paradigm
3. RUN MULTIPROCESS program

During the analysis the program will automatically perform the following steps:

- 1) Normalize the T1 data set to Talairach
- 2) Motion correct the epi data
- 3) apply the normalization in step 1 to the motion corrected data
- 4) execute an inter-run realignment for successive runs within the same subject.
- 5) Optionally perform a second direct epi to epi Talairach template normalization
- 6) Generate the SPMS for each run

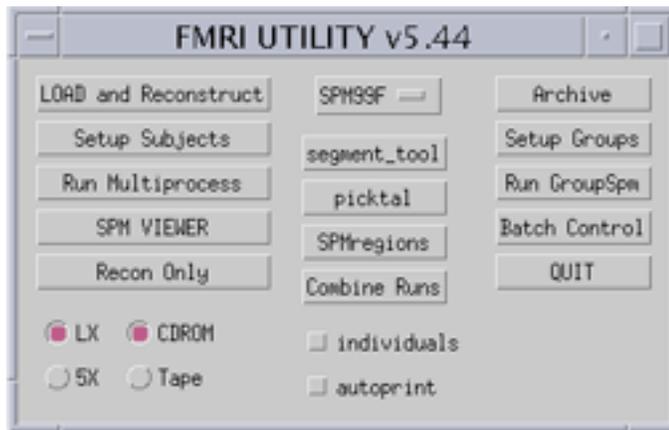
To compile the programs necessary for IDL_SPM, first, start idl. Then type:

@bootfmri

@bootspm

The programs should compile without errors, and the fmri_utility widget should pop open. If this is not the case, check with your system administrator, or read through the appendix on setting up IDL_SPM.

All functions are accessible from the fmri_utility widget:



This widget should pop open after you have compiled the programs. You can also bring it up by typing data_control.

A. LOAD AND RECONSTRUCT DATA

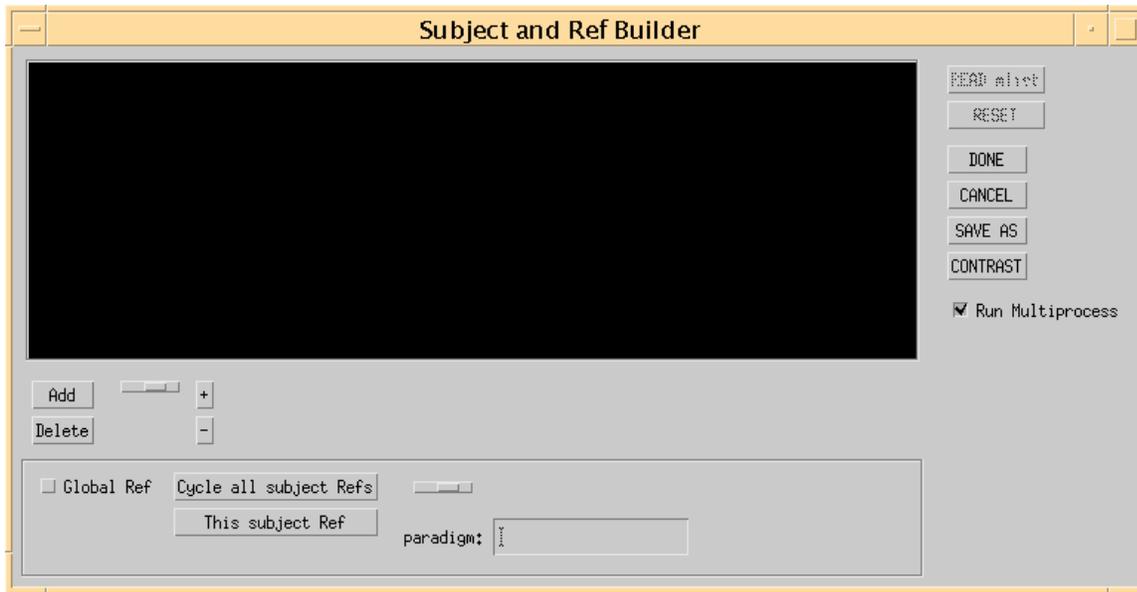
If you are using the autoftp and autorecon scripts (provided in /appl/IDL_SPM97/LX/LX_RECON) then you can skip this step. Put your tape in the DAT drive (remember to slide the write-protect tab), or your CD in the CD-drive. Make sure to click on CD if you have archived to CD. For tape archives, although the recon can autoroute between LX and 5X, the tape loader cannot (they use different blocking factors). You must select 5X or LX on the check box now for tapes. After selecting your platform, click on Load and reconstruct.



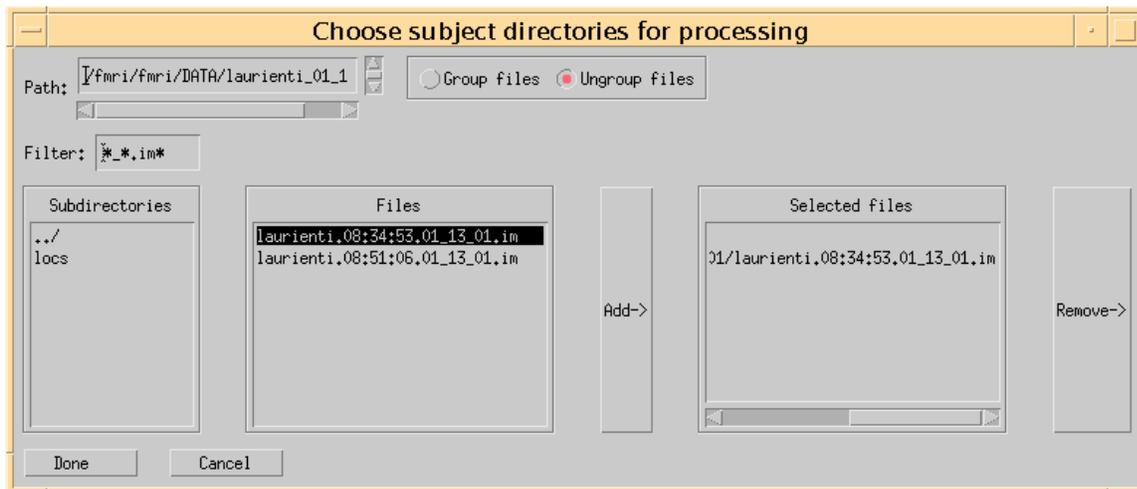
Select the data directory which will hold this subject's data. Make sure you have enough space in this directory to hold all the data (Each run can take up 40-100 Mb). The program will prompt you to enter a name to label the subject data. Some caveats: 1) don't use a name that already exists in your directory; 2) **don't put any spaces, periods, or backslashes in the name.**

B. SETTING UP AND RUNNING ANALYSIS

IDL_SPM is designed to handle multiple subject jobs. Click on SETUP SUBJECTS. A subject selector widget will pop-up:



In order to add subjects to the list, click on the add button. A multipickfile widget will open:



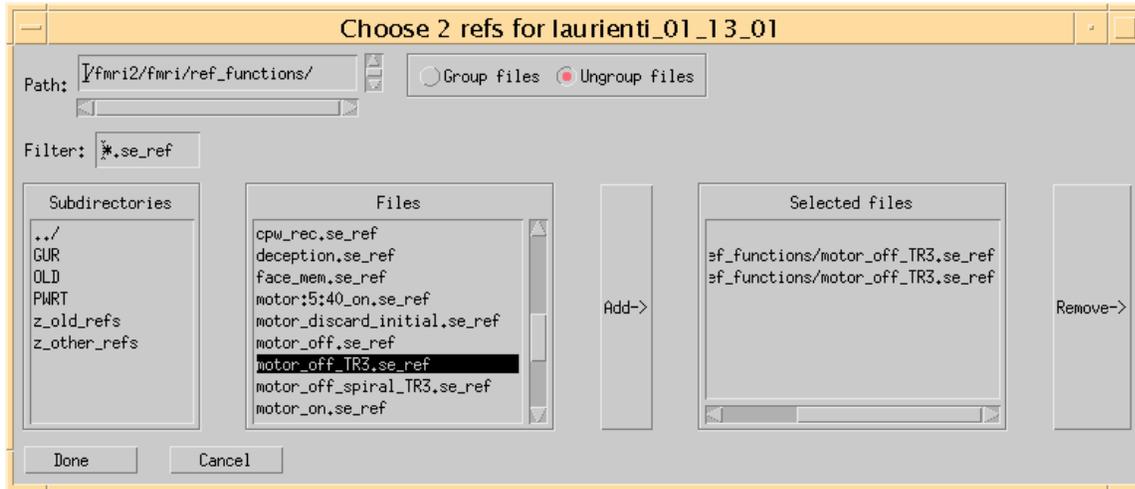
From here you can enter all the subjects that you want to analyze. Make sure to click on one of the .im files in the subject directory you want to use. You don't have to click on all of



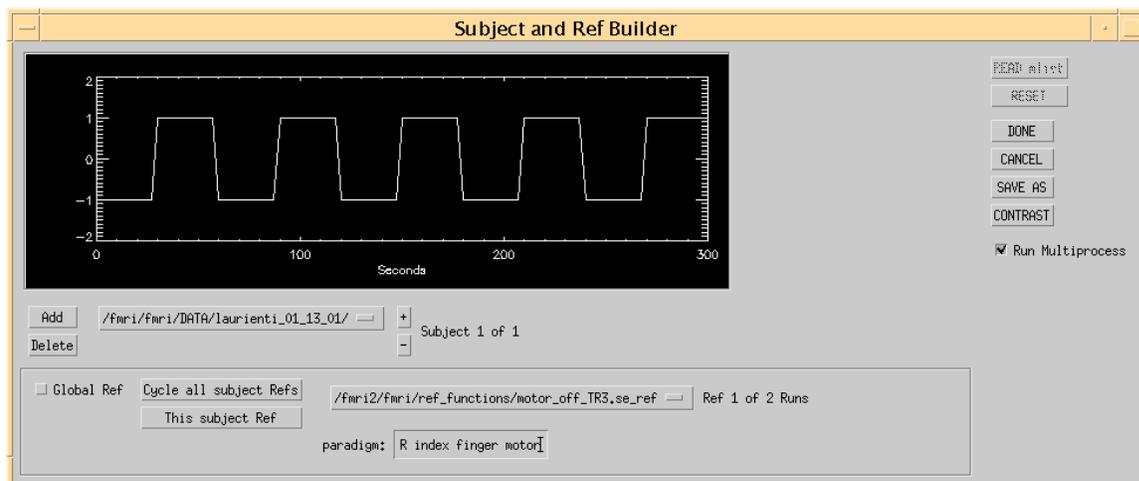
them. Just 1 will do. You can keep adding subjects while in this widget. If you make a mistake, use the Remove button on the widget. When you are done, if you want to add more subjects, just click on the add button again.

Defining the reference paradigm

After selecting all the subjects to be analyzed, click on <Cycle all subject refs>. The multipickfile will open again:



It will prompt you to enter as many ref files as you have runs for all your subjects: If all the refs are the same, click on the <global ref> button before clicking on the <cycle all subject refs> button. If you don't have a reference paradigm for your analysis already defined, see the section on **SETTING UP REFERENCE PARADIGM** earlier. If there is something wrong with your reference paradigm for your data sets, an error message will pop up indicating the problem reference function. When the reference functions have been successfully selected, the reference paradigms can be viewed in the plot window simply by selecting them from the widget list:



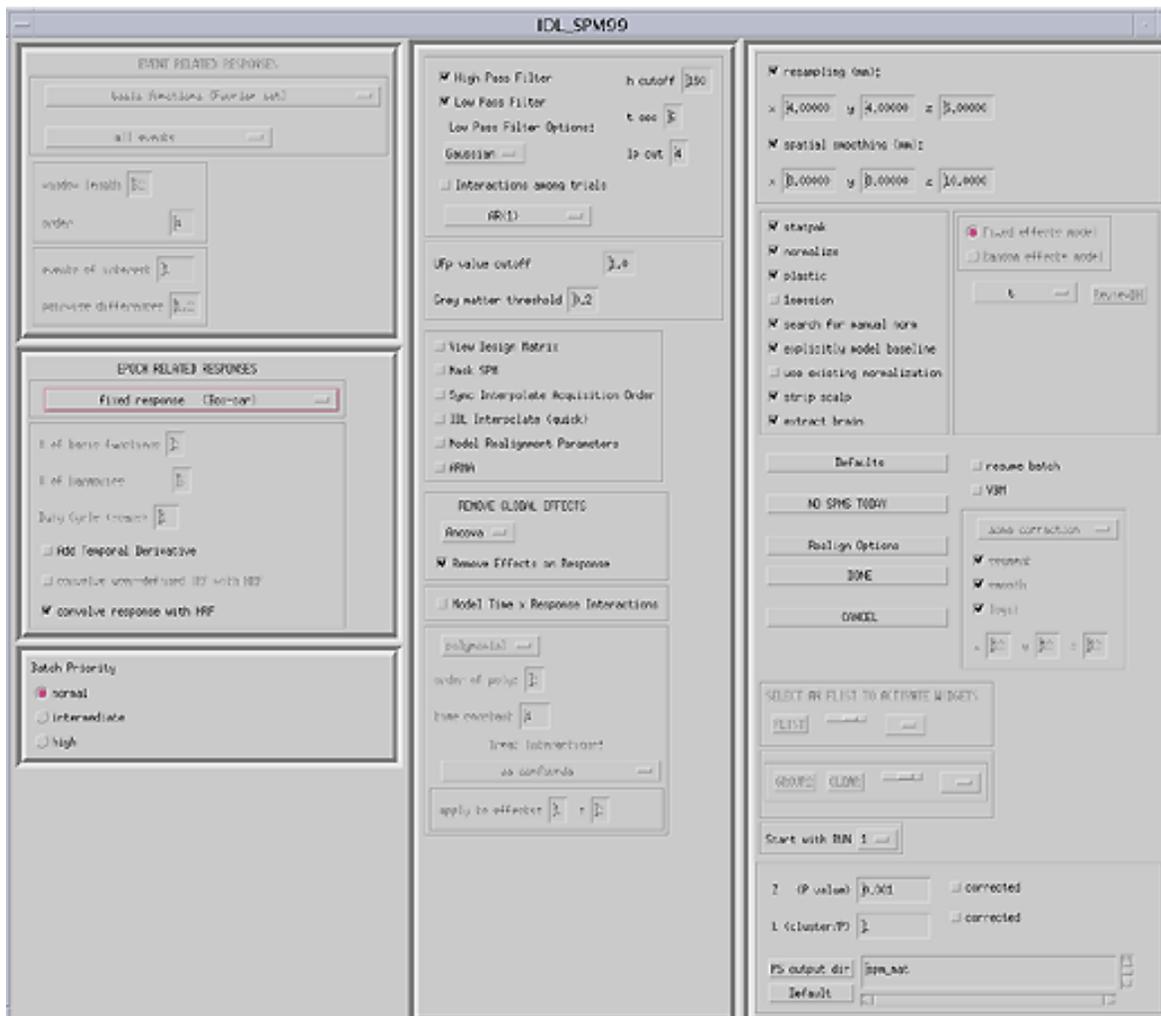
You should now enter some text in the "paradigm:" text box at the bottom of the widget for each paradigm describing this particular run. You can go to the next paradigm within a subject by hitting return. You can go to successive subjects using the drop list under the plot window, or by clicking the + or - button. When the reference paradigms have been entered and



all the subjects have been setup, click on the <DONE> button. You will be prompted to enter a filename to enter the list. If this is for a single subject, just enter the subject's name and select OK. If this is for multiple subjects, enter a name you want, and keep track of what directory you are putting it in. The program defaults to the location of the last subject entered. This file is also known as the "multiprocess" file, or the subject "flist"(filelist).

Setting up analysis parameters

The program will automatically bring up the analysis widget. You can also get here by clicking on the <Run Multiprocess> button (you will be prompted for the multiprocess file you created above in this case). For epoch-related analyses, the program defaults to a box-car analysis, with high-pass filter, F cutoff = 1, and grey matter cutoff = 0.2 . You can set any of the parameters you like, similar to in native SPM. For an event-related design, the program defaults to fourier basis functions. Note that you can get very significant differences in your results with different models. Sorry, I can't tell you what will work best. Gamma functions with derivatives seems to work well for event-related designs.



The UFP value sets an initial F-test cutoff. The default of 1.0 lets all voxels pass through. Native SPM96 and SPM97 use defaults of 0.001 for this. This throws out a lot of voxels, that may be interesting. SPM98 will be using a default of 1.0 . I suggest you use the default of 1.0 . The grey matter cutoff (gm) sets a threshold % signal intensity to determine if voxels should be included in the analysis. Native SPM96 and SPM97 use a default of 0.8 . If you want to include



all voxels, use a value of 0 . The problem however, is that you will be including voxels outside the brain, and making your p-values worse than they should be by including all these extra voxels. The default of 0.2 works well.

mask spm	Mask the background voxels
sinc interpolate acquisition order	correct for slice timing
IDL Interpolate	exactly same, but much faster
model realignment parameters	puts realignment parameters in DM

Filtering options:

High pass filter and click box

Note that Low pass filter is governed by Low Pass filter options drop list (not the check box), and the cutoff is in the lpcut box next to it. The low pass filter box above it is a throwback from SPM96/97 which I haven't gotten rid of yet.

AR(1) for determining intrinsic autocorrelation.

Adding Confounds/Regressors

Confounds can be added using a .confound file. Look at the /spm97/sample_refs directory for a description. The definition is similar to ref functions. You need to indicate the imaging TR, number of images, and number of confounds. Following this, there is a specific confound section defining the sampling interval for your regressor, and the values. One confound file can contain multiple confounds, defined one after another. Any .confound file found in the RUN folder or spm_mat subfolder for a particular RUN will be incorporated into the analysis. Put them there before hitting the DONE button so that the batch files can incorporate them.

```
tr          :          2
images     :          160
confounds  :           1
```

```
begin
;-----
;confound (leave 0 for sampling interval if it is the same as tr)
;-----
name      ,          my_regressor1
sampling interval , 0
1
4
6.4
10
12
14
20

end
```

Realignment Parameters

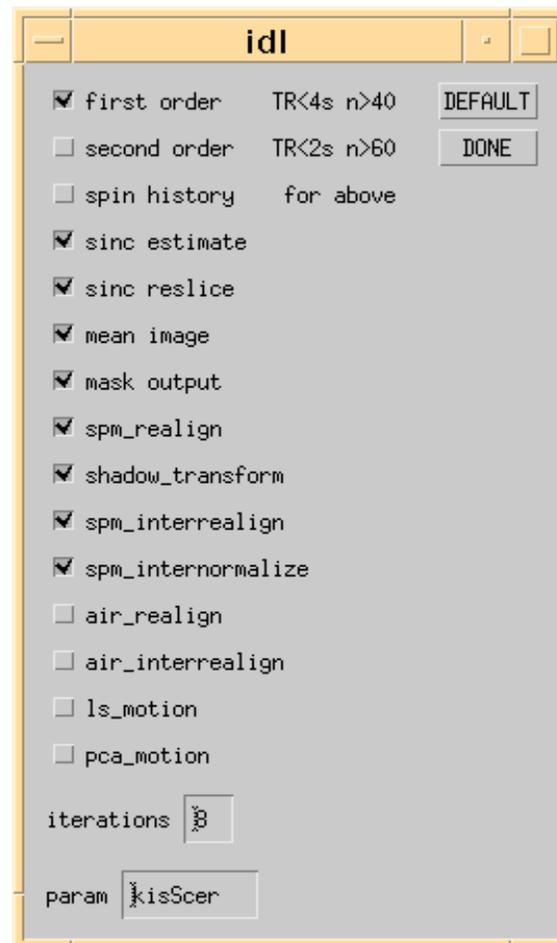
If you want to change (or view) any of the realignment defaults, just click on <REALIGN OPTIONS> . The selections should be intuitively obvious. You may want to add spm_internormalize if you have acquired the entire brain with your epi sequence. This will apply a second normalization step of the normalized epi images directly to the SPM epi



Talairach template. If you did not acquire the entire brain, this additional normalization will probably distort your data.

Smoothing Parameters

Smoothing FWHM sizes can be set at the top right. The default is 2X voxel size. If some of your data was scanned at different voxel dimensions, your output cubes, although normalized, may be of different dimensions. In this case you may want to specify the resampling size prior to smoothing to maintain constant output dimensions. The smoothing options include Gaussian (native SPM), and a Hamming filter (filter size defaults to 64x64x64).



Running the Data

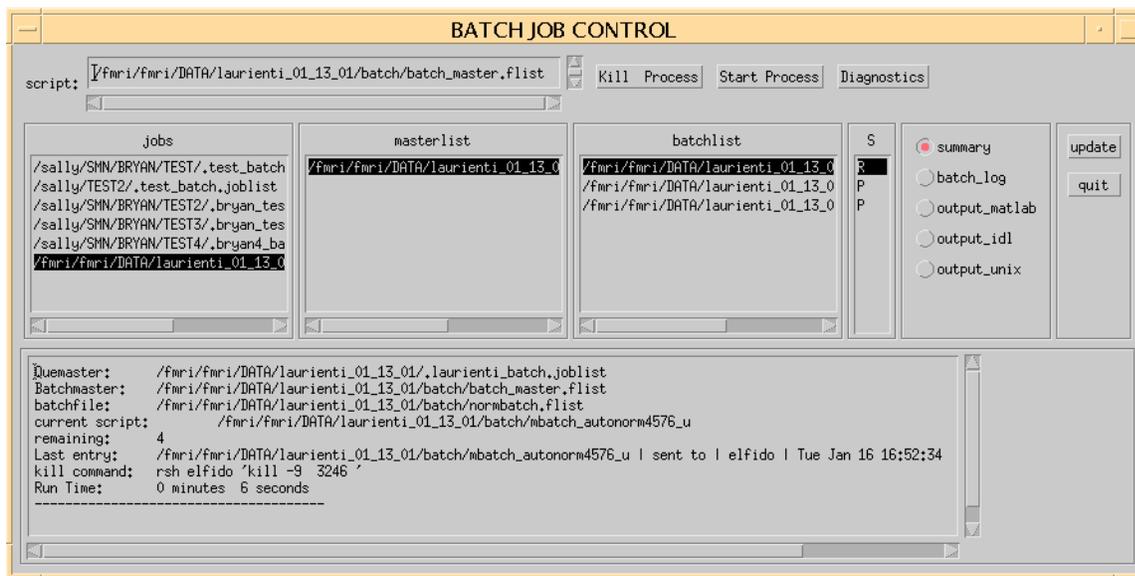
When you are satisfied with the analysis options, hit the DONE button. Batch mode will be in effect for the processing of your data. Batch scripts will be created for each of your subjects, and all runs. The batch scripts will be located in the batch subfolder of each subject. Within the batch folder there will be RUN folders which follow the same naming convention as for your actual DATA. Once the batch scripts are created the batch_control widget will pop



open. It is recommended that you exit idl at this stage, since by leaving it running, you are unnecessarily soaking up cpu resources which could be used for the analysis.

C. BATCH CONTROL

The `batch_control` widget will pop open after the batch scripts are created, or it can be accessed directly via the `fmri_utility` widget. This is the batch command center. You can check on your batch jobs, kill batch jobs, restart batch jobs, and run batch diagnostics from the `batch_control` widget:



The **jobs** window at the left lists all the batch jobs that have been submitted. This can be cleared of old jobs by erasing them from the `job_control` file (located in the `ref_functions` directory). The **masterlist** column lists the subjects master lists for the selected jobs (in this case, we only ran 1 subject). The **batchlist** column lists the batch jobs for this subject. The **Status** column next to it lists the status of that job (P = pending, R = running, C = completed, E = Error). The **button list** next to this column are the various types of output from each job. If there is an error, the `output_matlab` text can be very useful to figure out what happened. These output files are all located in the `/batch/RUN` subdirectory for that job. When jobs are completed, or when errors occur, email notifications will go out to the email address listed in your `.idlspm/global.ksh` file.

Recovering from an error

If an error occurred, the batch logs can be viewed to figure out what caused it. Often these errors result from file naming problems or lack of disk space. The batch is designed so that when an error condition occurs, the batch job can recover without having to repeat all the prior steps. The batch can be reinitiated in one of 2 ways:

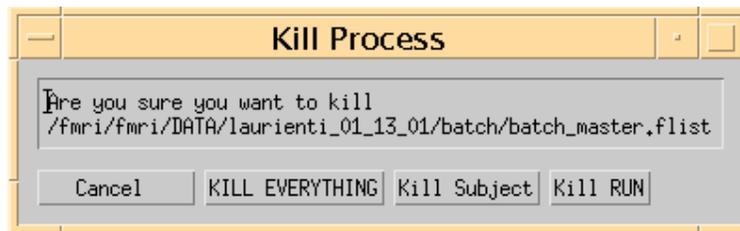
From the `batch_control` widget, highlight the batch job to restart, then click on Start Process. The second way is directly from the `batch/RUN` folder. Within each batch and `RUN`



folder, there is a file called quick_start. CD to the RUN folder from an xterminal and type quick_start. This will reinitiate the batch job for that RUN directory only. Note that if your error was due to a lack of disk space, you should reinitiate the entire batch for those RUNS (because the files were likely partially written and may be corrupted). You can make the batch redo previous steps by reloading the batch_que file with the scripts located in the batch.flist file in that RUN's folder.

Killing Batch jobs

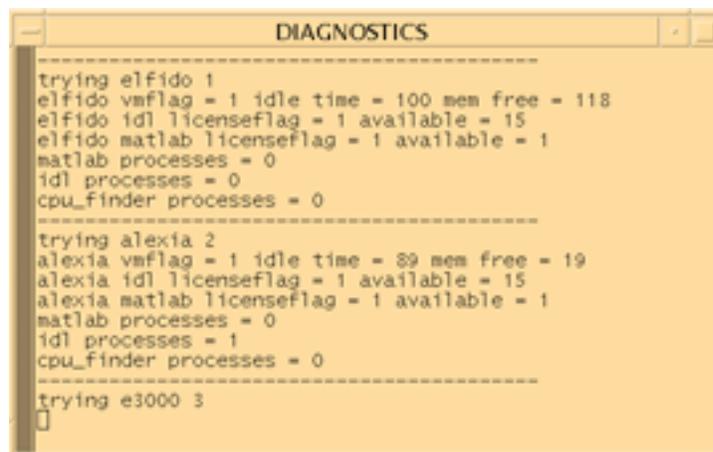
Sometimes it is necessary to kill batch jobs, for instance, when you decide to change the analysis parameters. This can be accomplished by highlighting the job you want to kill, and clicking on Kill Process:



This will give you the option of killing the entire job (all subjects), killing at the subject level (all RUNS), or just killing the highlighted RUN. An alternative way to kill a batch job is by creating a file called kill in the batch /RUN folder that you want to terminate.

Batch Diagnostics

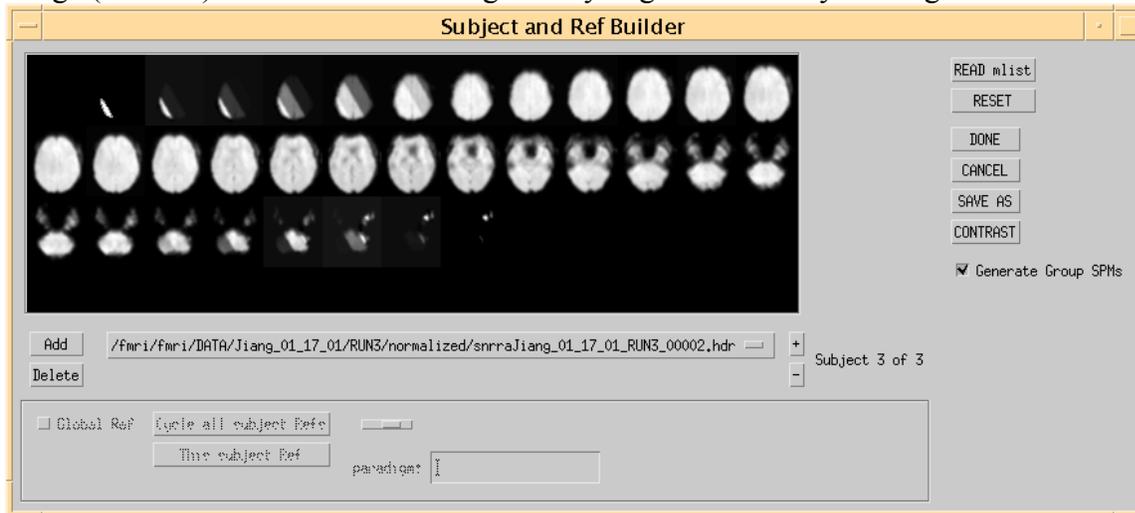
The batch diagnostics can be useful for diagnosing batch related problems. It will pop open an xterminal window and cycle through all the cpus in your cpu list and check for idl, matlab, and cpufinder processes. If you get errors here, there is something wrong with your batch setup (.rhosts file, .idlspm configuration files, etc). vmflag = 1 means the cpu is available. The output should look like this:





V. GROUP ANALYSES

In order to perform a group analyses, it is assumed that you have done the individual analyses on each subject already. Click on <SETUP GROUPS> . This will bring you to the familiar Subject selector widget. When you click on the <ADD> button, you will be prompted to select the actual normalized image file for the subject run to be included in the group. When prompted, go to the RUNx/normalized subdirectory of the subject and Run you want to add to the group. You should see at least 3 files in this directory. These are labeled mean.hdr, nrxxxx.hdr, and snrxxxx.hdr . Click on the snrxxxx.hdr file . n stands for normalized, r stands for realigned, s stands for smoothed . If this is the first run, it will be nrxxx or snrxxx . Subsequent runs have an additional r (nrxxxx or snrxxxx) because of the additional inter-run realignment . If you want to change the smoothing that was applied, click on the non-smoothed image (nrxxxx) and set the smoothing when you get to the analysis widget.

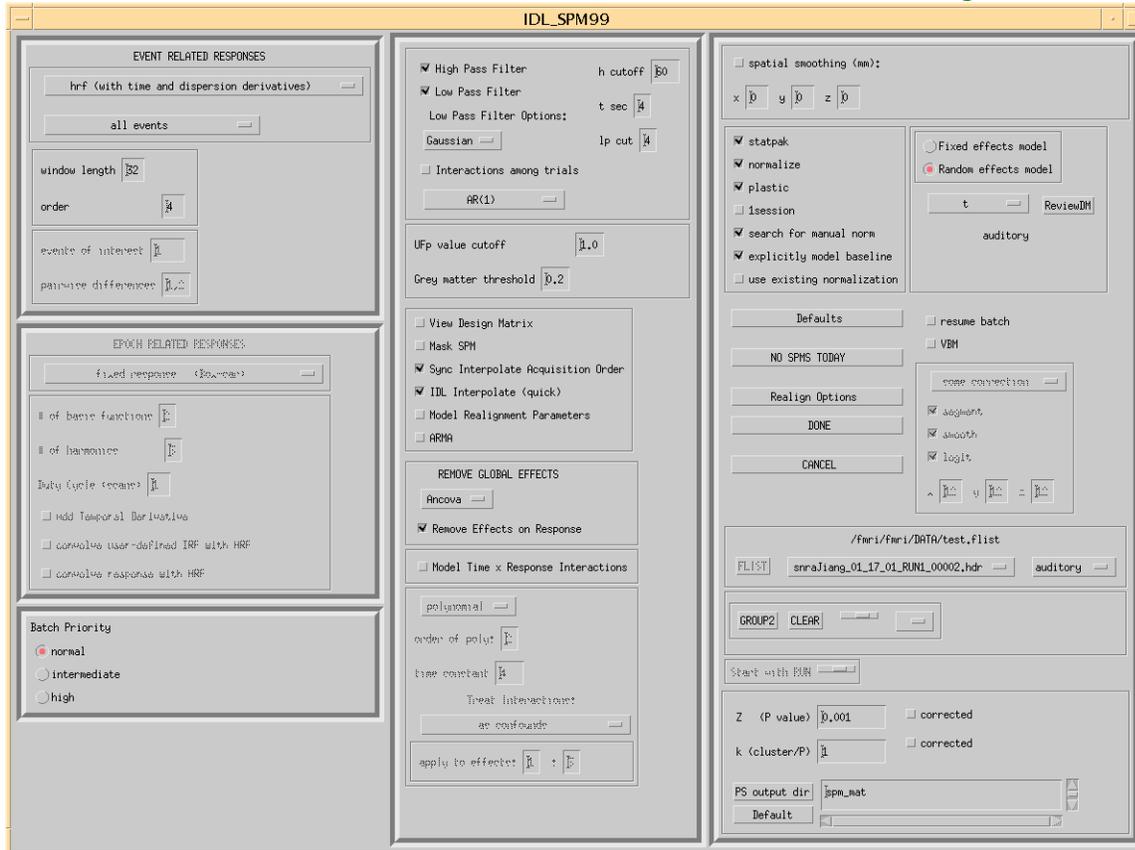


You can select all the desired runs while in the multipickfile widget, or you can add and delete files afterward. You can view the quality of the realignment within the display window simply by clicking on the appropriate subject run. If the data doesn't line up with the other runs, delete it from the analysis.

When you have selected all the subject runs, click on the <DONE> button. You will be prompted to enter a file name to store the list of subject runs. Remember where you put this file. You will also be prompted to select the directory to store the final maps. If you don't already have a unique directory to store the data, make one now. There are a lot of files created, and you want them all to be in a single directory, without other crap in it.

SETTING UP GROUP ANALYSIS PARAMETERS

The familiar analysis widget will pop up. You have an important decision to make now. You can use a fixed-effects model, or a random-effects model. If you are planning to make population inferences (which is what most people want to do with group data), you need to use a random effects model. If you use the fixed-effects model, this will implement the SPM96 style analysis (Note, if you want to concatenate runs instead, you can use the "COMBINE RUNS" from the fmri utility widget).



With the random effects model, the program implements the SPM99 random effects analysis. This is essentially a 1 sample T-test on the conxx.images corresponding to the contrast you are interested in. The con images collapse each individual subject fMRI run to a single image. Although your flist specified the snr images, the program will automatically find the proper conxx.img images to go into the random effects analysis. There is a selector beneath the random effects button which says t, F, BETA, and user_defined. The default is T. This will perform the SPM99 random effects analysis and automatically locate the appropriate con images. If you want to use some other images in a random effects analysis, then set it to user_defined. This will perform the 1-sample t test on the specific images defined in the group flist (the images in the group list must be analyze type .img images). Don't forget to select the contrast you are interested in (selector is on bottom right third of widget).

If you want to compare 2 groups (like men vs women) using the random effects model, click on the <GROUP 2> button, and select the list corresponding to this group (this assumes you have already created it). This will implement a 2-sample T test in SPM99 on your groups. Don't forget to select the contrast for group 2 using the contrast selector which will appear on the bottom third of the widget.



VI. VIEWING THE DATA

Click on <SPM VIEWER> from the fmri utility widget. You will be prompted to select the SPM.mat file corresponding to the data you want to view. For individual subject runs, you will find this file in the spm_mat subdirectory. Just click on it and hit OK.

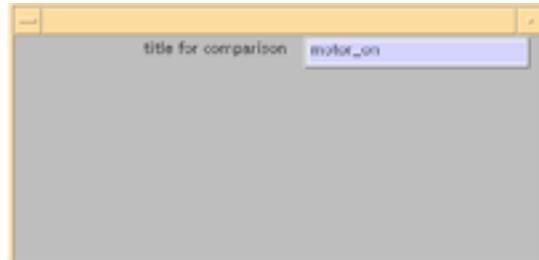


The program will then bring up the SPM99 contrast manager. The contrasts you are interested in should all be listed if you constructed your reference function properly. This is being displayed through matlab spm99, so you can add more contrasts from here, just like in spm99 if you like. Select the contrast you want and click on DONE:

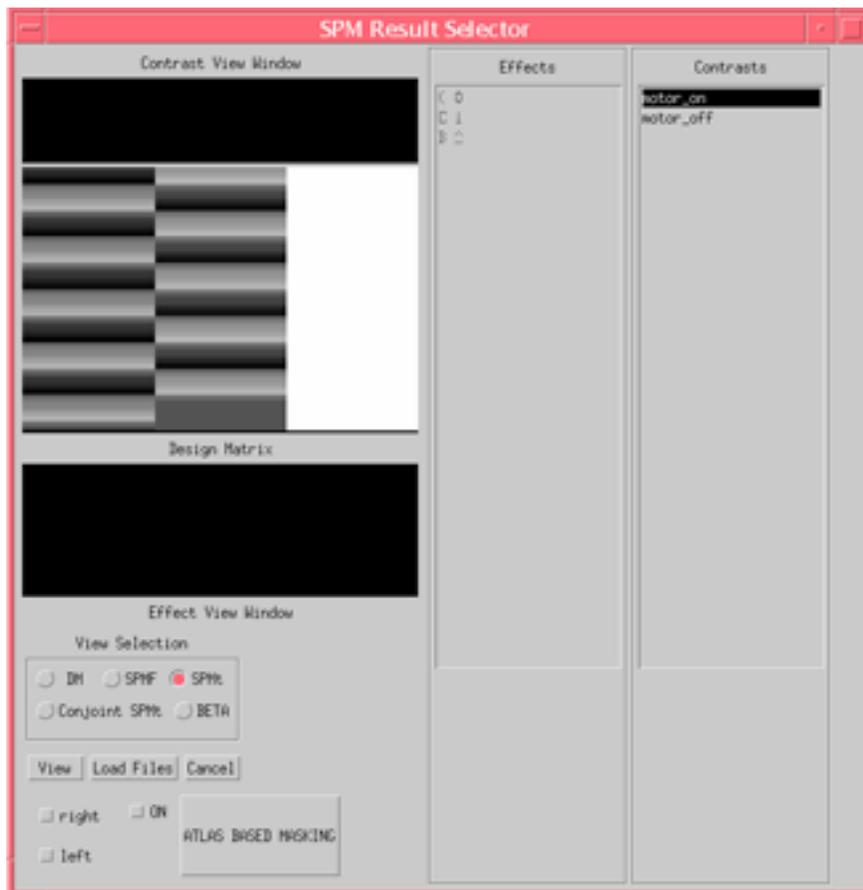




The program will then bring up another matlab spm99 window which will let you change the title of the contrast if you like. At this point, just hit return. The red thermometer will flash by as the contrast is computed on the fly.

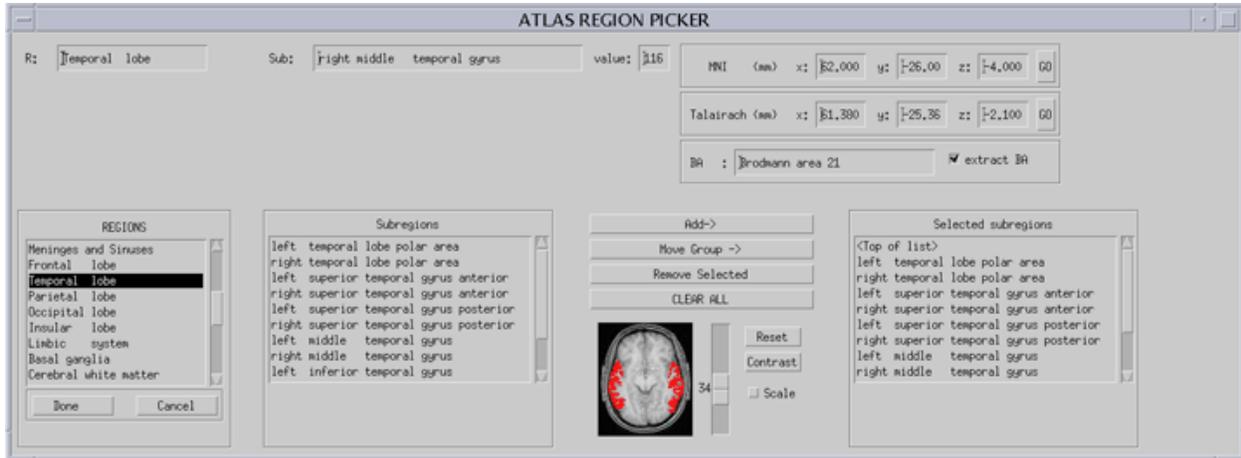


The program will automatically load all the appropriate files. You will then be brought to the SPM RESULT selector widget. All the contrasts you created will be visible here. There is no need to go back to choose another contrast. If you click on the <DM> button, you can view the waveforms of the various effects of interest and confounds that went into the analysis. When you are ready to view the data, make sure SPMt is selected, and then click the view button. Note that if you performed a single-event analysis, you may want to view the SPMF, not the SPMT.





If you want to apply an anatomic mask to your data, click on “ATLAS BASED MASKING”. This will bring up an atlas widget tool which will allow you to select anatomic regions to be included in the analysis. The atlas represents a segmented high resolution brain MRI which was previously registered to the SPM99 Talairach template.



When you are satisfied with the regions, hit the Done button. This will bring you back to the SPM RESULTS SELECTOR widget. There is no requirement to select a mask (masks are entirely optional, and should only be used for targeted hypotheses). Note: If you just want the right side or left side for bilaterally represented regions, you can click on right or left on the SPM RESULT SELECTOR widget (next to the Atlas Based Masking button at the bottom of the widget).

The functional data will be displayed on the patients/subjects normalized axial T1 (Talairach space). If this is a group SPM, the normalized atlas MR template will be used. If you want to change the overlay to the subjects localizer (pre-normalization, Radiologic space), or to another volume (sagittal, coronal, FSE, etc..), click on the Load Files button in the SPM RESULTS SELECTOR widget. This will bring up the SELECT FILES widget:



To use the atlas brain as the overlay, click on “Load Standard Brain”. To use some other volume for the overlay, click on the LOAD button near the Locs selector (second from bottom). Note that the endings for these files must be either .loc or .locs . To view the images displayed on a normalized epi volume, click off the Locs selector. When satisfied, click on the DONE button. It is not a good idea to play with any of the other selections.



When you have selected the contrast you want to view, and whatever masks you want to apply, and whatever overlay to use (the defaults will work fine too), then click on the VIEW button in the SPM RESULTS SELECTOR widget. This will bring up the SPMDISPLAY widget. This is a very complicated looking creature with a lot of buttons and sliders. There are only a couple that you need to know about. When the maps are displayed, anything you see in yellow is significant at a **corrected** voxel level of $P < 0.05$. If you look at the R max slider near the bottom right hand side, you will see the actual R - value which corresponds to this P value. Don't play with this slider, unless you are interested in changing the red-yellow cutoff. The widget defaults to a non-corrected cluster mode. If you are viewing SPMt maps, you can use either the Rstat field, or the P-value slider/field just below it to adjust the map cutoff. Note that this value is always a non-corrected value (for a 1-way t test). The SPMF slider pair is located above these 2 on the right hand side. There is a cluster size/ cluster p value slider pair near the middle of the panel. If nothing happens when you change one of the sliders (ie the matlab window doesn't flash), click on the <REDRAW> button at the bottom left.





Cluster level correction

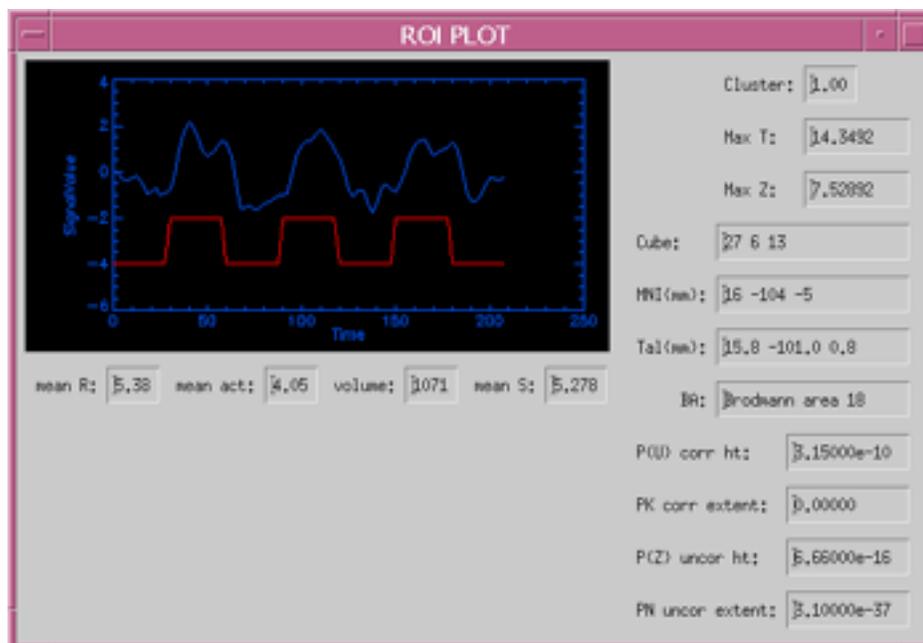
If you click on the corrected button (below the cluster sliders), you will no longer have direct control over the cluster size slider. You can adjust the cluster p value slider to set the corrected p-value you wish to view the maps at. Anything you see is significant at the cluster level. Some suggestions for the use of corrected mode. This initially may seem counterintuitive. The biggest problem in getting clusters to survive with the corrected mode on is the required cluster size. You can get this cluster size down by making the p value for the z map **more** significant. For example, in corrected mode, leave the cluster p value at 0.05, and change the tmap p-value slider to 0.0001 . When you do this, the required cluster size will drop, and you may see something survive. Of course, this assumes that you have data that can at least survive the initial tmap p-value cutoff.

Map level correction

At the bottom of the display window you will see a lot of text. The first entry specifies the map-level p-value. This computation is based on the number of surviving clusters vs the expected number of surviving clusters. The map-level p-value assumes your zmap threshold does not violate the rule of high cutoffs (ie, it's at least greater than 3). At low thresholds you will get a lot of tiny clusters which make the map-level p-value meaningless. Note that the rest of the text at the button tells you everything you need to know about the displayed map if you ever want to recreate it.

ROI Info

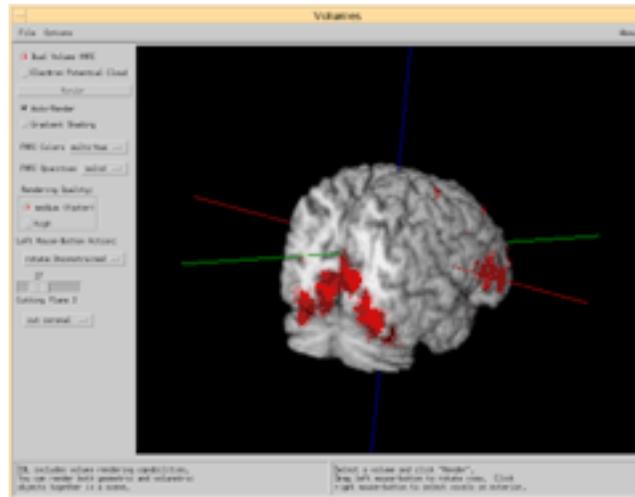
Just click on any cluster and all the statistics you could possibly want for that cluster will appear in the ROI PLOT widget. If you want to see a plot of the signal intensity, select the type of plot from the DISPLAY window. Choices are maxplot, fullplot, implot (means raw data prenormalized). Plots will not be displayed for group data (takes too long to load). Information will also be displayed in your idl terminal window regarding anatomic region you clicked on (based on the internal atlas) and the click coordinates.





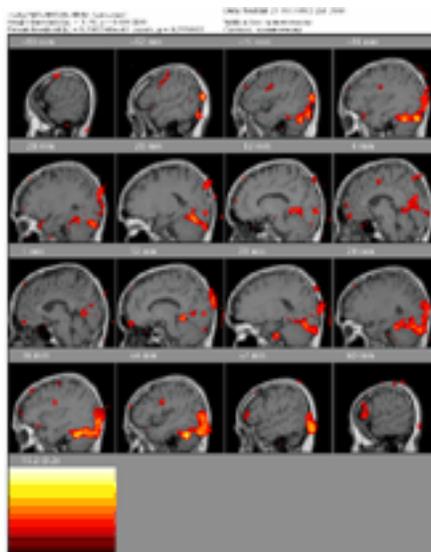
3D DISPLAY

This can be done 3 ways. If you click on the RENDER button, you will get the native SPM surface rendered display in the various orthogonal views. If you click on the SURFACE button, you will get an interactive dual-volume 3D surface display widget. You can rotate the display and cut into it. Makes for lovely pictures (use your system level snapshot utility for pictures). You can also use the MINISLICER for some really impressive 3D volume renderings (see below).



MULTIPLANAR DISPLAY

The MULTIPLANAR button will display the activation resampled in either axial, coronal, or sagittal planes (on the same anatomic template currently in use). You can specify the slice thickness, and slicing interval as well, to customize the output. Maximum and minimum thresholding values can also be set for standardizing the color scale.

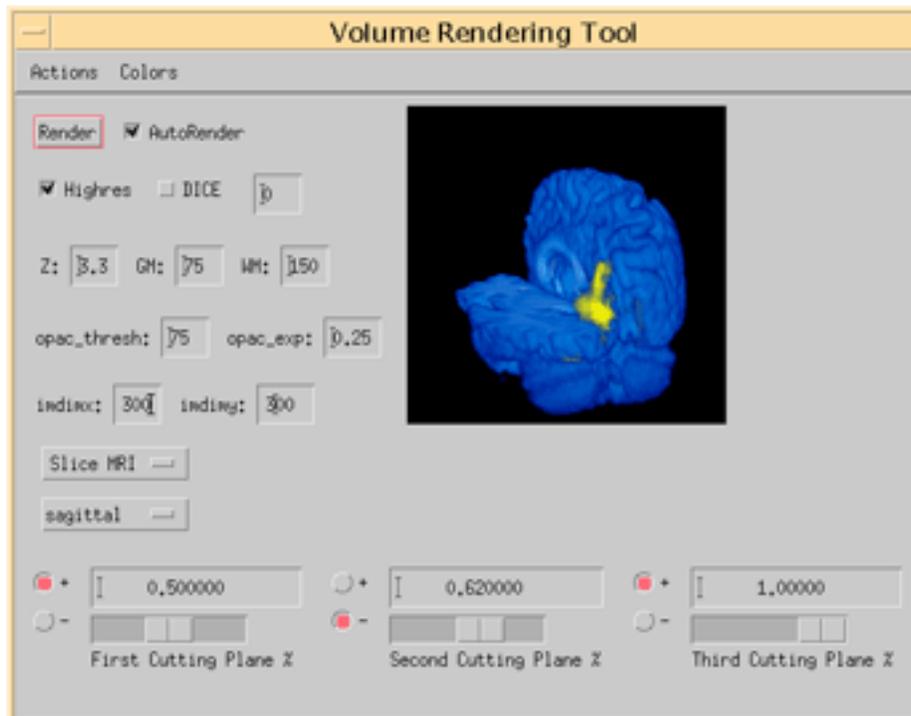




MINISLICER

The MINISLICER button will activate an interactive widget for 3D volume rendering and multiplanar slicing of your volume with rendered activation blobs. Orientation is controlled by mouse clicking and rotating an imaginary trackball on the display window. Slicing is controlled by the cutting plane sliders. Set the high resolution button when you have achieved the orientation you want. You can save your volume rendering to a TIFF image using the SAVE button under the Actions barmenu. The dimensions of your output TIFF file can be modified using the imdimx and imdimy fields in the widget. This tool can also be called from the command line:

```
minislicer, object =object,target = target, mrivol=mrivol,petvol=petvol,zscore = zscore
```





SAVING IMAGES

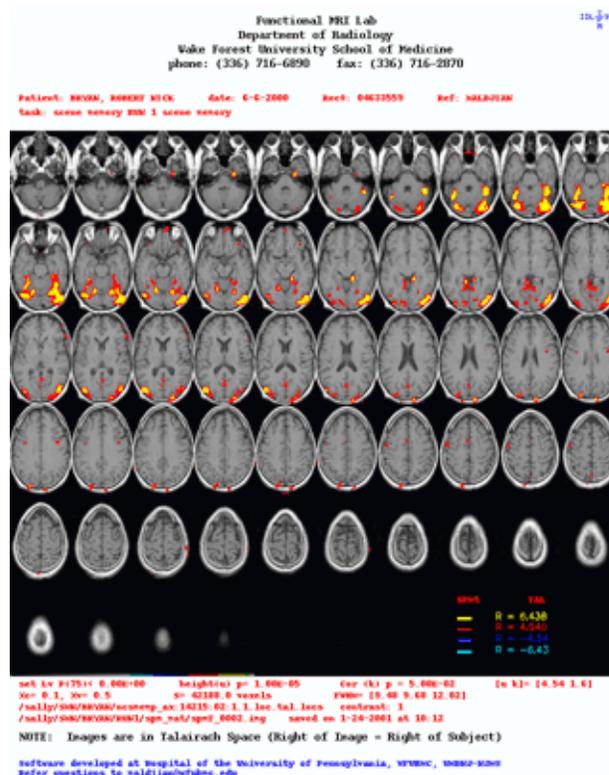
If you want to save an image of the display, this can be done using the SNAPSHOT button, or the PRINT button. Do not use the OPENWINDOWS snapshot tool that lets you take just a portion of the screen. If you do this, you will lose valuable text information at the bottom of the display which tells you all about how the map was created, and what its significance really is.

Snapshot

The snapshot button will make a TIFF file of the display.

Print

The print button will make also make a TIFF file of the display, then load it into the OPENWINDOWS IMAGE VIEWER. From here, if your workstation is setup properly, you can print the image directly. The lab banner at the top of the printout is determined by your !banner system variable file defined in the .idlspm/site/global_profile.pro file. The easiest way to change this is to edit the lab_banner.flst file directly (located in appl/IDL_SPM97/lab_banner.flst). The number at the top of the file defines the number of lines to be read. Alternatively you can define a new lab_banner.flst file in the .idlspm/site/global_profile.pro file by adding the statement: !banner = '/directory_location/new_lab_banner.flst' to the end of the file just before the end statement. You can determine the location of your .idlspm/site folder by executing the following command from the UNIX prompt: ~/.idlspm/echo_globals
The site_globals variable lists the .idlspm/site directory.





Masking the data

The SPM statistics are heavily dependent upon the number of voxels used in the analysis. The more voxels, the harder it is for clusters to survive correction. If you have an apriori hypothesis, you can reduce the number of voxels by masking your data. You can mask your data using the atlas tool as previously described in this section. An alternative method of masking is to define a manual mask. Click on the <SEGMENT TOOL> button on the data_control widget. You should load an axial .loc file for segmenting. Not a .tal.loc file. Note that the .loc file you load should be in radiologic coordinates (not talairach coordinates). I suggest you set up a special directory for holding various types of data masks, along with the original .loc file they are derived from. You will also need to put the sn3d.mat file corresponding to this .loc file in that directory.

Loading a mask

When you are in the SPM_RESULT SELECTOR (after the SPMVIEWER), click on the <LOAD FILES> button. A file widget will open which should have most of its entries already loaded. One of the selections near the bottom will be MASK. Load the mask you want to apply into this field.

Shortcut

The statistics are adjusted based on the number of voxels in the mask. You will see the number in the upper right hand box (S) of the DISPLAY_TOOL change. This is an editable field. If you know how many voxels you are correcting for, you can just change this field, without loading the mask. The P-values will be adjusted just like with the mask applied, however, clusters will not be masked.

Radiologic Viewing

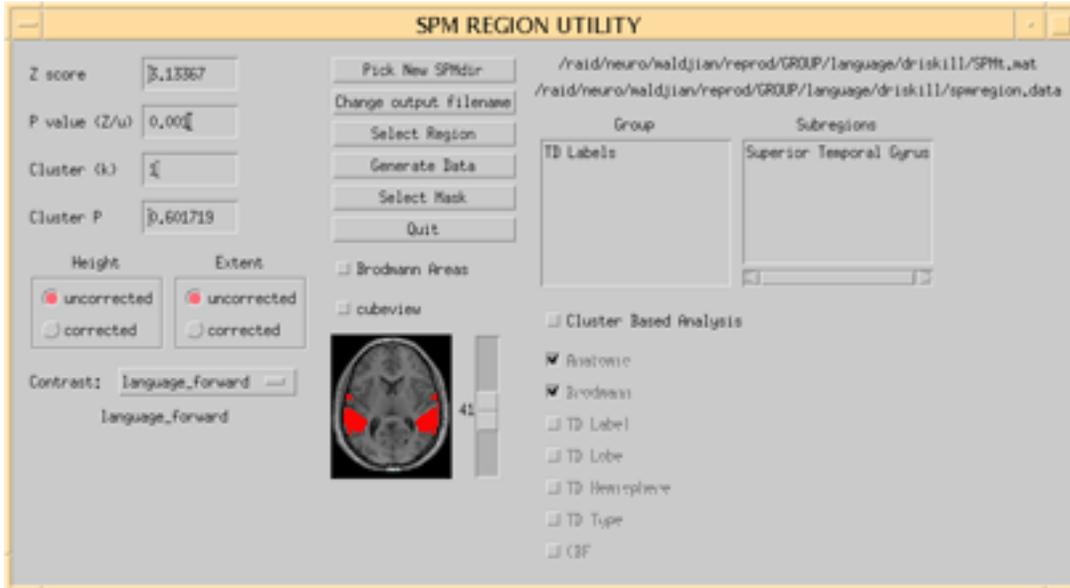
When you are in the SPM_RESULT SELECTOR (after the SPMVIEWER), click on the <LOAD FILES> button. In the locs field, click on the .loc file you would like to use. You can click on any of the .locs (including sagittal, or even from a different subject). The display will be in the coordinates of this .loc file. Talairach .locs have the .tal.loc ending.

VII. AUTOMATED ROI EXTRACTION AND ANALYSIS

IDL_SPM can output automated ROI data based on the anatomic atlas. From the FMRI UTILITY widget, click on SPM REGIONS. This will bring up the SPM REGION UTILITY widget. Select the contrast you are interested in, set your height and cluster thresholds and whether or not you want any height or cluster corrections applied. You will have to hit return after changing any of the P-values. Click on SELECT REGION. This will bring up the atlas region picker widget. Now you can select what regions you want data on. Click on DONE when you have selected the regions you want. Now click on GENERATE DATA. The program will automatically extract the anatomic region data from your SPMt maps, along with all the statistical information for each region you could possibly want. This will be output to a text file called spmregion.data in the spm_mat subdirectory for this data set. Click on the Brodmann Areas checkbox if you want to output Brodman areas with each region.



If you want a break-down of all the Brodmann areas or anatomic regions within a cluster (rather than the clusters within a region), click on the “cluster based analysis” button and select the atlases you would like information from. When you click on “Generate Data” a printout will be created listing all the clusters, and all the Brodmann areas and anatomic regions within each cluster, along with the cube, MNI, and Talairach coordinates for easy cross-referencing. The output text file will be called `spmregion.data.cluster`.



VIII. VOXEL BASED MORPHOMETRY (VBM) ANALYSIS

VBM analysis is built into IDL_SPM. You need to create a `vbm.flist` file listing the directories of the `.loc` files to be included in the VBM analysis. You can also use the `.flist` created during setup subjects (this will also only list the directories you are interested in). From the FMRI UTILITY widget, Click on RUN MULTIPROCESS. Select the `.flist` file when prompted. When the large ANALYSIS widget window comes up, click on VBM (at the right side of the widget). You may also want to turn off plastic for the VBM analysis. Hit DONE. The program will create `spm99` batch scripts for preparing your data for VBM. It will generate a new VBM directory where the normalized, segmented, inhomogeneity corrected, smoothed, and logit transformed images will reside. The smoothed grey matter segmented images can then be taken forward into a 1 sample T-test (or into a 2 sample t-test with a large group of normals as the second group) for the VBM statistical analysis. This can also be done through IDL_SPM by creating a `.flist` with the preprocessed images and entering it into a GROUP ANALYSIS.



IX. INDEPENDENT COMPONENT ANALYSIS (ICA)

ICA analysis capabilities can be added by obtaining the FSL distribution for Solaris and Linux from www.fmrib.ox.ac.uk/fsl. You can obtain the binaries for both platforms there. You may need to install the libX11.so.6.1 library for Solaris (also available at that web address). You will need to place the linux distribution in a separate linux_fsl folder. You will also need to add the fsl/bin path to your path environment variable (and fsl_linux/bin for linux). If installed properly, you should be able to start fsl from either Solaris or Linux platforms, simply by entering fsl at a shell prompt. To run an ICA analysis from within IDL_SPM, all you need to do is click on the ICA button in the large analysis widget. The program will check to see if melodic and avwmerge (part of the fsl distribution) are within your executable path before attempting to create the ICA scripts. The ICA output will appear in an ICA subdirectory within each RUN subdirectory. There should be a reports subdirectory generated which will display the various components of your ICA analysis in html format. At this juncture, no other integration is built into the IDL_SPM platform. Although it is a powerful model-free method of extracting activations, the major drawback is determining which of the component maps are useful.

X. FUNCTIONAL CONNECTIVITY ANALYSIS

Functional connectivity is a method of determining areas of the brain which have fluctuations in signal which are in synchrony with each other. It is usually performed on a resting-state data set. This has been incorporated into IDL_SPM by extracting the signal time course from areas of the brain you are interested in, and entering these into an SPM analysis as regressors. Each region should be performed as a separate analysis. This can be easily accomplished using the fcon routine. The syntax is:

```
fcon,/multiple,/analyze,/atlas,/separate
```

The program will prompt you for the analyze file to extract the signal from. Choose the snr file for this. If you don't set the /analyze option, the program will prompt you for a raw data .imfile to use to extract the time course data (the snr should give you much better results since that's what is ultimately going into the analysis). It will then open up the pickatlas viewer, and allow you to choose the regions you want to use from the atlas. The program will create subdirectories for each region, write a confound file (which contains the time course data for that region) and launch an spm analysis on each subdirectory. It will also automatically create a reference function with a single event. This is necessary to allow the spm analysis to proceed. Note that the program will rename any existing .se_ref files in your RUN folder so that they don't interfere with the analysis. When the analysis widget pops open, you should set the event section to hrf only, and set the mask data option.

Each subfolder will have a functional connectivity analysis with a weight of 1 assigned to the regressor. Although you can attempt to extract all the regressors in a single analysis (by not setting the /separate flag), this is not a good idea. You may be modelling regressors which will decrease the "activation" for the regressors you are really interested in. The results can be viewed just as any other SPM analysis using the SPM viewer.

You can also implement a functional connectivity analysis at the cluster level, or peak voxel level directly from the spm viewer. When viewing your data, set the map selectors to the



type of plot you would like (maxplot (peak), entire roi, adjusted data, raw data, etc), then click on the cluster you are interested in. The ROI window should display the plot. If you now click on the “FCON” button, the plot will enter into a functional connectivity analysis. Note that your original ref file will be renamed, and a single_event ref function will be created in that RUN folder. Depending on your plot selectors, the signal can represent either the peak voxel, or the entire cluster.

XI. MNI SPACE, TALAIRACH SPACE, and BRODMANN AREAS

The SPM templates are really in Montreal Neurological Institute (MNI) space. This is an approximation of Talairach space, but not really Talairach. When you normalize anything in native SPM, and you get out coordinates, these are in MNI space. This release incorporates a non-linear transformation defining the conversion between the two coordinate systems. This was adapted from the mni2tal.m and tal2mni.m programs written by Matthew Brett. A complete description of the problems of MNI space, and some solutions can be found at: <http://www.mrc-cbu.cam.ac.uk/Imaging/mnispace.html>

Brodmann Labels are extracted using the Talairach Daemon binary executables (<http://ric.uthscsa.edu/projects/tdreadme.html>). This program establishes a connection to the Talairach Daemon server, sends it talairach coordinates, and returns the Brodmann labels. Currently, the executables are compiled for the Solaris platform, so you will not be able to use the Brodmann labels if you are running IDL_SPM on a Linux Box.

XII. ATLAS DATA

IDL_SPM includes an anatomic atlas based on the Kikinis MR atlas. As of release 5.6, a full Brodmann atlas is included in the distribution. The Brodmann atlas was constructed using the Talairach Daemon. In addition, an anatomic label atlas, hemisphere atlas, lobar atlas, and tissue type atlas has been incorporated from the Talairach Daemon. A vascular territory atlas is also included. All the atlases can be accessed for masking data, cluster-analysis, and direct interrogation via the pickatlas tool. The atlases are all registered to the MNI SPM template.

XIII. WHAT IF I DON'T HAVE GE DATA

Viewing

IDL_SPM can be used to view any data sets that were processed in spm99. Just click on the SPM.mat file you want to load.

Running the analysis with DICOM data

If you have DICOM images for the localizers AND the functional runs you are in good shape. The DICOM data for the functional images MUST be 1 image per file. Some scanners generate Mosaic DICOM files for functional runs, in which all the images are stuffed into a single file. If your scanner does this, you're out of luck for now. You will have to follow the instructions in the “Running the analysis on non GE data without DICOM data” below.



You will need to have multiple subfolders of DICOM images, 1 folder for each series. So if we have a sagittal anatomic, an axial anatomic, and 3 functional runs, we should have 5 folders of DICOM images something like this:

DICOM_DATA_FOLDER

1.3.46.670589.11.8009.6.108181500282781183	(subject folder)
1.3.46.670589.11.8009.6.1081815002827811830101	(sagittal anatomic)
1.3.46.670589.11.8009.6.1081815002827811830201	(axial anatomic)
1.3.46.670589.11.8009.6.1081815002827811830301	(functional run 1)
1.3.46.670589.11.8009.6.1081815002827811830401	(functional run 2)
1.3.46.670589.11.8009.6.1081815002827811830501	(functional run 3)

Note that for the functional runs there MUST BE 1 file per image. No Mosaics. In each functional run directory there can be thousands of images. For example, for a study with a TR = 2, 30 slices, and 4 minutes of imaging (120 temporal volumes) there should be 3600 image files in each functional directory (30 slices * 120 volumes).

Now execute the following command in idl_spm:

```
cycle_dicom,/keep
```

When prompted for a directory selected the PARENT directory (in this example you would select: 1.3.46.670589.11.8009.6.108181500282781183)

The program will automatically generate a new directory and convert the anatomic series to .loc files, and convert the functional runs to .im files. The resulting folder will be labeled with patient_name_date_of_study. In this example, if the subject name was JM, scanned on 12-22-01, a new folder would be created in the main data folder with the following files:

DICOM_DATA_FOLDER

```
JM_12_22_01
  _sagittal.loc
  _axial.loc
  JM_10:40:01_12_22_01.im
  JM_10:46:01_12_22_01.im
  JM_10:53:01_12_22_01.im
```

You can run the IDL_SPM analysis directly on this folder from “SETUP SUBJECTS” forward. You will have access to all the features of IDL_SPM as if the studies were done on a GE scanner.

The cycle_dicom routines takes the following keywords:

```
cycle_dicom, dir = dir, keep = keep, discard = discard
```

If you want to keep the DICOM images, set keep = 1, otherwise, they will be removed after conversion to .loc and .im files.

If you want to **discard** several volumes at the beginning of each functional run, set the discard keyword to the number of volumes to be discarded. For example, if we want to throw away the first 3 volumes:

```
cycle_dicom, discard = 3
```



Running the analysis on non-GE data without DICOM data

You can still use IDL_SPM, however, you will have to reconstruct the raw data and generate analyze images on your own.

The structure of the analyze image filenames must conform to the following:

patient_name_RUN1_00001.img ...

patient_name_RUN2_00002.img ...

Create a patient folder for this subject. Within the patient folder create RUN subfolders for as many runs as you have:

RUN1

RUN2

RUN3 ...

Within each run folder create the following 2 folders:

spm_mat

normalized

Put your analyze images for each run within the respective patient_name/RUN/normalized subfolder. So the analyze data from your first run would go in the /patient_name/RUN1/normalized directory, the analyze images from the second run would go in the /patient_name/RUN2/normalized directory, etc..

Now create a .se_ref reference function for your runs as described in the reference function section. Copy your .se_ref file for each run, rename it run1.se_ref.im, run2.se_ref.im, etc, and put these files in the patient directory. So now your directory structure should look like:

```
patient_name
  run1.se_ref.im
  run2.se_ref.im
  RUN1
    spm_mat
    normalized
      patient_name_RUN1_00001.img ...
      patient_name_RUN1_00001.hdr ...
  RUN2
    spm_mat
    normalized
      patient_name_RUN2_00001.img ...
      patient_name_RUN2_00001.hdr ...
```

Note that the name of the parent patient folder (patient_name) must be identical to the beginning part of the analyze file name, as indicated.

Now you can setup your analysis as previously described (through setup subjects). The program will recognize the .se_ref.im files as if they were separate raw data files. When the large analysis window pops up, you have to turn off some features. Deselect the normalize button (unless you have GE .loc files). Click on REALIGN OPTIONS. Turn off spm_shadow



transform (since this would have been based on a GE localizer). You will want to turn on `spm_internormalize`, otherwise your data will not be normalized at all. Now you can hit DONE to get out of `REALIGN_OPTIONS`, and hit DONE to start the analysis. The batch scripts will be appropriately created for your data sets, and you will be able to view your data without a problem. The built-in atlas template will be used for overlays during viewing.

XIV. POWER USER

You know who you are. You eat kshell scripts for breakfast. If you think `awk` is an expression used by Peanuts characters, or that a unix prompt can be found at a Clinton white house reunion, this is not for you. There are many routines which are part of this release which can probably already do anything you were thinking of writing (including bi-directional dicom image converters, matlab engine execution scripts, image registration routines, diffusion processing routines, perfusion processing routines, Linux /Solaris cross platform compatibility...). Here are just a few of the more useful ones:

Some Header Routines

```
fh=read_header(filename = filename, get_cube = cube)
    Reads any IDL_SPM image file (including SPMs, .locs, .bse, analyze)
    Returns header structure in fh
    Returns image volume in cube
helpstruc,structure
    Lists the contents of a structure
gh = read_genheader(filename = filename)
    Reads genesis header (5X or LX)
rh = read_rawheader(filename = filename)
    Reads rawdata header (5X or LX)
restorest56,imfile,stack,coords
    Reads .imfile into stack variable
image = open_square_image(filename = filename)
    Reads any "square" image file (regardless of header, ie i.00 GE images)
```

Some Registration Routines

```
auto_norm,filename=filename,noplastic=noplastic,mi=mi,coregister=coregister
    Performs a batch mode spm normalization of your .loc. If you set the coregister
    keyword, it will use spm_coregister to perform the normalization (and generate a
    .sn3dmat). If you set the MI keyword, it will perform a mutual information
    normalization.
```

```
spm_coregister,target=target,object=object,o_mode=o_mode,t_mode=t_mode,mi=mi
    Performs spm coregister of object to target (analyze images). o_mode and t_mode
    describe modality of object and target (ie., 'EPI', 'T2')
    Set mi keyword for mutual information coregistration
```



locs2analyze,filename=filename

Will convert a .loc to analyze format

mt = find_object2target(object = object,target = target,/compute,outv = outv)

Computes the affine transform between object and target image files, and will optionally apply it to the object volume, and return the resampled object in outv. This assumes there is a way to go between the files (ie., previously normalized, or acquired during the same imaging session).

resampled=tran3d(volumein=cube,mt=mt,outmatrix=outmatrix,step=step,start=start)

returns resampled volume

volumein=starting image volume

mt = 4x4 affine transformation matrix for resampling

Outmatrix=[x,y,z] dimensions of resampled volume

Start=vector describing start position for resampling (default=[0,0,0])

Step=vector describing step size of resampling (default = [1,1,1])

coregister_tool

Interactive widget for manually coregistering 2 image volumes. Very useful if you have fiducial landmarks on both volumes, otherwise, you can use anatomic landmarks. The program listing 10 fiducials, however, a minimum of 6 fiducials need to be defined to compute the 4x4 affine transformation. You can set more by using the fiducials keyword (ie., coregister_tool, fiducials = 20 for 20 possible fiducial markings).

regions=pick_atlas()

Just try it. Talairach atlas widget with anatomic labels.

Some image conversion and viewing routines

locs2analyze,filename=filename

Will convert a .loc to analyze format

write_header,header=header,cube=cube,filename = filename

Will write an image filename with header and cube

imviewer,volume1 = volume1,volume2=volume2,title = title,[file1=file1,file2=file2]

Cheesy image viewer if only 1 volume provided

Will also perform color overlays if volume2 is provided, Permits thresholding and selection of color tables with dual volumes. If file1 and file2 are provided (.locs) it will compute the affine transform between file1 and file2, and overlay file2 images on file1 images.

im2analyze

will generate analyze format volumes from a .im file.



`minislicer,mrivol=mrivol,petvol=petvol,object=object,target=target,zscore = zscore`
3D dual volume display widget with multiplanar slicing capabilities. Makes for beautiful surface renderings.

`dicom2im,dir = dir,keep = keep,discard = discard`

Will convert a directory of dicom images to .loc or .im (automatically detects type of study based on dicom header). Set discard keyword to number of initial volumes to discard for functional runs. Set keep keyword to prevent removal of DICOM images after conversion.



APPENDIX I (INSTALLATION)

System Requirements

1. IDL
You need to be running IDL 5.4 or better .
2. Matlab
You need to be running some version of Matlab 5.3 or Matlab 6.0
Currently, Matlab 6.1 is not supported. It is still buggy with SPM99.
3. Operating System
Solaris Sparc and/or Redhat Linux v7.0
4. Multiple floating network IDL and Matlab licenses. Since this release makes use of spm batch mode, and farms out processes to available cpus, you need to have as many licenses as you want to have cpus in the processing architecture. Note that a floating IDL or Matlab license will work on both the SUN Solaris platform, and the Linux platform.
5. Hardware
When I want to add more CPU power, I just add a Linux box sans monitor for around 1K. These are inexpensive, and blow Sparcs out of the water. I have found that 128 MB of RAM is adequate for single subject and random effects analyses without any speed penalties from disk swapping (although I usually get 256M RAM). If you want to do Fixed Effects analyses, then you will need a computer with a lot of RAM (512 or more), otherwise performance will suffer from disk swapping. Fixed effects analyses load all the data sets into RAM at once, so this will bring a low-RAM computer to a crawl.

Software Download Site

Open a web browser to the following address:

Some Preliminaries for a smooth install before you start

1. Cross mount your home directory to all the cpus you are planning to use. This way you will have the same user environment regardless of what cpu is being used, and you only have to do the configuration once.
2. Setup rsh between all your cpus so that you can do:
rsh hostname ls
from any cpu to any other cpu without a problem
3. make sure idl and matlab aliases are working on all cpus
4. If you are going to have many short-term collaborators, consider creating a generic fmri user account for their collective use.

Configuring the main program directory

Decide on the location for the main program directory (appl). There must be at least 500 MB of available space for the programs, and this directory should be write protected from generic users. To simplify the upgrade process, this directory should be on 1 computer, and it should be cross-mounted on all the other computers which will be using IDL_SPM. Download the appl_idlspm_5.70.tar.Z file from the program web site. In this example, we will create the main program directory off the /space partition on the server called moog:



```
cd /export/home/download
become superuser
uncompress appl_idlspm_5.70.tar.Z
tar xvf appl_idlspm_5.70.tar
cd appl_idlspm_5.70
./upgrade_idlspm
```

The upgrader/installer will prompt you for the location of the installation directory
In this example we will enter:
/space

The installer will create the /space/appl program directory with all the necessary subdirectories.

Upgrading to the latest release

Now you will need to upgrade to the latest version. Upgrades are easy to perform using the automated installer. First, download the latest upgrade from the program web site. Then:

```
cd /export/home/downloads
uncompress appl_idlspm_5.71.tar.z
tar xvf appl_idlspm_5.71*
cd idlspm_5.71
become superuser
./upgrade_idlspm
```

supply the path of the program directory when prompted (/space/appl)

Note that within the upgrade, I also automatically patch your spm99 matlab program directory. Periodically, the spm group releases bug fixes for their code. I try to keep up to date on these modifications so that the idl_spm distribution remains current.

Creating the ref_functions directory, cpufile, and job_control files

You will need to create a directory which will contain all your reference functions. This directory will also need to be cross-mounted to your other computers:

```
cd /DATA
mkdir ref_functions
cd ref_functions
cp /appl/IDL_SPM97/spm97/sample_refs/* .
```

In the ref_functions directory, you will need to create a cpufile. This will contain the names of all the cpus to be included in batch mode processing:

Open a file called cpufile (in the ref_functions directory) in any text editor and enter the names of the host cpus one line at a time:

```
natasha
aw_fmri
moog
```

Note that the fastest cpus should be at the top of the file

Now create an empty file called job_control in the ref_functions directory.



Configuring Users, IDL, and Matlab

User configuration is done at 2 levels. One is at the level of the site. Site specific variables (like matlab and idl paths) need only be set in one place for all users. This is done in the `.idlspmsite` directory. The second level is at the level of the user. Each user will have a `.idlspm` directory within their home directory. Files within this directory permit user-specific configuration (ie., email address, data directories etc).

There are 2 ways of creating the `idlspmsite` folder. One is manually by editing templates of the files provided in this distribution. The second is using the `idlspm_wizard` installation tool. The wizard will work if :

- 1) you have configured rsh permissions for your sparc and linux box
- 2) you already have created aliases in your setup for idl, matlab5, and matlab4 (if you use matlab4)

IDL_SPM wizard

This is a program located at `/appl/IDL_SPM97/idlspm_wizard.pro` . It can be run independently of the rest of the programs.

- 1) start idl
- 2) cd to the IDL_SM97 distribution directory
- 3) `.run idlspm_wizard`
- 4) `idlspm_wizard`

Then just follow the prompts. The wizard will configure your `idlspmsite` directory. You will still need to configure each users' `.idlspm` directories, but that's trivial. The wizard reads your current configuration, so your matlab5 and idl aliases had better be correct before you start.

Wizard user configuration (create_user)

The `idlspm_wizard` also includes a tool to automate configuration of individual users. To use it:

```
Start idl
Cd to the /appl/IDL_SPM97 folder
.run idlspm_wizard
create_user
```

This will prompt you for information on the user (username, homedirectory, default data directory, and email address). Make sure you have already configured the `.idlspmsite` directory (see above section), filled your `cpufile` with the cpus you will be using, and you need to know the path of your `.idlspmsite` folder. You can run the wizard as the user. If you are not running as the user, then you will need write permissions to the user home directory. It will automatically do the following:

- 1) create the user's `..idlspm` directory in the user home directory
- 2) create the `.global.ksh` and `.user_profile` files
- 3) populate the folder with the remaining necessary files
- 4) create the `bootfmri` and `bootspm` scripts in the home directory
- 5) create a `.rhosts` file if one doesn't exist based on the entries in your `cpufile`
- 6) add the source `~/..idlspm/.cshrc.idlspm` line to the user's `.cshrc` file



Manually configuring the .idlspmsite directory

System administration is performed through the idlspmsite directory. This directory should be on a partition which is cross-mounted to all cpus.

1. Copy the /appl/IDL_SPM97/SPM99F/SHELL_SCRIPTS/idlspmsite folder to where it will reside.
2. Edit the cshrc.idlspm.global and cshrc.idlspm.global.linux files so that the idlx and matlabx aliases are set correctly. These must point directly to the idl and matlab executables, and must not go through xterm. Make sure the remaining idl and matlab paths are set properly in these files. Note that the cshrc.idlspm.global.linux file differs somewhat from the cshrc.idlspm.global file. In my configuration, I also use the cshrc.idlspm.global.linux file to set standard paths. You may not want this. Just comment out the portion on standard paths if you want to set them elsewhere. However, the idl and matlab paths and aliases must be set properly in the cshrc.idlspm.global.linux file.
3. Edit the global.site.ksh file. This is one of the most critical files. All your global variables – including idl and batch execution are now read from this file.

Manually configuring user .idlspm directories

1. Copy the /appl/IDL_SPM97/SPM99F/SHELL_SCRIPTS/idlspm folder to your home directory. Then rename it .idlspm:

```
cp -R /appl/IDL_SPM97/SPM99F/SHELL_SCRIPTS/idlspm* ~/.idlspm
```

2. Edit the .global.ksh file so that the site_globals variable points to your idlspmsite directory and set your user specific email info.

Add the following line to your .cshrc file:

```
source ~/.idlspm/.cshrc.global
```

3. Edit the following files in ~/.idlspm using the embedded instructions within the files:

```
user_profile  
.global.ksh
```

They will contain site specific modifications and some user-specific modifications. Make sure the site_globals variable in the .global.ksh file points to your idlspmsite directory and set your user specific email info.

4. Once you have created your .idlspm folder for yourself, copy this folder to the home directory of all your users. They will only have to edit the .global.ksh file so that the email address is correct for them.
5. Create the following 2 files in your home directory:

```
bootfmri  
bootspm
```

In bootfmri put:

```
cd,current = home  
cd,'~/.idlspm'  
@bootfmri  
cd,home
```



```
In bootspm put:  
cd,current = home  
cd,'~/idlspm'  
@bootspm  
cd,home
```

6. Setup your .rhosts file

In order for batch mode to operate properly, each user must have rsh permissions across all cpus listed in the cpufile. Create a .rhosts file in your home directory and enter the names of the cpus within this file. Sometimes Linux boxes will be unhappy with a .rhosts file unless the permissions are set to 555:

```
chmod 555 .rhosts
```

You should now be able to:

```
rsh moog ls  
rsh kyle ls  
rsh natasha ls  
rsh aw_fmri ls
```

From any of your computers, and get a listing without any password prompts. If not, then you need to check your .rhosts file again.

Testing your installation

Type idl (idl should start)

Type matlab (matlab5 should start)

Type idlx (idl should start from within the same terminal window)

Type matlabx (matlab5 should start from within the same terminal window).

Start idl:

type @bootfmri (programs should compile without complaints)

type @bootspm (programs should compile without complaints)

If you get any complaints, check your bootfmri and bootspm files as described above.

Now enter:

```
check_install
```

This will test your matlabx, idlx, and batch mode operations:

Hit return at the prompts. Note that it will complain that your matlab alias is not set for matlab4, but that's OK, since we're not supporting matlab4 and SPM96 anymore. It will write an outputlog to your home directory. At the end, a new xterminal window will open. Within this window, it will test your batch mode operations for all the cpus listed in your cpufile. If you see error messages in this window, then you must correct your configuration. This is usually related to lack of rsh permissions. Look at the DIAGNOSTICS window output carefully. It should look like the example. Depending on your version of lmgrd, sometimes lmstat doesn't return the expected output (case sensitive), so you end up with no idl licenses available. If this is the case, rename the executable script check_license_now_new to check_license_now. It is located in the IDL_SPM97/SPM99F/SHELL_SCRIPTS directory. You can also test for licenses from this directory by entering: check_license_now idl 1 or check_license_now matlab 1



```
DIAGNOSTICS
-----
trying elfido 1
elfido vmflag = 1 idle time = 100 mem free = 118
elfido idl licenseflag = 1 available = 15
elfido matlab licenseflag = 1 available = 1
matlab processes = 0
idl processes = 0
cpu_finder processes = 0
-----
trying alexia 2
alexia vmflag = 1 idle time = 89 mem free = 19
alexia idl licenseflag = 1 available = 15
alexia matlab licenseflag = 1 available = 1
matlab processes = 0
idl processes = 1
cpu_finder processes = 0
-----
trying e3000 3
```

Configuring Additional Users

Once you have one user setup, the rest are trivial. You can either use the `create_user` program from the `idlspm_wizard` (see above), or you can do it manually by copying a working user's `.idlspm` folder and making some minor edits. Just copy the `.idlspm` folder into each user's home directory, copy the `bootfmri` and `bootspm` files into their home directories, and edit the `.cshrc` file so that it sources the `./idlspm/.cshrc.idlspm` file.

You can make some minor user-specific edits within the `user_profile` and `.global.ksh` files (ie, setting email notifications, data directories). Note that you can have separate `cpufiles` and `ref_function` directories for different users if you like. These can be controlled in the `global.ksh` file within each user's `.idlspm` directory.

Tuning up the batch processing

You will probably need to edit your `/etc/inetd.conf` file, especially on your linux boxes, to allow multiple rsh without causing the rsh server to shut down. Increase the number of successive rsh attempts allowed to 4000 in your `inetd.conf` file like this:
shell stream tcp nowait.4000 root /usr/sbin/tcpd in.rshd

Upgrading to Matlab6, IDL 5.X and Beyond

Matlab6 and 6.1 engine support is provided in the latest release (for both solaris and linux). Batch mode appears to work fine with matlab6, however, there is a postscript printing problem with matlab6.1 and spm. For now, matlab6.1 is not recommended with spm99. If you upgrade your version of matlab, the following files will need to be modified (these files are in the `.idlspmsite` folder – modifications will affect all users):

global.site.ksh

- modify the `matlab_dir` variable to reflect the new matlab path (note that this occurs in 2 places – 1 for Solaris, and 1 for linux)
- modify the `idl_dir` variable to reflect the new idl path (this also occurs in 2 places)



make sure the `lmstat` executable lives in the new `matlab_path/etc` folder. If not, modify the `lmstatmatlab` variable (near the bottom of the `.global.site.ksh` file) to reflect the correct location of the `lmstat` file.

cschrc.idlspm.global

modify the `matlabx` alias to reflect the new `matlab` binary path. Remember to add the `–nojvm –nosplash` flags, otherwise `batchmode` will choke
create a `matlab6.X` alias for your new version of `matlab`
check the `matlab5`, `matlab6`, and `idl5` aliases to make sure they are correct

cschrc.idlspm.global.linux

modify the `matlabx` alias to reflect the new `matlab` binary path. Remember to add the `–nojvm –nosplash` flags, otherwise `batchmode` will choke
create a `matlab6.X` alias for your new version of `matlab`
check the `matlab5`, `matlab6`, and `idl5` aliases to make sure they are correct

Changing the Location of the appl folder

If you need to change the location of the program distribution, you will need to modify several files in the `.idlspmsite` folder to reflect the new location. If you do not know the location of the `.idlspmsite` folder, execute the following commands from your home directory:

```
cd .idlspm  
./echo_globals
```

You should see multiple lines of output, one of which will say something like:

```
site_globals=/fmri2/fmri/.idlspmsite
```

That is the location of your `.idlspmsite` folder. The environment variable to be modified is `appl_path`, and it appears in 3 of the login scripts in the `.idlspmsite` folder:

global.site.ksh

modify the `appl_path` variable to reflect the new `appl` path

cschrc.idlspm.global

modify the `appl_path` variable to reflect the new `appl` path

cschrc.idlspm.global.linux

modify the `appl_path` variable to reflect the new `appl` path

SPM96

This release is no longer explicitly supporting SPM96. Although SPM96 support is built into the code, the problem is at the `matlab` end of things. With the latest release of `matlab 6.0`, the `matlab` license manager will not also support `matlab 4.2c` (which is required for SPM96). If however, you have a burning desire to use SPM96/97 style analyses, you will need to have a way



of running both matlab4 and matlab5. I can't help you with that. However, if you have this capability (either by maintaining an old version of matlab4.2, or running the matlab license manager for 5.x and 4.2c and never upgrading to matlab 6), you can unlock the built-in SPM96/97 functionality very easily by setting your matlab alias in your .cshrc file to run matlab4.2c instead of matlab5 (your matlabx alias should still run matlab5/6). You will also have to change the soft link in your /bin or /usr/bin directory which matlab creates during installation so that it points to the matlab4.2c executable. When you run the check_install routine, you should now see both the matlab 4.2 window flash, followed by the matlab5/6 window.



APPENDIX II (Data Acquisition and Archival)

These scripts are located in the IDL_SPM97/LX/LX_PSDS subdirectory. They include scripts for automated ftp, LX database image extraction based on SQL, and raw data archival utilities. There is a README file in the subdirectory which describes minor site-specific modifications which will be necessary for implementation (setting ftp remote host, data directories, and hardware archival flags). See Appendix IV (Archiving to CD) for information on using the fmriarch and extract scripts for CD-archival.

Archiving fMRI Data

The automated archiving script will do the following in the background:

1. rename Pfiles with subject label and time date stamp
2. ftp Pfiles to DATA directory (/fmri/fmri/DATA) on predefined host (relito)
3. initiate remote recon on host
4. archive Pfiles to tape (if you have a tape device)
5. archive Pfiles to CDR (if you have installed a CD-writer)

To start the script:

Open a terminal window.

Type: `fmriarch` or `fmriarch notape nocd` (for **n**owrite to tape, and **n**o write to CD)

You should see some messages verifying the ftp connection.

If you have a tape device, the program will prompt you to pop a tape in the drive at this point (it will autodetect your hardware). It will not let you continue until you put a tape in the drive. If you want to override writing to tape, use the `notape` flag when you start the script (`fmriarch notape`)

You will be prompted to enter a patient label:

Enter the last name of the subject/patient, no spaces, no funny characters

Now you can continue scanning. As pfiles are acquired, they will automatically be handled in the background. When done scanning, check the output log to make sure all your files have been archived. When the last file has been archived, hit `ctrl C` to exit the `fmriarch` script. The `fmriarch` script will work with any Pfiles. The automated recon will work with both spiral and `fmriepi`. See Appendix IV for using `fmriarch` with a CD-burner.

Running fmriepi:

`fmriepi` is a psd based on `epi`. It is not spiral. The image quality is comparable to spiral.

The advantages over spiral:

More slice coverage (with asymmetric FOV)

Distortion Correction

Better sensitivity to BOLD effect ?

Magnet coordinates are correct (the spiral sequence needs some minor modification for now)

Disadvantages relative to spiral: more warping (vs blurring in spiral)

Perform the sagittal and volume axial scans as you normally would. It is important to include the entire brain on the volume axial T1 with some air above the top of the head (otherwise the SPM normalization will warp the head).

PSD location: `/usr/g/bin/psd/jam/fmriepi`

Scan parameters:



2D
axial
EPI
gradient echo
1 shot
TE 50
TR whatever (3000)
flip angle 90
5 mm
no skip
FOV 24 cm
matrix 64x40
FOV ratio 0.625
frequency direction A/P

TR of 3 will allow 36 slices
TR of 2 will allow 24 slices

Set up slices (using S and I values)

Scan

This will output images to the console, where you can check your position. Make sure you have started the fmriarch script in a terminal window before continuing. If satisfied with position:

1) Perform distortion correction

modify CVS
rawfmri = 1
csiflag = 1
SCAN

This will acquire the distortion correction file (about 1:40)

2) Perform fMRI Runs

modify CVS
csiflag=0
realreps=number of images you want
Scan

So, for a 5 minute paradigm, TR =3, that would be 100 images (20 images per minute), so realreps = 100

Note that the scan clock will say 5:18 (or 5:20). The actual computation is $(\text{fix}(20 \text{ sec/TR}) * \text{TR})$ for ddas. This is to allow for achieving steady state. No images are acquired during this time. The actual image acquisition in this example will start when the clock hits 5:00. You can either build the delay into your paradigm, or start your paradigm when the clock hits 5:00. It's important to know the TR, because the initial delay will be some multiple of the $\text{TR} \leq 20$ seconds.

3) You can just keep hitting the scan button for more runs. There is no need to repeat the distortion correction unless you change slice position, or matrix size.

4) Exit the fmriarch script after the last file has been written.



Archiving the anatomic images

The anatomic images can be archived using the extract script. The extract script uses SQL to access information in the LX patient database. It then makes successive calls to ximg to extract images (with LX headers). As the images are extracted they are moved to a unique folder created in the ~/LOCS directory on the scanner. Once all the images are extracted, this folder is tarred and ftped to the remote server. If the ftp connection is down, the tar file will still be there. You can open multiple simultaneous instances of extract, since each image is moved as it is obtained. Note that currently, the extract script does not work on the cardiac magnet. If you are archiving to tape as well, you can just leave the tape in the drive that you used for the fMRI.

To use the extract script, you need to know the exam number. If you don't, just enter:
extract

This will display a list of subjects and exam numbers currently on the scanner database. To extract all the anatomic images for a particular subject:

extract examnum

To extract specific series:

extract examnum seriesnum1 seriesnum3 seriesnum7 ...

The extract script will prompt you for a patient label. If this is part of an fMRI examination, enter the same patient label (including capitalization) that you used for the fmriarch script. Your data will be automatically ftped and sorted in the same subject directory on the remote host. See Appendix IV for using the extract script with a CD-burner.



APPENDIX III (Directory Structure and File Naming Conventions)

DIRECTORY STRUCTURE

patient_name

patient_name_time_datestamp
 patient_name_time_datestamp.im
 patient_name_time_datestamp.im.bse
 efgre3d_ax.loc
 efgre3d_ax.loc.img
 efgre3d_ax.loc.hdr
 efgre3d_ax.loc.img
 nefgre3d_ax.loc.hdr
 nefgre3d_ax.loc.tal.locs
 patient_name.flist
 cor00.dc

batch

RUN1
RUN2

locs

examnum
001

I.001
I.002 ...

002

I.001
I.002 ...

RUN1

normalized

snrpatient_name_RUN1_00001.img ...
 snrpatient_name_RUN1_00001.hdr ...

spm_mat

SPM.mat
 spmT_0001.img/spmF_0001.img/beta_0001.img/con_0001.img ...

RUN2

normalized

snrpatient_name_RUN1_00001.img ...
 snrpatient_name_RUN1_00001.hdr ...

spm_mat

SPM.mat
 spmT_0001.img/spmF_0001.img/beta_0001.img/con_0001.img ...



FOLDER CONTENTS

batch folder

contains separate RUN folders for each RUN:

batch.flist	= lists all scripts for this analysis
batch_que	= lists scripts remaining
batch_log	= batch output log
output_unix	= batch unix output log
output_matlab	= batch matlab output log
output_idl	= batch idl outputlog
_u	= unix shell script
_idl	= idl script
_x.m	= matlab script
.m	= matlab batch script to be sent to bch.mat

locs folder

contains GE structural image files (extracted via ximg or extract)

run folders

Contains ref_function.se_ref file and spm_mat and normalized folders for this RUN:

spm_mat folder

contains SPM output data files

normalized folder

contains analyze images entering into analysis

FILE NAMING CONVENTION

no .suffix	= raw data Pfile (patient_name_timestamp)
.im files	= reconstructed image volumes with raw data header
.loc	= anatomic image volume with GE 5X or LX header
.tal.locs	= normalized anatomic volume, internal format
.bse	= internal image format (similar to .loc)
.se_ref	= text file reference function definition
.img/.hdr	= analyze format image volume
s prefix	= smoothed
n prefix	= normalized
r prefix	= realigned
l prefix	= logit transformed (part of VBM)
I.001	= GE image file (single image)
.dc	= distortion correction parameters
.flist	= text file with # of entries followed by files/directories for analysis



APPENDIX IV (Archiving to CD)

Program Files

The archiving programs are located in /appl/IDL_SPM97/LX/LX_PSDS.
The following files belong in the home directory of the magnet (/export/home/sdc):

autoftp
extract
fmriarch
ftp_check
remote_launch
imenu

The following directory belongs in the home directory of the magnet
(/export/home/sdc/CDRECORD):
CDRECORD

The following files belong in the home directory of your ftp host:
autolocsort_x
autorecon_x

If you want to implement autoftp and autorecon, you need to edit the ftp_check file to enter your site specific information.

In order for autorecon to work, your idlhost (in the ftp_check file) must be configured for batch mode operations in idlspm, and you need to have a .rhosts file in your home directory on the idlhost which allows the magnets to rsh in.

You will also need to edit the autolocsort_x and autorecon_x files near the top to indicate your display server.

To turn off autoftp and autorecon:
edit the fmriarch and extract routines

At the top of the file, in the user edit section, set:
ftpflag=0
reconflag=0

IMPORTANT: When modifying any of the user editable flags, syntax and spacing is important. Edit exactly as displayed in the files. ie.,
No spaces when setting variables (ftpflag=0 not ftpflag = 0).

CD archiving information

To use the CDRECORD programs, you need to change the permissions on the files. To do this, you must have root access on the magnet:

```
cd CDRECORD
become superuser (on the magnet)
chown root *
chmod 4711 *
```



This sets the sticky bit so that all users can have access to the cdrecord programs the fmriarch program will look for the CDRECORD directory on your SGI
If you plan to use CD archiving, it is important to purchase a SCSI CD that supports burnproof (a feature of the newer Sanyo drives).

If you don't use a drive supporting burnproof, CD archiving will still work, but you will likely create coasters when burning large Pfiles.

Burnproof is a Sanyo patented technology which prevents buffer underruns.

The fmriarch suite of programs has been evaluated using the following CDR device:

Sanyo BP-4 external drive 16x40x10x with burnproof

It was purchased from www.caloptic.com (around \$300)

If you set your SCSI id to 6 on your CD drive, the programs are likely to work without making any edits to the CD-R settings.

However, you should edit the CD-R settings in the extract and fmriarch routines to reflect your CDR scsi settings if they are different:

```
#-----  
#CD-R settings  
#-----  
cdrflag=1  
cdr_speed=16  
cdr_controller=1  
cdr_scsi_id=6  
cdr_target=0  
cdr_dir=$HOME/CDRECORD  
burnproof=1  
hinvdetect=0
```

Note that you don't have to change the tape or CDR settings in the USER edit section. The programs use autohardware detection to get your device information. However, you should set the cdr_controller, cdr_scsi, and cdr_target settings for your system, otherwise, everytime you run it, you will be prompted from the autohardware detection loop that your settings are wrong (it will show you the correct settings for your system).

The programs will automatically detect the speed of your CD media, and adjust the burn speed appropriately, however, it will never burn faster than what you set for cdr_speed. The system will also autodetect burnproof capability for your drive.

fmriarch

syntax: fmriarch [noftp nocd notape all dummy]

fmriarch is the Pfile archiving program. You need to start this from the unix prompt BEFORE you start acquiring your first raw data file. The program uses autohardware detection. If you have both a CD drive, and a tape device, it will preferentially write to CD. The fmriarch program is designed to burn CDs in multi mode. This means the CD is not 'closed' and more files can be added as they are acquired. However, fmriarch will only start up with a new CD (to maintain the principle of 1 CD per subject). If you insert an already burned CD, the program will kick it out and prompt you to insert a new CD. You can use the command line flags to disable ftp, cd, and tape writing.



noftp - disable autoftp
nocd - disable CD archiving
notape - disable tape archiving
all - write to all devices found (CD and tape)
dummy - use dummy mode for CD writes (laser turned off)

The dummy flag is a debugging tool which will perform CD writes in dummy mode, meaning the laser is turned off, and debugging messages are displayed. The program will crash in dummy mode after the first “dummy” write, because it will be unable to determine the laser position for the next write since nothing was burned.

extract

syntax: extract examnumber [series list] [noftp nocd notape dummy]

extract is a utility for extracting the anatomic images (i.xxx). The extract program will ask you if you want to close the CD when it is done. This will lock the CD and prevent any more writing to it. If you want to extract multiple patients to the same CD, respond no to the prompt, and after each extract, just start it again.

closedc

syntax: closedc [dummy]

If you still have an unlocked CD and wish to close it, just enter closedc at the unix prompt.

Potential problems

I performed alot of the debugging on a Solaris system, before moving it to the SGI. I have noticed that when the operating system is first brought up, the first burn is a bad one due to a SCSI transport problem. I have only seen this problem with Solaris thus far, however, I was worried that this might happen on the SGI. In order to get around this, fmriarch performs a “dummy burn” when “initializing” your CD in which it goes through all the steps of a burn, with the laser off. It uses an actual Pfile I have included in the CDRECORD directory (burntestfile) which is large enough to bring out any problems. If everything checks out, you won’t notice anything. However, if there is a problem, the CD will be ejected and you will get a message to check your CD. When there is a SCSI problem, even with dummy mode enabled, the CD will get burned with a thin stripe. You can tell this by removing your CD, turning it around and looking for this stripe. If it is not there, then just stick the CD back in, and fmriarch will work fine. If you do see this stripe, then that CD is a coaster, and you need to insert a new CD. fmriarch will not need to be restarted.

Some useful scripts

This distribution includes some nice utility kshell scripts which can be used for other purposes:

imenu

A kshell menu program.

#Syntax: imenu choices=choice1,choice2,choice3,... [title=Select a Subdirectory]

[filename=filename] [-other]

#This will display ‘title’ as your message, and give the user an option of choices

#-other: set the other keyword if you want to allow a user defined response as 1 of the choices

#filename: Supply a filename for the response to be written to a file. Your calling programs can then read and remove this file.



autoftp

```
An automated ftp program
#ftptest='autoftp $remotehost $user $pass $remotedir $remotesubdir(will create)
$hostdir $filename_to_ftp [-testflag]'
#to ftp the file junk located in the directory /export/home/DATA
#to remote host moog for user fmri with password abracadabra
#into remote subdirectory MYDATA located in directory /fmri/fmri/DATA on moog:
remotehost='echo moog'
user='echo fmri'
pass='echo abracadabra'
remotedir='echo /fmri/fmri/DATA'
remotesubdir='echo MYDATA'
hostdir='/export/home/DATA'
filename_to_ftp='echo junk'

autoftp $remotehost $user $pass $remotedir $remotesubdir $hostdir $filename_to_ftp
```

SOLARIS/SUN issues

These routines have been written for the SGI platform (GE LX scanners). However, they were created and debugged on SOLARIS, and I have included some OS checking to change some of the defaults appropriately. If you want to run any of these on SOLARIS you need to keep in mind the following:

/bin/sh will not work properly

use /bin/ksh or /usr/xpg4/bin/sh

The default tr and awk will not work properly: use /usr/xpg4/bin/tr and /usr/xpg4/bin/awk

For tr and awk, I have implemented some OS checking, so you shouldn't have to make any changes.

CDRECORD issues:

CDRECORD will need to be compiled for SOLARIS. I have some precompiled binaries, so you can contact me if you need them. For CDRECORD to work properly on SOLARIS, you will need to change the SCSI driver to scg. This is done through a package add procedure. The documentation is on the CDRECORD web site

(www.fokus.gmd.de/research/cc/gclone/employees/joerg.schilling/private/cdrecord.html). In

addition, everytime you use CDRECORD on SOLARIS, you will have to disable automounting. SOLARIS tries to mount the CD as soon as it is inserted. This locks up the CD device driver.

To disable automounting:

become su

/etc/init.d/volmgt stop

To re-enable automounting:

become su

/etc/init.d/volmgt start



APPENDIX V (Describing the methods for a paper)

The following is a short blurb which describes the methodology used here for purposes of publications, complete with references. You don't have to describe it this way.

Imaging consisted of a sagittal T1-weighted localizer, followed by a T1-weighted acquisition of the entire brain performed in the axial plane (24 cm FOV, 256x256 matrix, 3 mm slice thickness). This sequence was used both for anatomic overlays of the functional data, as well as for spatial normalization of the data sets to a standard atlas. Functional imaging was performed in the axial plane using multislice gradient-echo echo planar imaging with a field of view of 24 cm (frequency) x 15 cm (phase), and an acquisition matrix of 64x40 (21 slices, 5 mm thickness, no skip, TR = 2000, TE = 40). This sequence delivered an effective voxel resolution of 3.75 x 3.75 x 5mm.

The fMRI raw echo amplitudes were saved and transferred to a SUN Ultrasparc workstation (SUN Microsystems, Mountain View, CA) for off-line reconstruction using software developed in IDL (Research Systems Inc., Boulder Colorado). Correction for image distortion and alternate k-space line errors was performed on each image on the basis of data acquired during phase-encoded reference imaging (Alsop, 1995). Statistical parametric maps (SPMs) were generated using SPM99 (Friston et al., 1995a; Friston et al., 1995b) (from the Wellcome Dept. of Cognitive Neurology, London, UK) implemented in Matlab (The Mathworks Inc. Sherborn MA, USA), with an IDL interface. The T1-weighted images were normalized to a standard template in Montreal Neurological Institute (MNI) coordinate space within SPM99. The functional data sets were motion corrected (intra-run realignment) within SPM99 using the first image as the reference. The functional data sets were normalized to MNI space using image header information to determine the 16-parameter affine transform between the functional data sets and the T1-weighted images (Maldjian et al., 1997), in combination with the transform computed within SPM99 for the T1-weighted anatomic images to MNI space. The normalized data sets were resampled to 4x4x4 mm within MNI space using sinc interpolation. A second realignment step (inter-run realignment) was then performed (within SPM99) between successive normalized runs within each subject, using the initial normalized run as the target. This was done to eliminate motion between the successive runs within each subject. The data sets were smoothed using a 8x8x10 mm Full Width at Half Maximum Gaussian smoothing kernel, and SPMs were generated using the general linear model within SPM99. {For blocked paradigms: A 6 second time-shifted box-car waveform was used as the reference paradigm, and the ANCOVA model with global activity as a confound was employed for the statistical analysis.} {For single event paradigms: A canonical hemodynamic response function with time and dispersion derivatives were employed as basis functions, with proportional scaling of the image means.} Temporal smoothing, detrending and high-pass filtering were performed as part of the SPM analysis. {Intrinsic correlations were removed by estimating the auto-correlation structure using an AR(1) first order auto-regressive equation.} The SPM{t} were transformed to the unit normal distribution SPM{Z} and thresholded at a $P < 0.001$, corrected for spatial extent ($P < 0.05$), using the theory of gaussian fields as implemented in SPM99.

A second-level analysis was performed to generate group SPMs using a random-effects model within SPM99 with the individual contrast maps (Holmes and Friston, 1998). {While a fixed-effects analysis could have been employed (as implemented in SPM96), the resulting maps would only be valid for the group of subjects studied (ie., a pseudo-subject analysis). The



random-effects model takes into account intersubject variance in generating the group maps, permitting population-level inferences. The procedure essentially performs a 1-way T-test using individual contrast images computed in the single subject analysis as input.} The resulting maps were transformed to the unit normal distribution SPM{Z} and thresholded at a $P < 0.001$, corrected for spatial extent ($P < 0.05$), using the theory of gaussian fields as implemented in SPM99. Anatomic regions were automatically defined using an anatomic MRI atlas (Kikinis et al., 1996) which we have previously normalized to the same MNI SPM template for use with our fMRI data. MNI coordinates were converted into the Talairach coordinate system (Talairach and Tournoux, 1988) using a non-linear transform (Duncan et al., 2000). Brodmann areas (BA) were determined for activation maxima using the Talairach Daemon (Lancaster et al., 1997; Lancaster et al., 2000).

References

- Alsop DC. Correction of ghost artifacts and distortion in echo-planar MR imaging with an iterative image reconstruction technique (abstr). *Radiology* 1995; 197: 388.
- Duncan J, Seitz RJ, Kolodny J, Bor D, Herzog H, Ahmed A, et al. A neural basis for general intelligence. *Science* 2000; 289: 457-60.
- Friston K, Ashburner J, Poline J, Frith C, Heather J, Frackowiak R. Spatial registration and normalization of images. *Human Brain Mapping* 1995a; 2: 165-189.
- Friston K, Holmes A, Worsley K, Poline J, Frith C, Frackowiak R. Statistical parametric maps in functional imaging: A general approach. *Human Brain Mapping* 1995b; 2: 189-201.
- Holmes A, Friston K. Generalisability, Random Effects & Population Inference. *NeuroImage* 1998; 7: s754.
- Kikinis R, Shenton M, Iosifescu D, McCarley R, Saiviroonporn P, Hokama H, et al. A digital brain atlas for surgical planning, model driven segmentation and teaching. *IEEE Trans visualization computer graphics* 1996; 2: 2323-241.
- Lancaster JL, Summerlin JL, Rainey L, Freitas CS, Fox PT. The Talairach Daemon, a database server for Talairach Atlas Labels. *NeuroImage* 1997; 5: S633.
- Lancaster JL, Woldorff MG, Parsons LM, Liotti M, Freitas CS, Rainey L, et al. Automated Talairach atlas labels for functional brain mapping. *Hum Brain Mapp* 2000; 10: 120-31.
- Maldjian J, Schulder M, Liu W, Mun I, Hirschorn D, Murthy R, et al. Intraoperative functional MRI using a real-time neurosurgical navigation system. *JCAT* 1997; 21: 910-912.
- Talairach J, Tournoux P. Co-planar stereotaxic atlas of the human brain. 3-dimensional proportional system: an approach to cerebral imaging. New York: Thieme Medical Publishers, 1988.



APPENDIX VI (FAQ)

Where do I get software upgrades?

Open a web browser to the following address:

How do I compile the programs?

Open a command window and enter:

idlspm

Alternatively:

Start idl. Then enter:

@bootfmri

@bootspm

How do I recover from an error?

retall

How do I get the fmri_utility widget up?

data_control

How do I convert the .im file to analyze?

im2analyze

How do I convert a .loc to analyze?

locs2analyze,filename = locfile

How do I resample a .loc into the space of another .loc ?

mt = find_object2target(object = loc1, target = loc2,./compute, outv = newloc)

Your resampled images are returned in newloc.

The forward 4x4 affine transform is returned in mt.

How do I really resample images?

First you need to compute the forward 4x4 transformation matrix:

mt = find_object2target(object = object, target = target)

If there is pathway between the object and target files (that is, if they are from the same session or there is a normalized intermediary file), this routine will return the 4x4 transform from object to target space in mt.



To apply the transform:

```
outv = tran3d(volumein = object_images, matrix = mt, outmatrix =  
[target_x,target_y,target_z]/ step,step = step, start = start)
```

where step = 1 if you want to resample to target dimensions, or some resampling fraction if you want to change the output dimensions. If you want to reduce the output dimensions by 1/2 in x and y, but maintain the z target size, then step = [2,2,1] .

How do I view the contents of the .im file?

```
restorest56, imfile, stack  
imviewer,images = stack(*,*,*,volume)  
or  
imviewer,images = stack(*,*,slice,*)
```

How do I view the contents of a .loc file?

```
lh = read_header(filename = locfile, get_cube = locimages)  
imviewer,images = locimages
```

How do I view the contents of an analyze file?

```
lh = read_header(filename = locfile, get_cube = locimages)  
imviewer,images = locimages
```

How do I write a .loc file?

This is a little tricky. There are 2 type of loc files. There is a .loc (which has a GE header) and a .locs (which has an IDL_SPM labheader). Although the formats are different, the read_header command is designed to return a consistent labheader structure. To write a .locs, the command is:

```
write_header, header = header, cube = cube, filename = filename
```

The write_header command will write a single binary file with header followed by the cube image data. You can use the read_header command to get a locheader and then write it out using your new image volume, however, you have to be careful to set the lh.an structure fields properly for bit depth if your new volume is not of the same datatype that you stole the header from. The read_header command returns a locheader, with an embedded analyze header (lh.an). The programs which read the .loc files use the information in the .an part of the header structure to determine datatype, number of images, etc.

How do I write an analyze file?

First, you need to get an analyze header. If your modified image data is from a previous analyze file, you can just use:

```
ah = read_header(filename = analyze_file)
```

Note that in this case, the read_header command will return the analyze header directly into ah.



If your modified image data is from a .loc:

```
lh = read_header(filename = locfile)
ah = lh.an
```

Before you write the analyze files, you must be sure the analyze datatypes and x/y/z dimension fields reflect your new image data appropriately, otherwise you will not be able to open the file. To write the files:

```
openw,unit,newfile.hdr,/get_lun
writeu,unit,ah
close_lun,unit
openw,unit,newfile.img,/get_lun
writeu,unit,newimagedata
close_lun,unit
```

How do I list the contents of a header structure?

```
lh = read_header(filename = filename)
helpstruc, lh
```

How do I normalize a locfile from the command line?

```
auto_norm, locfile = locfile
Set the noplastic keyword to turn off elastic deformations.
Set the coregister keyword for a “coregister” type normalization (instead of spm_sn3d)
Set the mi keyword for a mutual information normalization
```

How do I perform a manual normalization?

```
picktal
```

How do I coregister 2 volumes from the command line?

```
spm_coregister, object = object, target = target, object_mode = omode, target_mode = tmode
```

For a mutual information coregistration:

```
spm_coregister, object = object, target = target, /mi
```

For a manual coregistration:

```
coregister_tool
```

How do I rerun the statistical inference part an analysis?

Click on Setup_Groups

Select the snrfile from the normalized directory for each Run you want to repeat.

Save the flist

Quit the analysis when the large analysis widget comes up

Click on individuals

Click on Run_Group_Analysis

Select the flist you created above



The program will redo the statistical analysis for your runs, and put the output back into the spm_mat folder for each respective subject/run in your flist. It will overwrite your previous analysis in the spm_mat folders.

Can I hack the code to make it do something else?

Don't do it. If you want your own version of some programs, create your own folder and your own copies of the programs, and put them somewhere other than in the IDL_SPM distribution. During the upgrade process, files in the IDL_SPM folders will get overwritten, including files you may have hacked.

How do I overlay 2 volumes?

If you have already coregistered the images and they exist as variables:

```
imviewer,volume1 = volume1, volume2 = volume2
```

If you have 2 files, and the transformation pathway between them exists (ie., they are normalized, or a normalized loc exists from the same session):

```
imviewer,file1 = file1, file2 = file2
```

How do I test my knowledge of neuroanatomy?

```
reply = pickatlas()
```

How do I change the lab information at the top of the printout when I print fmri maps?

Modify the !banner system variable in the ./idlspsite/global_profile file so that it points to a .flist file with your information. See the lab_banner.flist file in the IDL_SPM97 distribution folder for the appropriate format of this file.

How do I kill a batch process?

- Click on batch_control in the fmri_utility widget
- Select the Run you want to kill
- Click on kill process

Alternatively:

From within Unix, cd to the batch directory you want to kill. Create a file in this directory called kill. You should also execute the command in the spmbatchlicense file to immediately kill any running process.

How do I restart a batch process

- Click on batch_control in the fmri_utility widget
- Select the job and run you want to start
- Click on Start_process

Alternatively:



From within Unix, cd to the batch directory and run you want to start. Execute the command quick_start.

How do I really kill the batch?

kill_process, /all

This command should only be executed as a last resort. It will kill **all** processes related to batch processing, including matlab and idl sessions. It is likely to kill your current idl session, and any non-batch related idl and matlab sessions as well, so use this with extreme caution. If you have root access, it will kill everyone else's matlab and idl processes as well (and may shut down your idl license server – I'm not sure about that though, since I haven't tried). The program will iteratively hunt down batch, matlab, and idl processes on all the cpus listed in your cpufile and kill them.

How do I get rid of the list of old batch jobs in the batch_control widget?

Delete the old jobs from the job_control file (located in your ref_functions directory).

How do I add or delete cpus on the processing farm?

Cpus are defined in the cpufile, located in your ref_functions directory.

Can I have my own cpufile?

You can have your own personal reference_function directory, cpufile and job_control file if you define them in your .global.ksh file in your ~/.idlspm directory. This is a hidden file. The default configuration is to use a common job_control, cpufile, and ref_functions directory to simplify system administration.

How do I generate those cool 3D volume renderings?

minislicer, mri = mri, pet = pet, object = object, target = target

If the volumes are already coregistered, you can just send them as variables in mri and pet (ie., the activation cube). Alternatively, you can send the object and target file names, and the program will search for the affine transformation for object2target (target should be your MRI volume).

How do I perform a functional connectivity analysis?

fcon,/multiple,/analyze,/atlas,/separate

or directly from the SPM viewer for peak voxel/cluster functional connectivity



What's up with that BORG logo?

Resistance is futile. You will be assimilated.

I've tried everything and I still can't get the damn thing to run.

Problems in installation or batch processing are usually related to:

- A) wrong matlab or idl license file definition
- B) rsh permissions are not setup properly
- C) user and group names/ids are not identical between cpus which leads to write permission and rsh problems.

1. Read the Appendix on installation.

2. Execute the following script from your home directory in a Unix command window:
./idlspm/echo_globals

Verify that all the global variables are correct. Check the idl and matlab license file definitions carefully.

3. Run the following script from the command line:

/appl/IDL_SPM97/SPM99F/SHELL_SCRIPTS/diagnostics

Verify that the output is correct and there are NO error messages. Error messages may indicate that either your matlab/idl license definition is wrong, or you have not setup your rsh permissions properly.

Log in to all your cpus in the cpufile and execute the same command. You need to check this from every cpu separately.

4. Verify that user and group names and **IDs** (VERY IMPORTANT!) are identical between cpus, and that you can write to your data directories from any cpu.

I keep getting font errors when I start IDL and core dumps when I click on any widget button.

If you are running CDE in Solaris, try setting the language to C-----Posix. This is available under the Options button of the login screen. The default US English setting will give you IDL font errors. You can also try starting using OpenWindows instead of CDE.

I upgraded matlab and now I get IDLSPM engine errors and batch mode won't work.

First make sure you can start matlab properly from an xterminal window. If not, you need to get matlab working. If matlab starts OK, then check that the following aliases are working:

matlab5 and/or matlab6

matlabx

idlx



IDL_SPM determines the location of the matlab executable for engine calls at startup based on the matlab5 and/or matlab6 aliases. Batch processing uses the matlabx alias. These aliases are defined in the following files in the .idlspsite folder:

```
global.site.ksh
cshrc.idlspm.global
cshrc.idlspm.global.linux
```

Check to make sure the aliases are correct for your new setup.

The location of the .idlspsite folder can be determined by entering the following command from your home directory: `.idlspm/echo_globals`

The line reading

```
site_globals=/fmri2/fmri/.idlspsite
```

is your .idlspsite folder. You should also check that the information on matlab directories, matlab license location, idl directories, and idl license location is correct. See the section on Upgrading to matlab6 in the Appendix for more info.