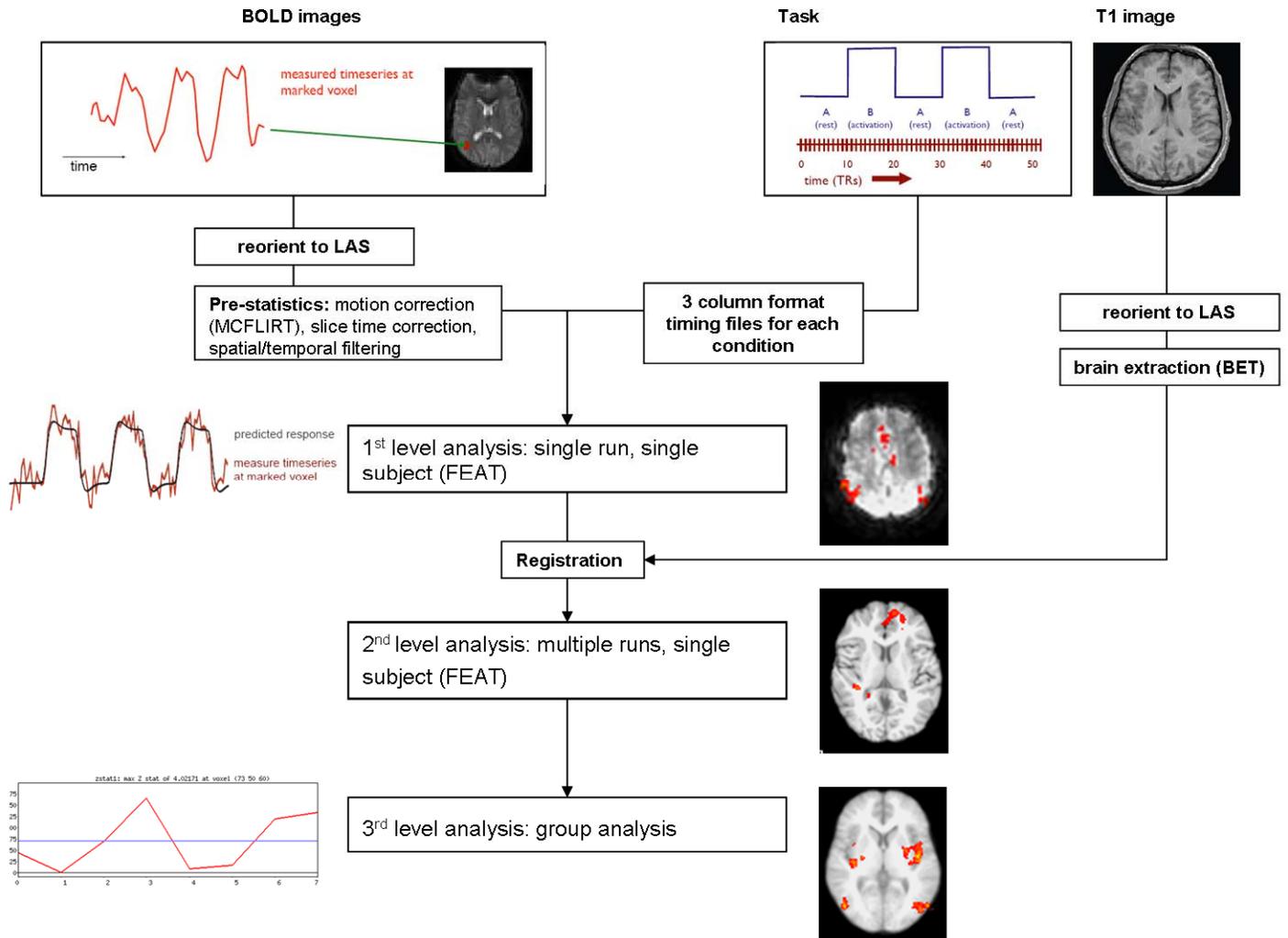


FSL Tutorial

01-07-2013

- Part I: Getting started with FSL
- Part II: FSL pre-statistics using FEAT
- Part III: FEAT 1st Level Analysis
- Part IV: FEAT 2nd Level Analysis
- Part V: FEAT 3rd Level Analysis
- Part VI: Scripting

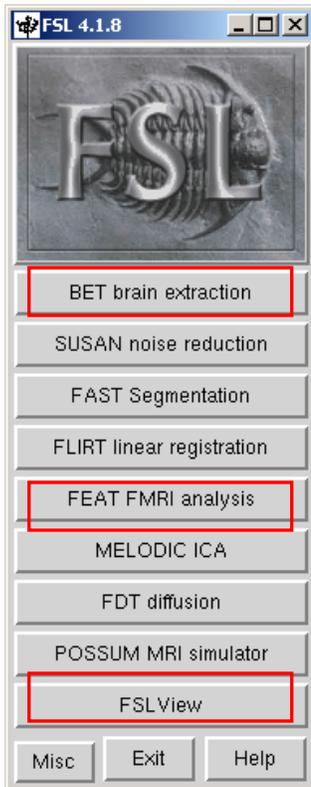


Part I: Getting Started with FSL

1. FSL Overview

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

We will be using the highlighted tools to perform simple fMRI analysis.



← Skull stripping of T1 images

← Pre-statistics and general linear model analysis of BOLD data

← Data viewer

2. Some useful Unix commands:

All these commands are entered on the command line

cd – “Change Directory”, allows you to move into a new directory from the directory you are currently in. This is similar to clicking on different folders in the Finder on a Mac, or in Windows Explorer, but you are moving folders by typing commands into the terminal instead.

For example:

```
cd `findexp Class.01`  
cd ~/experiments/Class.01  
cd Class.01/FSLtutorial/1
```

Note: You need to type everything with EXACTLY the correct paths

cd ../ - changes current directory to the directory higher up in the hierarchy. For example, if you are in ~/experiments/Class.01 typing **cd ../** will take you to ~/experiments

pwd – “Print Working Directory” this command will display the directory you are currently in. This is helpful if you get lost while navigating through your folders.

ls - list the contents of the current directory

3. Preparing BOLD images for analysis

BOLD images at BIAC are in DICOM format and come with a .BXH header. From the Finder, look in the RUN1 folder in your computer folder and make sure you see the .bxh header (called FUNC4_V.bxh). You can open this header in Wordpad, these headers contain all the information about the scans (image size, TRs, sequence used etc.) The other images are the DICOM images.

BIAC images need to be reoriented from LPS orientation to LAS orientation. FSL templates are LAS and registration in FSL will only do rotations, so starting in LAS orientation will prevent registration errors in the future. We will use the BXH tools to reorient the data. For more information on the BXH tools, see <http://www.biac.duke.edu/home/gadde/xmlheader-docs/>

You can also type `bxhreorient --help` to get information on the tool from the command line. You can do this with any of the bxh tools.

Type **pwd** (print working directory) on the command line to make sure you are in the Class.01/FSLtutorial/# folder where # is your computer number. If not, type on the command line:

cd ~/experiments/Class.01/FSLtutorial/# where # is your computer number

Reorient from LPS to LAS orientation using the command **bxhreorient** by typing while you are in the Class.01/FSLtutorial/# folder (where # is your computer number):

bxhreorient --orientation=LAS RUN1/FUNC4_V.bxh RUN1/run01.bxh

This will create both a .bxh header called run01.bxh as well as a .nii.gz file for functional run 1. FSL uses .nii.gz formatted data. You would need to do this for all 5 runs (for example **bxhreorient --orientation LAS RUN2/FUNC4_V.bxh RUN2/run02.bxh** etc) so that you create run01.bxh-run05.bxh and run01.nii.gz – run05.nii.gz, except we have already done this for runs 2-5.

4. Preparing T1 images for analysis

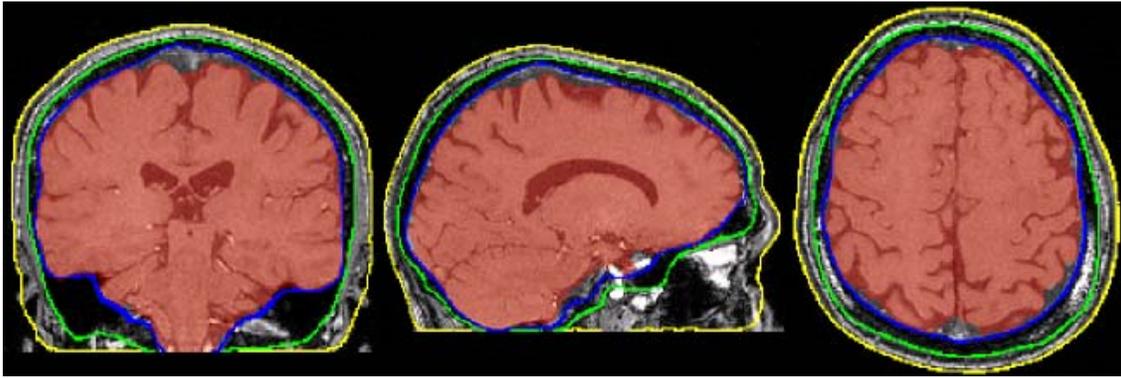
T1 images at BIAC are in DICOM format and come with a .BXH header (here called ANAT1_Whole_brain.bxh)

Reorient from LPS to LAS bxhreorient by typing:

bxhreorient --orientation LAS ANAT1/ANAT1_Whole_brain.bxh ANAT1/reoriented_anat.bxh

You still need to be in the ~/experiments/Class.01/FSLTutorial/# folder. This will create both a .bxh header as well as a .nii.gz file called reorientated_anat.nii.gz.

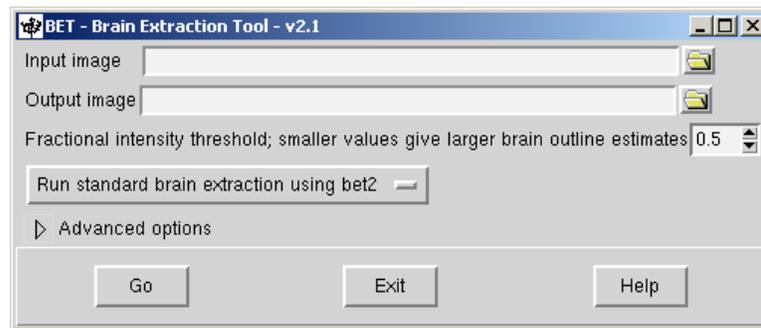
You also need to remove the non-brain material from the T1-image using FSL's brain extraction tool (BET).



The above image is from the BET research page: <http://www.fmrib.ox.ac.uk/analysis/research/bet/>. Here you can see that the brain is separated from the skull by the blue line.

To access the BET GUI type: **fsl &**

The GUI will pop up. Select **BET brain extraction** by clicking the button on the main FSL GUI. The BET GUI will pop up:



The input image should be your NIFTI (reoriented_anat.nii.gz) formatted T1 image. The output image, by default, will be the name of the input image with **_brain** appended (reoriented_anat_brain.nii.gz). An important option to be aware of is the fractional intensity threshold. Changing **Fractional intensity threshold** from its default value of 0.5 will cause the overall segmented brain to become larger (<0.5) or smaller (>0.5). This threshold must lie between 0 and 1. There are other useful options, please see the BET research page user guide for more information. <http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/BET/UserGuide>

For our purposes here, the default 0.5 should be fine.

Close the BET GUI.

5. Inspecting raw data in FSLview

Open FSLview by typing **fslview &** on the command line, or by clicking the FSLview button on the main FSL GUI.

Go to File → Open in the FSL GUI and select your BOLD data. Remember FSL uses .nii.gz images, so you can load run01.nii.gz from the RUN1 folder. You can look at the BOLD data volume by volume by looking at in movie mode. You can check for gross motion or other artifacts this way. Close the BOLD image by clicking the x in the corner of the window (but don't close FSLview!)

After closing the BOLD data, also load the reoriented_anat.nii.gz image and reoriented_anat_brain.nii.gz from the ANAT1 folder to make sure the brain extraction looks ok.

File → Open → reoriented_anat.nii.gz

Overlay the brain extracted T1 image on top:

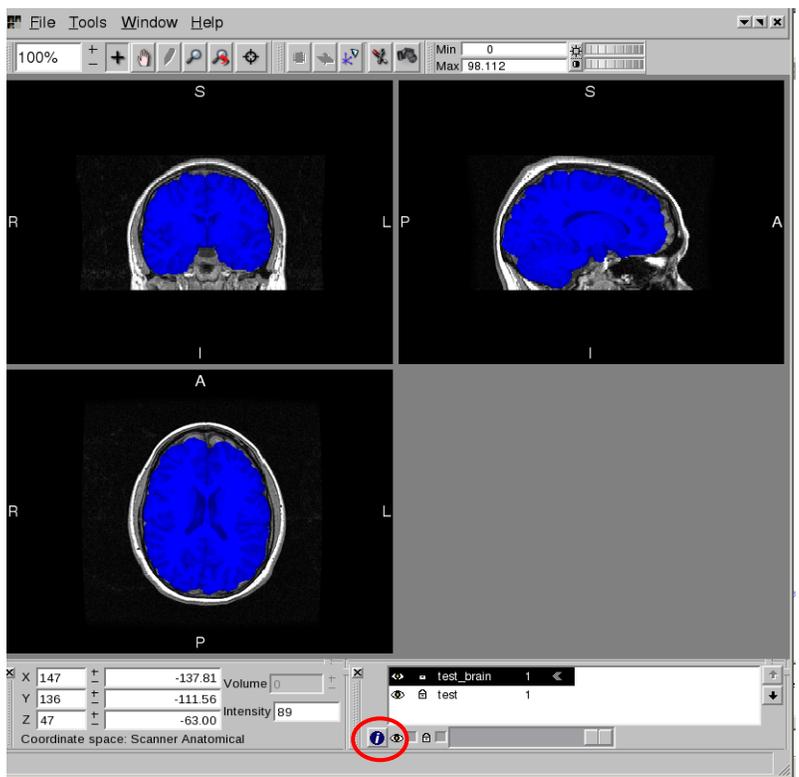
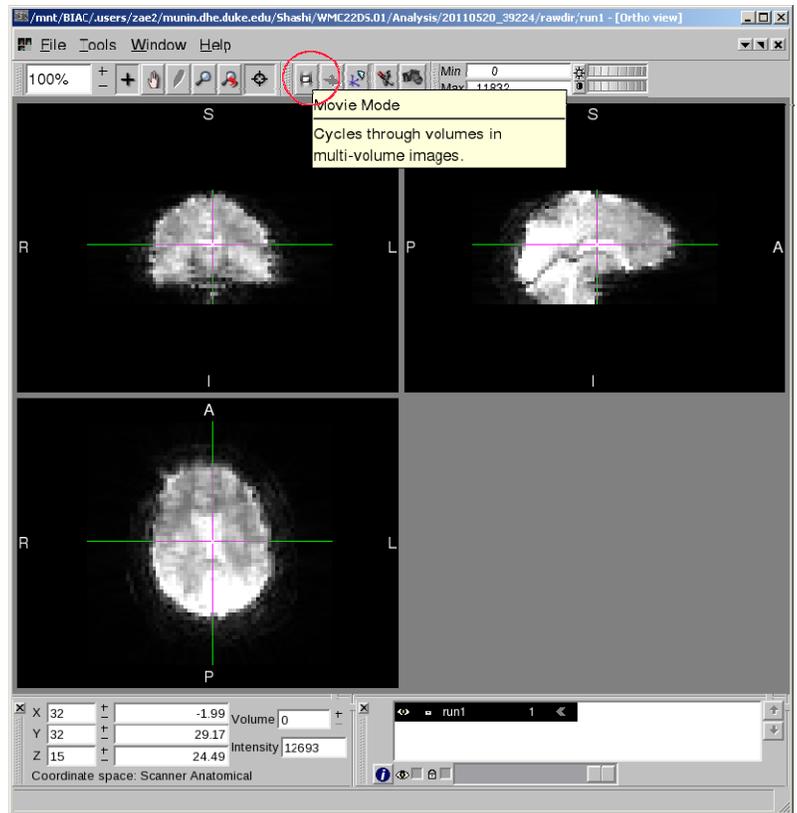
File → Add → reoriented_anat_brain.nii.gz

Change the color of the brain extracted image to blue:

Select the reoriented_anat_brain.nii.gz image on the bottom right window. Click the  button. In the **Lookup table options** pull down menu, change the color.

You can now see the brain extracted image over the T1 image so you can see how well your brain extraction turned out. If you want to improve it, rerun BET with different options (for example changing the fractional intensity).

Note: you only need to brain extract the T1 image, the BOLD data is brain extracted automatically by FEAT when you run the pre-statistics portion of the analysis.



6. Specifying event timing

FSL uses three column tab-delimited text files describing the timing of your events. The first column is the onset, the second the duration of the event, and the third a weighting factor (usually 1 unless for some reason you want to weight them differently). You need one file for each condition of each run for each subject (so if you have 3 conditions, 3 runs, and 20 subjects, you have $3 \times 3 \times 20$ files = 180 text files. Usually you obtain the three column formatted files directly from the output of your task. There are a bunch of tools that can convert different output file types into three column formatted files:

<http://www.biac.duke.edu/home/gadde/xmlheader-docs/> see eprime2xml, presentation2xml, showplay2xml and eventstable2xml.

Onset	Duration	Weight
30	1	1
60	1	1
90	1	1
120	1	1

We already have the three column format files set up for this task. Here you have 2 timing files files for each run in Class.01/FSLtutorial/#/Timing/run01 – run05 called ev1.txt and ev2.txt. You should open and look at one of these files in Wordpad.

You now have all the components you need to start using FEAT, FSL's general linear modeling fMRI analysis tool! To recap, you need **BOLD data in LAS orientation in NIFTI format, T1 data in LAS orientation in NIFTI format that has been brain extracted** using BET, and you need **three column formatted text files** with your event timing (one for each condition). You also need to have **inspected** your raw T1 and BOLD data and checked that your brain extraction did a good job. **You should always look at raw data.**

Part II: FEAT pre-statistics

Description of example data:

Each of these datasets consists of a set of volumes of functional data taken from one run of a motor response task done by one subject. This task is a blocked design. Throughout the experiment, the subject squeezed both hands whenever a flashing checkerboard stimulus was presented. Each functional run has 131 time points.

IMAGE PARAMETERS FOR ALL RUNS

Field Strength: 1.5 T
Sequence: Spiral Gradient Echo
Matrix size: 64 x 64
Field of view: 24
Voxel size (x, y, z): 3.75mm, 3.75mm, 5mm
Orientation: Axial
TR: 2000 ms
TE: 40
Disdaqs: 3

EXPERIMENTAL PARAMETERS

A static checkerboard is displayed for the first two TRs (4s). Alternating blocks begin with the third TR. The first block displays a flashing checkerboard for 10 TRs (20s). The second block displays a static checkerboard for 10 TRs (20s). This set of flashing/static blocks is repeated six times. After the final static block, another 6 TRs (12s) of flashing checkerboard appeared before the run ended. Ev1.txt describes the timing of the flashing checkerboard (both hands squeezed), and ev2.txt describes the timing of the static checkerboard (rest).

Before running the GLM statistical analysis to look for activation in your data, you need to preprocess the data. Pre-statistics processing includes motion correction using MCFLIRT, slice-time correction, and temporal and spatial filtering. There are other options, but they are beyond the scope of this tutorial, and can be looked at here: <http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FEAT/UserGuide#Pre-Stats>

To continue on at this point, you need to have:

- NIFTI formatted BOLD data in LAS orientation (for each runs 1-5)
- NIFTI formatted T1 data in LAS orientation that has been skull stripped using BET
- Tab delimited 3 column format text files for the timing of your data (for each runs 1-5)

To open FEAT, type: **fsl &** or if you already have the fsl GUI open, select the **FEAT** tab.

1. Data tab

In the two drop down menus at the top, you should have **First-level analysis** and **Full- Analysis** selected

Click the **Select 4D data** button and select the .NIFTI file you want to analyze. First, you should be using run01.nii.gz, which is the raw BOLD data from run 1 that has been reoriented to LAS in the RUN1 folder. Once the data is loaded, the number of time points and TRs should be filled in automatically. If for some reason the TR is not filled in automatically, change it to 2 s.

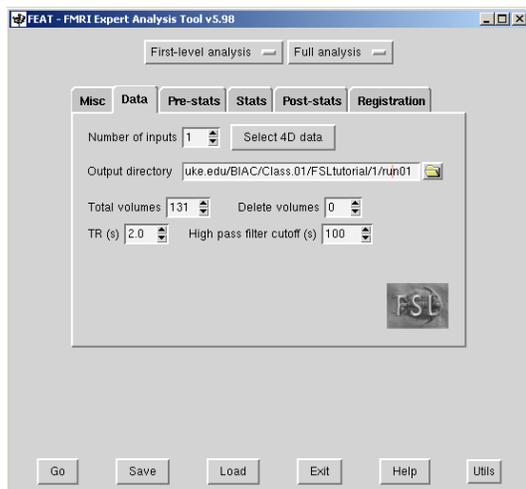
This warning may pop up:



Click OK.

Fill in the output directory by entering the foldername as : **/mnt/BIAC/munin2.dhe.duke.edu/BIAC/Class.01/FSLtutorial/#/run01** where # is your computer's folder number. This will make the output directory called run01.feats with your first level analysis for RUN1.

Leave the high pass filter cut off in seconds (for a blocked design this cut off is adequate). This is chosen to remove the worst of the low frequency trends, and is also long enough to avoid removing the signal of interest. In general you need to ensure that this is not set lower than your maximum stimulation period (in this case $20 * 2 = 40$ s).



2. Prestats tab

Leave motion correction at MCFLIRT.

Leave B0 unwarping unchecked.

Slice Timing Correction should be changed to interleaved using the drop down menu.

BET brain extraction should be checked (this refers to automatic brain extraction of the BOLD data).

Spatial smoothing FWHM (mm) should be left at 5 mm. **Spatial smoothing** is carried out on each volume of the data set separately. This is intended to reduce noise without reducing valid activation; this is successful as long as the underlying activation area is larger than the extent of the smoothing. Thus if you are looking for very small activation areas then you should maybe reduce smoothing from the default of 5mm, and if you are looking for larger areas, you can increase it, maybe to 10 or even 15mm. To turn off spatial smoothing simply set FWHM to 0.

Intensity normalization, temporal filtering, and perfusion subtraction should be unchecked, high pass filter should be checked.

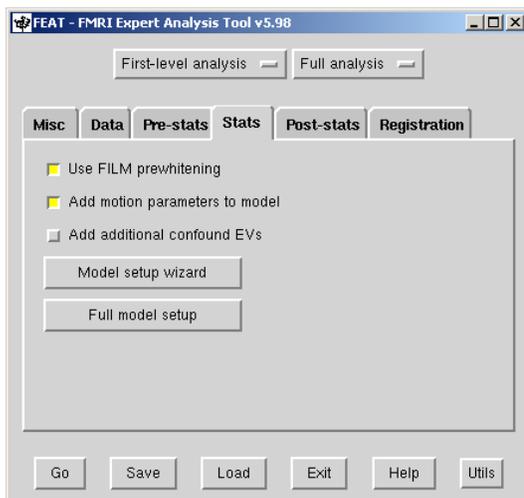
Uncheck MELODIC.

**For more information regarding these options, see <http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FEAT/UserGuide#Pre-Stats!>

Part III: FEAT first level analysis

1. Stats Tab

This tab is where the general linear model that models the expected hemodynamic response is set up. The BOLD data is then matched to this model and areas of activity are voxels where the time-course of the BOLD data is statistically similar to the model you created.



FILM prewhitening should be selected (FILM: FMRIB's Improved Linear Model). From the FSL website: FILM uses a robust and accurate nonparametric estimation of time series autocorrelation to prewhiten each voxel's time series; this gives improved estimation efficiency compared with methods that do not pre-whiten.

Add motion parameters to model should be selected. This includes the head motion parameters (as estimated by MCFLIRT motion correction in the Pre-stats processing) as a confound explanatory variable in your model. This can sometimes help remove the residual effects of motion that are still left in the data even after motion correction

Click the **Full model setup** button to set up your analysis.

Since in our task we only have two conditions to model (the flashing checkerboard with hands clenched and the static checkerboard with hands at rest), we only have two EVs, or explanatory variables. Therefore, the **Number of original EVs** should be 2. However, if you have more than 2 conditions, you add the number of conditions there.

Change the **Basic shape** to **Custom (3 column format)**. This is where you are specifying the timing of your events. Once you have changed to this three column format shape, you will load your timing file, **ev1.txt** from the **TIMING/run01** folder. Again, this ev1.txt file specifies the timing of your events with the first column being the onset time in seconds of the events, the second being the duration of the events, and the third being the weighting factor. You will do this for each EV. Each EV has its own tab, so in this case you will need to set up tab 2 in the same way as tab one, except that you will load ev2.txt.

Change the **convolution shape** to **Double- Gamma** with all the standard options using the drop down method—the convolution is the basic shape of the hemodynamic response that you want in your model. There are other shapes you can use, see here:

<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FEAT/UserGuide?highlight=%28%5CbCategoryFEAT%5Cb%29#EVs>

Uncheck the **temporal derivative** - Clicking this option shifts the waveform slightly by adding a fraction of the temporal derivative of the EVs original waveform. Doing this will make another waveform show up in the design matrix next to the EVs waveform from which it was derived.

Leave the **apply temporal filtering** checked (this means that you are applying the temporal high pass filtering that you added in the Data tab also to the model you are creating for that EV).

Give the EV a name, you don't have to do this but if you have a complicated task it is useful to label the EVs so you know which part of your task that EV is modeling. Here I called it "on" but you really can give it any name that is descriptive to you. I called EV2 "off"

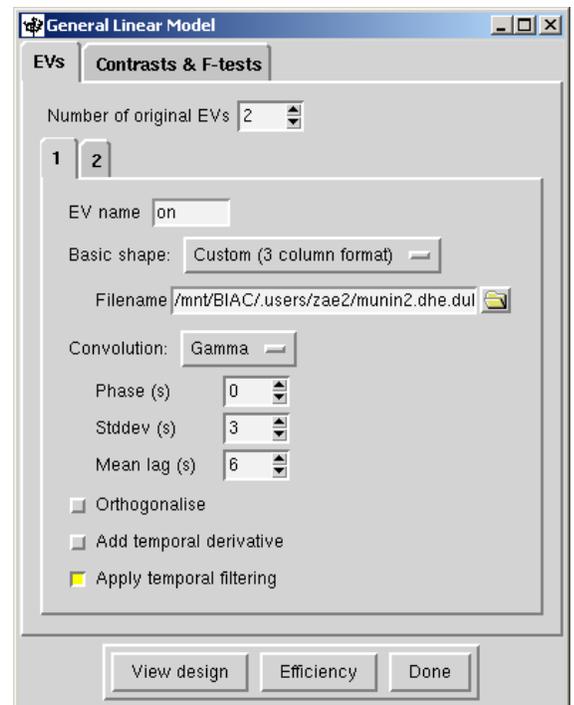
You need to set up one of these tabs for each EV! Make sure you have set up both EV tabs completely!

Important note! The best way to set up an analysis in FSL is to set up all your contrasts of interest at the first level, and then carry them up as you combine runs and groups.

You then need to set up your **Contrasts and F-tests** by selecting the Contrast and F-tests tab:

You will be setting up your contrasts and F-tests for the **Original EVs**. The other option is "real EVs" which is beyond the scope of this tutorial but more information can be found here:

<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FEAT/UserGuide?highlight=%28%5CbCategoryFEAT%5Cb%29#EVs>

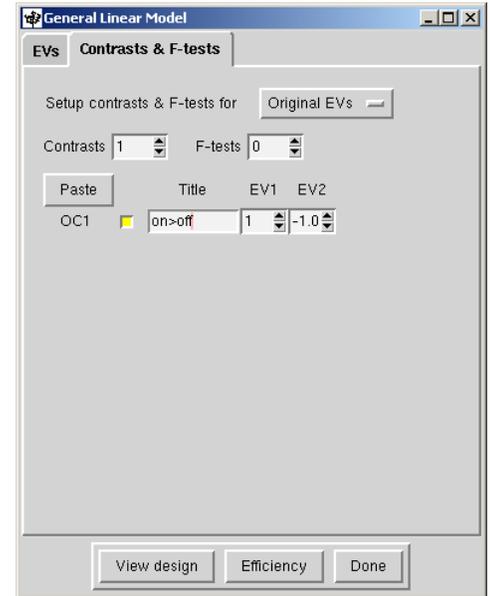


We have only two EVs and we will only have one contrast. Here we are asking the question, where is the activity in EV1 greater than the activity in EV2 (where in the brain is activity for hands clenched greater than activity for hands at rest)?

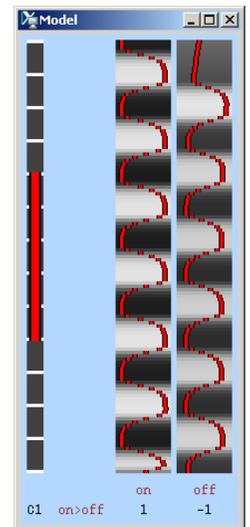
To contrast EV1 and EV2, you would give EV1 a value of 1 in this contrast, and EV2 a value of -1. In a more complex design where you require more contrast vectors, increase the **Number of contrasts**. A Z statistic image will be generated for each contrast vector. Again, give it a descriptive name so you know what is being modeled. I again called it “on>off”.

F-Tests:

If you want to investigate whether several contrasts are significantly different from zero, you can use an F Test. The F Test is not very informative, however, because you will not be able to ascertain what is significantly different from zero (it is more of an “or” statement, where does the brain respond to EV1 *or* EV2).



Click **Done**, and a picture of the model you just created will pop up. This is a graphical representation of the design matrix and parameter contrasts. The bar on the left is a representation of time, which starts at the top and points downwards. The white marks show the position of every 10th volume in time. The red bar shows the period of the longest temporal cycle which was passed by the highpass filtering. The main top part shows the design matrix; time is represented on the vertical axis and each column is a different (real) explanatory variable (e.g., stimulus type). Both the red lines and the black-white images represent the same thing - the variation of the waveform in time. Below this is shown the requested contrasts; each row is a different contrast vector and each column refers to the weighting of the relevant explanatory variable. Thus each row will result in a Z statistic image.



You can close this box now.

2. Post Stats Tab

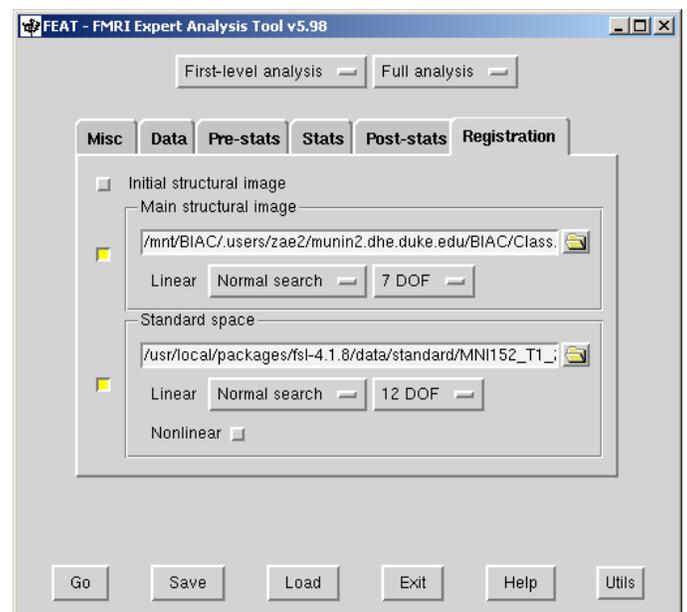
Leave all options here at the default. Look at FSL’s explanations of these options:

<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FEAT/UserGuide?highlight=%28%5CbCategoryFEAT%5Cb%29#EVs>

3. Registration Tab

Click the Main structural image box, and select the brain extracted and reoriented anatomical image (reorientated_anat_brain.nii.gz). All other options should remain the same. You only need to use Normal Search rather than Full Search because you already know the T1 and BOLD data is in the same orientation as the FSL standard template because you took care of that using bxhreorient to LAS. DOF is degrees of freedom. You can leave this as the default for now.

****Important note** about FSL registration: at the first level, the registration matrices are created (in other words the calculations that are necessary for the transformations to occur are generated) but they are not actually applied until the 2nd level. That is because data in higher resolutions takes up more space, so this is a space-saving measure. First the transformation from BOLD to T1 and from T1 to standard



is computed; ultimately the T1 to standard transformation is applied to the BOLD to T1 data to take the BOLD to standard space. However, in the first level all statistic images are still in the BOLD space because the transformations have not actually been applied yet.

**** Another important note: the T1 and BOLD data must be reoriented to the same orientation (LAS in this case) as the FSL template to get a good registration; luckily we take care of that in the first step (bxhreorient)**

Click the Go button to run your first level analysis. As the analysis is running, an FSL webpage will pop up and print out what is happening in your analysis as it goes. **Wait a moment for the analysis to run.**

5. Check out your output:

In the Finder, open the Class.01/FSLtutorial/#/run01.feats folder. This is the output of your first level fsl analysis.

The final z-statistic image with all activity that has survived thresholding and correction for multiple comparisons is called thresh_zstat1.nii.gz, in the main folder run01.feats (not in a subfolder). All of the unthresholded statistic images are in the run01.feats/stats folder. The transformations necessary for registration between the images are in the run01.feats/reg folder.

Open the thresholded z-statistic image for EV1 in fslview:

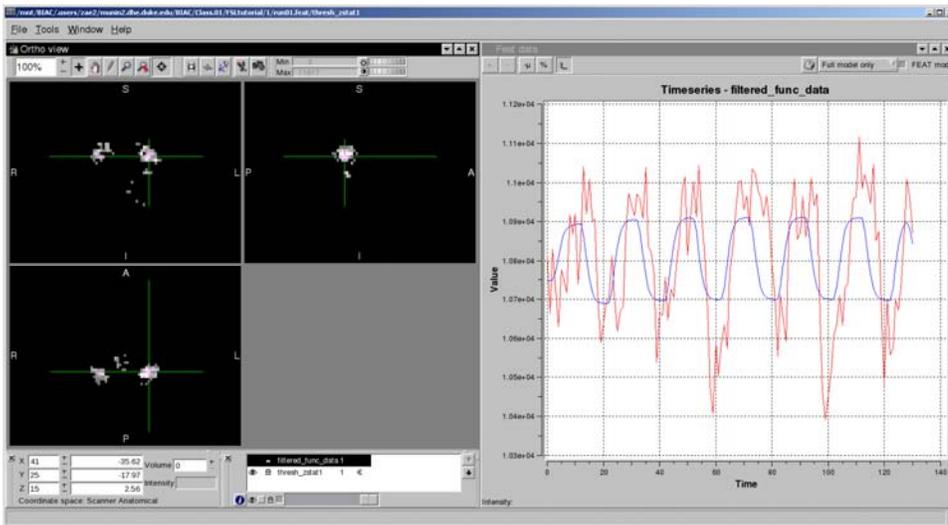
****note, you only have one thresh_zstat image because you only have one contrast, but you would have one for each contrast in a more complicated task.**

type **fslview &** on the command line, or click the FSLview button on the main fsl GUI.

File → Open → run01.feats/thresh_zstat1.nii.gz

What will be displayed is a panel with the data on the left and the timeseries and model on the right (FEAT mode). You will notice that both the z-statistic image and filtered_func_data are loaded (it should be loaded automatically). The filtered_func_data.nii.gz image is the BOLD data after it has been preprocessed, and therefore has 131 volumes. As you click on voxels with high significance, you will see that the data (red) better matches the model (blue). Again, this data is still in native BOLD space, it has not been transformed to standard space, so you cannot overlay it over a standard or T1 image.

Again, the red line here is the filtered functional data (filtered_func_data.nii.gz) which is the data that is actually compared to the model; it has been processed already.



Another important output to note are the report output files (.html). There is one for each stage of the first level analysis inside the main run01.feats folder.

First open **report_prestats.html** by double clicking. This is a summary of the motion correction. You would use this to exclude subjects or identify time points with a lot of motion.

If you select **report_poststats**, you will see a summary of activity for that run under Thresholded activation images. As I said before, the statistics at this level are still in native space. If you click on the statistic images, you will get a list of clusters and information about each cluster such as location of center of mass and maximum z-score.

There are also time series plots for the voxel with maximum activity (as assessed by the highest z-score). The red line is the data; the blue line is the full model. The best match between the red and blue lines will give the highest z-score. The “partial model fit” is the model fit due simply to the contrast of interest and is not usually easily interpretable unless you have simple non-differential contrasts.

Report_reg.html is the registration report. There are example images of each of the transformations (BOLD→T1, T1→Standard, BOLD→ Standard). All of the calculations for these transformations have been done, however the transformations have not been applied at this level and all data is in native space.

You have completed a level one analysis for run 1. We already ran the analysis for runs 2-5, so you don’t have to do it again. But it would be done exactly the same way.

Part IV: FEAT second level analysis

A second level analysis combines all the runs for a single subject.

Open the FEAT GUI by typing **fsl &** on the command line, and then click the FEAT button.

1. Data Tab:

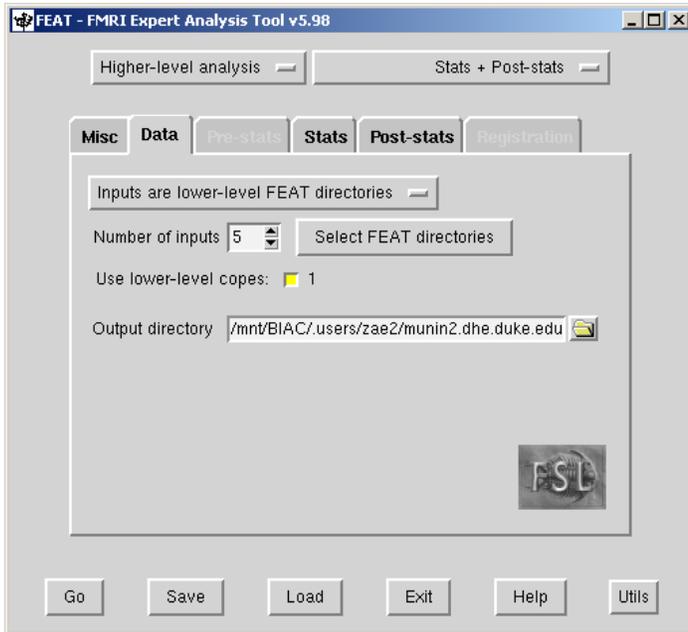
Change the top dropdown menu to **Higher level analysis**.

In this case, we are going to leave the **Inputs are lower-level FEAT** directories as-is. You can also load lower level cope images instead of entire directories.

Create an output directory in `/mnt/BIAC/munin2.dhe.duke.edu/BIAC/Class.01/FSLtutorial/#` (where # is your computer number). Call it something like 2ndLevel so that you know it's the second level analysis for this subject. The output will be in a folder called 2ndLevel.gfeat.

The **number of inputs** is going to be 5, since you have 5 runs.

Click **Select FEAT directories** and select run01.feats – run05.feats for that subject (your computer number)..

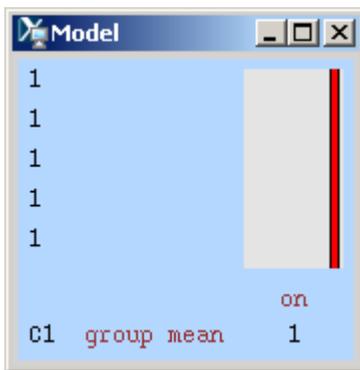
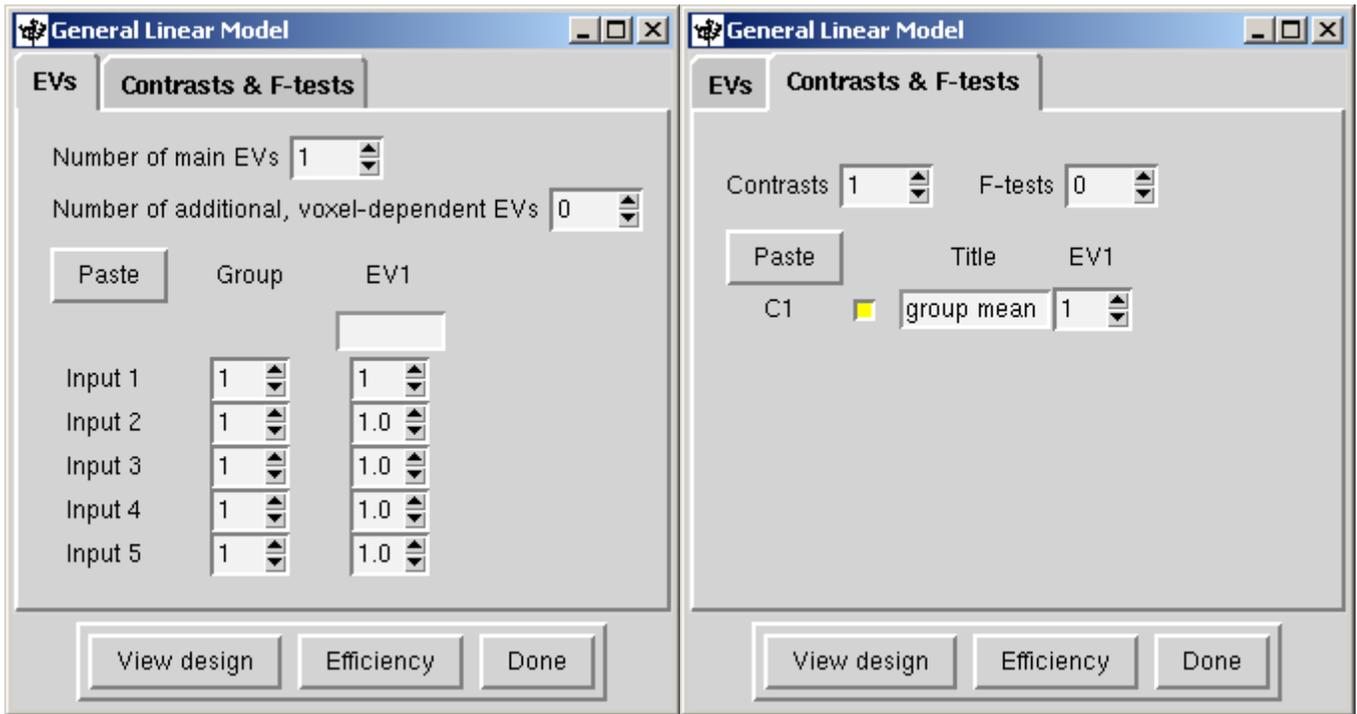


There will be only 1 cope box checked next to **Use lower-level copes** because we only have one contrast (cope means **contrast of parameter estimates**) but if you had more contrasts, you will have more copes.

2. Stats Tab

Here we are going a group average of the 5 runs. We are going to use Fixed effects (as opposed to Mixed Effects, in Fixed Effects reported activation is with respect to the group of sessions or subjects present, and not representative of the wider population. Fixed effects ignores cross-session/subject variance)

Click the **Full model setup** button. You will have only one contrast, the group mean of all the runs on>off contrast. We are asking the question “where on average across all 5 runs is there activity for the contrast on> off?”



3. Poststats Tab

Same as in first level, leave as default

Run this analysis by clicking Go.

4. Output

Open 2ndLevel.gfeat folder. There will be a cope.feat folder within this folder for each contrast. Here we only have one contrast, so there is only one cope.feat folder (cope1.feat) The thresh_zstat1.nii.gz in the cope1.feat folder contains the voxels that are significantly active over all 5 runs for cope 1.

Look at the report.html outputs for each cope (in the 2ndLevel.gfeat/cope1.feat folder). The report_poststats.html contains the summary images for activity that is significant over all 5 runs for that cope. There is no report_prestats.html or report_reg.html because that was done in the first level.

Load thresh_zstat1.nii.gz in fslview.

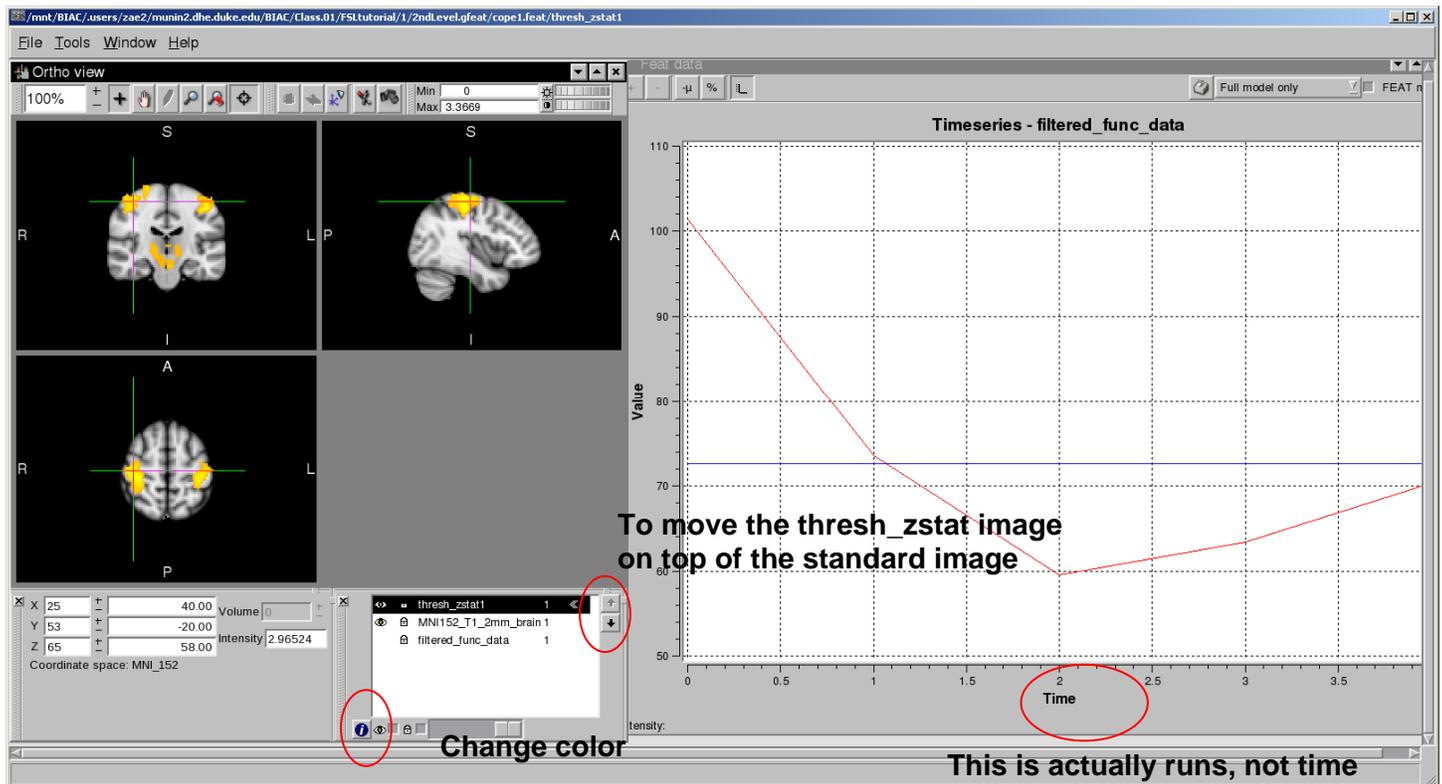
File → Open → 2ndLevel.gfeat/cope1.feat/thresh_zstat1.nii.gz

Now the data is in standard space, so you can add the standard image to overlay it onto.

File → Add standard → MNI152_T1_2mm_brain.nii.gz

In the left window in the list of images, move the thresh_zstat1 image to the top, and change the color to Red-Yellow by pressing the “i” button and changing it in the look up table options drop down menu.

The voxels that are colored now are the voxels that are significantly active on average over 5 runs. In this case, the model isn't the timeseries of the data; the model represents representing the mean across the runs. The panel on the right shows in red the data, or the activity for each of the runs (in this case the x-axis is not Time, it is Runs)



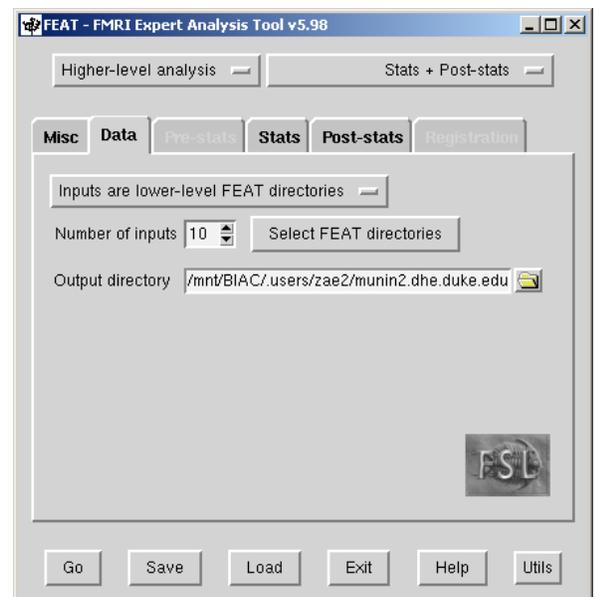
**Note: In this case, the filtered_func_data.nii.gz has 5 volumes, it is the concatenated statistics for each of the runs

Part V: Feat third level analysis (group analysis)

Here we are going a group average of multiple subjects. We are carrying up the analysis of where brain activity while squeezing hands is greater than when not hands at rest first to the 2nd level (across runs) and now to the group level.

1. Data Tab

First be sure that the drop down menus on the top are set to Higher-level analysis and Stats+Poststats. **Here the inputs will be lower level FEAT directories, in this case the cope1.feats directories inside the 2nd level directories of each of your subjects.** Remember there is a cope#.feats directory for *each contrast* that you have at the second level. This means that you will have a 3rd level analysis for each contrast. In this case, we only have one contrast at the second level (cope1.feats is on>off) so you will only have 1 third level analysis.



For this portion of the tutorial, you will not actually be able to do the analysis, since we only have data from one subject. In the case of a real third level analysis, the Number of Inputs should be set to the number of subjects. We will pretend that we have 10 subjects.

Click Select FEAT directories and load in the second level analysis from each subject.

Also enter an Output directory (for example /mnt/BIAC/munin2.dhe.duke.edu/BIAC/Class.01/FSLtutorial/#/3rdLevel). Output will then be in FSLtutorial/3rdLevel.gfeat.

2. Stats Tab

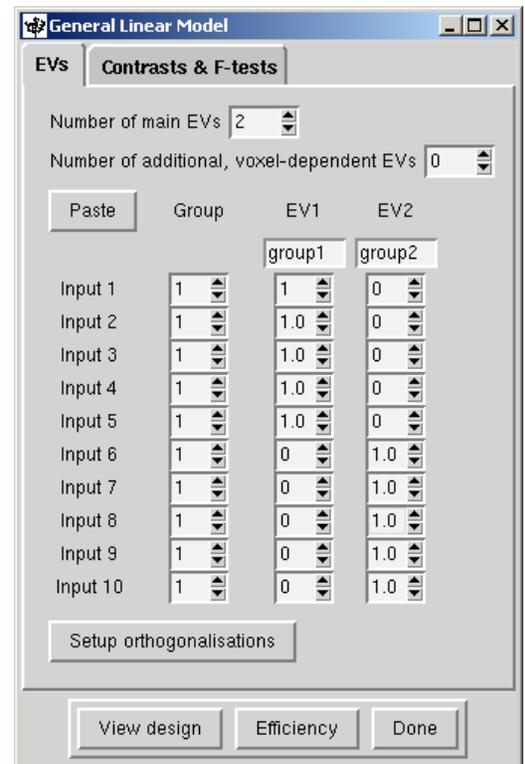
We are going to use **Mixed effects** (as opposed to Fixed Effects, Mixed effects statistics can be used to make inferences about the wider population). Be sure that the drop down menu at the top of this tab is set to Mixed effects: FLAME 1. FLAME stands for FMRIB's Local Analysis of Mixed Effects. Here is more information on the statistics used by FSL: http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FEAT/UserGuide#Group_Statistics

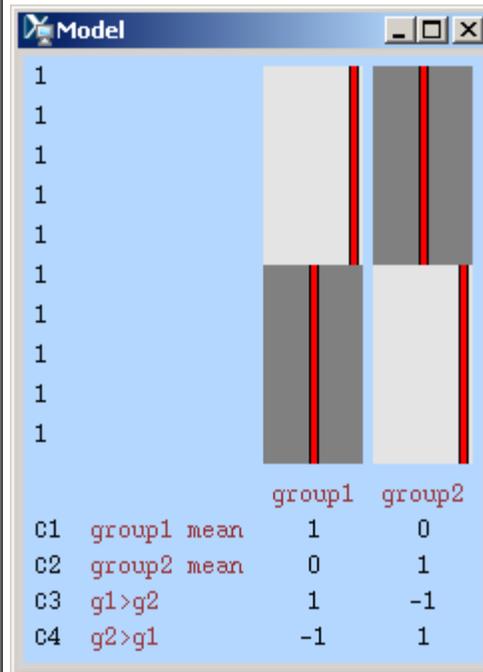
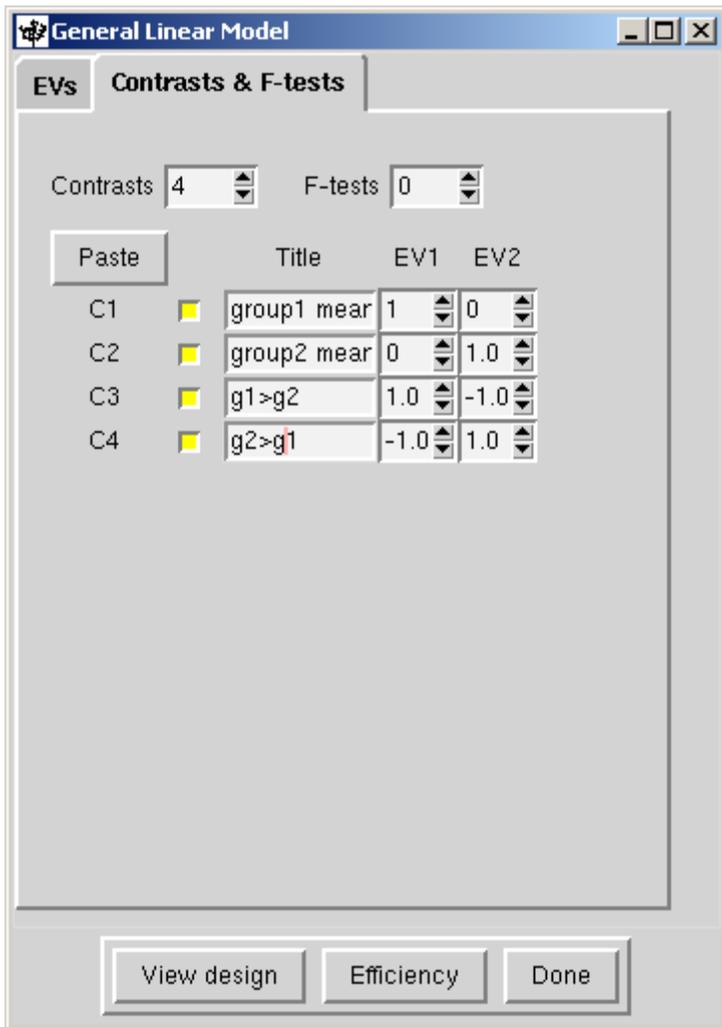
Click the **Full model set-up button**. Here we will specify the contrasts between groups of subjects that we would like to make. For this analysis, we will pretend that we have two groups of subjects, group1 and group2 and the first 5 subjects belong to group1 and the second 5 belong to group2. Therefore, we will have two EVs, one for each group. If we want to see which group activates more during the task, we would set up the EVs as such:

This is a very important note about group assignment (the column labeled Group with a number for each input):

If you ask for more than one group, each group will end up with a separate estimate of variance for the higher-level parameter estimates; for example, if the first 5 inputs are second-level FEAT outputs from group1 and the next 5 inputs are second-level FEAT outputs from group2 you can setup two different groups and each will end up with its own variance estimates, possibly improving the final modeling and estimation quality (). If you setup different groups for different variances, you will get fewer data-points to estimate each variance (than if only one variance was estimated). **Therefore, you only want to use this option if you do believe that the groups possibly do have different variances.** Unless your groups have widely different characteristics and therefore should have different variances, you should keep everyone in the same group with one variance estimate.

You then need to set up the contrasts. We will have four contrasts, the mean activation of group1, the mean activation of group2, as well as activation that is greater in group1 than group2, and activation that is greater in group2 than in group1:





This set up will answer the question:

Where in the brain is activity to hands squeezed greater than hands not squeezed 1.) on average across all subjects in group1; 2.) on average across all subjects in group2; 3.) in group1 more than group2; 4.) in group2 more than group1.

3. Post-stats Tab

Leave these options on default.

Part VI: Scripting

Running a FEAT first level Analysis Using a Script and Template:

Bash scripts can be used to automate an FSL analysis for multiple runs/subjects. This is useful if you are planning on running the same analysis over and over again for each run or subject. The basic idea is that you take the analysis template, the .fsf file, created when the FSL analysis is set up and replace all the hardcoded path names in the template with variables that are then manipulated using a script. The script then reads the template and replaces the variables with new paths, and then runs the FSL analysis. This is much faster and less prone to human error than trying to analyze all your runs and subjects through the GUI.

You can open the .fsf file in Wordpad. For example, here I opened from run01.feat (first level analysis for run 1) the file design.fsf file, it looks like this:

```
# FEAT version number
set fmri(version) 5.98

# Are we in MELODIC?
set fmri(inmelodic) 0

# Analysis level
# 1 : First-level analysis
# 2 : Higher-level analysis
set fmri(level) 1

# Which stages to run
# 0 : No first-level analysis (registration and/or group stats only)
# 7 : Full first-level analysis
# 1 : Pre-Stats
# 3 : Pre-Stats + Stats
# 2 :           Stats
# 6 :           Stats + Post-stats
# 4 :           Post-stats
set fmri(analysis) 7

# Use relative filenames
set fmri(relative_yn) 0

# Balloon help
set fmri(help_yn) 1

# Run Featwatcher
set fmri(featwatcher_yn) 1

# Cleanup first-level standard-space images
set fmri(sscleanup_yn) 0

# Output directory
set fmri(outputdir)
"/mnt/BIAC/.users/zae2/munin2.dhe.duke.edu/BIAC/Class.01/FSLtutorial/1/run01"

# TR(s)
set fmri(tr) 2.0

# Total volumes
set fmri(npts) 131

... etc
```

The highlighted part is the path to the output directory, which when you set up the first level analysis using the GUI, you entered in. The .fsf file contains all the same information as the GUI.

Again, the basic idea of setting up a script to run the FSL analysis is that you **replace these hardcoded paths to files and folders with variables** that are manipulated by a bash script.

The first step is to replace all the hardcoded paths with variables.

For instance, I will replace this hardcoded output path with a variable: **OUTPUTDIR**. It is useful to use all capital letters so that it sticks out and is easy to find in the template.

```
# Output directory
```

```
set fmri(outputdir)
"OUTPUTDIR"
```

You will do this for all hardcoded paths in the template that you saved. There are a bunch of instances where you need to put variables in the place of paths for a 1st level analysis:

1. replace the output directory with a variable: ei OUTPUTDIR as shown above
2. the 4-dimensional BOLD data that is being analyzed:

```
# 4D AVW data or FEAT directory (1)
set feat_files(1)
"/mnt/BIAC/.users/zae2/munin2.dhe.duke.edu/BIAC/Class.01/FSLtutorial/1/RUN1/run01"
```

replaced with:

```
# 4D AVW data or FEAT directory (1)
set feat_files(1)
"DATA"
```

3. The path to the structural image:

```
# Subject's structural image for analysis 1
set highres_files(1)
"/mnt/BIAC/.users/zae2/munin2.dhe.duke.edu/BIAC/Class.01/FSLtutorial/1/ANAT1/test_brain"
```

replaced with:

```
# Subject's structural image for analysis 1
set highres_files(1)
"ANAT"
```

4. The paths to each of the three column format files (there are two)

```
# Custom EV file (EV 1)
set fmri(custom1)
"/mnt/BIAC/.users/zae2/munin2.dhe.duke.edu/BIAC/Class.01/FSLtutorial/1/TIMING/run01/ev1.txt"
```

replaced with:

```
# Custom EV file (EV 1)
set fmri(custom1) "EV1"
```

Once all the paths are replaced with variables, save the template in the Scripts folder in your computer's folder (save it as something like first_level_template.fsf). There is an example first level template in there already (called first_level_template_example.fsf).

Writing a bash script:

Important notes about scripts:

- if this symbol appears at the beginning of a line in a bash script, this line is 'commented out' and will not execute

\${VARIABLE} or \$VARIABLE- replaces whatever the value of the variable has in that position

For example:

```
STUDYDIR=~ /experiments/Class.01/FSLTutorial
```

```
DATA=${STUDYDIR}/1 # here ${STUDYDIR} is replaced by ~/experiments/Class.01/FSLTutorial
```

You can edit the script in any text editor on the cluster, but I suggest using gedit.

Type **gedit &** to open a gedit window or **gedit scriptname.sh &** to open a script that already exists.

The next step is to create the script that will replace the variables with real paths in the template and run the analysis. An annotated example of a script is shown below. This script is also in your computer's folder under Scripts.

You can open it in gedit also by typing **gedit levelone.sh &** in the Scripts folder.

```
#!/bin/sh (indicates that this is a bash script)

# This is a BIAC template script for jobs on the cluster
# You have to provide the Experiment on command line
# when you submit the job the cluster.
#
# > qsub -v EXPERIMENT=Dummy.01 script.sh args
#
# There are 2 USER sections
# 1. USER DIRECTIVE: If you want mail notifications when
# your job is completed or fails you need to set the
# correct email address.
#
# 2. USER SCRIPT: Add the user script in this section.
# Within this section you can access your experiment
# folder using $EXPERIMENT. All paths are relative to this variable
# eg: $EXPERIMENT/Data $EXPERIMENT/Analysis
# By default all terminal output is routed to the "Analysis "
# folder under the Experiment directory i.e. $EXPERIMENT/Analysis
# To change this path, set the OUTDIR variable in this section
# to another location under your experiment folder
# eg: OUTDIR=$EXPERIMENT/Analysis/GridOut
# By default on successful completion the job will return 0
# If you need to set another return code, set the RETURNCODE
# variable in this section. To avoid conflict with system return
# codes, set a RETURNCODE higher than 100.
# eg: RETURNCODE=110
# Arguments to the USER SCRIPT are accessible in the usual fashion
# eg: $1 $2 $3
# The remaining sections are setup related and don't require
# modifications for most scripts. They are critical for access
# to your data
```

```

# --- BEGIN GLOBAL DIRECTIVE ---
#$ -S /bin/sh
#$ -o $HOME/$JOB_NAME.$JOB_ID.out
#$ -e $HOME/$JOB_NAME.$JOB_ID.out
#$ -m ea
# -- END GLOBAL DIRECTIVE --

# -- BEGIN PRE-USER --
#Name of experiment whose data you want to access
EXPERIMENT=${EXPERIMENT:? "Experiment not provided"} (Checks that the experiment has been
                                                    specified when the script was
                                                    called)

EXPERIMENT=`biacount $EXPERIMENT`
EXPERIMENT=${EXPERIMENT:? "Returned NULL Experiment"}

if [ $EXPERIMENT = "ERROR" ]
then
    exit 32
else
#Timestamp
echo "----JOB [$JOB_NAME.$JOB_ID] START [`date`] on HOST [$HOSTNAME]----"
# -- END PRE-USER --
# *****

# -- BEGIN USER DIRECTIVE --
# Send notifications to the following address
#$ -M user@somewhere.edu (enter your email address and you will get an email when the
analysis job is complete)

# -- END USER DIRECTIVE --

# -- BEGIN USER SCRIPT --
# User script goes here

#Need to input EXPERIMENT, SUBJ and RUN NAME
#Example qsub -v EXPERIMENT=Class.01 levelone.sh 1 1

SUBJ=$1 this indicates that the subject number is the 1st input after the script name
when the script is called
RUN=$2 this indicates that the run number is the 2nd input after the script name when the
script is called

#Set the directories
FUNCDIR=$EXPERIMENT/FSLtutorial/$SUBJ base directory for functional data
BEHAVDIR=$EXPERIMENT/FSLtutorial/$SUBJ/TIMING/run0$RUN base directory for task timing
data
ANAT=$EXPERIMENT/FSLtutorial/$SUBJ/ANAT1/reoriented_anat_brain.nii.gz path to T1 brain
extracted image
TEMPLATEDIR=$EXPERIMENT/FSLtutorial/$SUBJ/Scripts path to fsl first level analysis
template
OUTPUT=$EXPERIMENT/FSLtutorial/$SUBJ output base path name

#Set some variables
OUTPUTDIR=$OUTDIR/run0$RUN name of output directory (builds on output base path name)
DATA=$FUNCDIR/RUN${RUN}/run0${RUN}.nii.gz 4D data file

EV1=$BEHAVDIR/ev1.txt timing file for condition 1
EV2=$BEHAVDIR/ev2.txt timing file for condition 2

cd $TEMPLATEDIR change into the template directory

```

#Makes the fsf file using the template [this section uses the sed command to replace the variables in the template with the paths defined above:](#)

```
for i in 'first_level_template.fsf'; do
sed -e 's@OUTPUTDIR@'$OUTPUTDIR'@g' \
-e 's@ANAT@'$ANAT'@g' \
-e 's@EV1@'$EV1'@g' \
-e 's@EV2@'$EV2'@g' \
-e 's@FSLDIR@'$FSLDIR'@g' \
-e 's@DATA@'$DATA'@g' <$i> ${OUTPUTDIR}/FEAT_${RUN}.fsf
done

#Run feat analysis
feat ${OUTPUTDIR}/FEAT_${RUN}.fsf this runs the analysis!

# -- END USER SCRIPT -- #

# *****
# -- BEGIN POST-USER --
echo "----JOB [$JOB_NAME.$JOB_ID] STOP [`date`]----"
OUTDIR=${OUTPUTDIR:-$EXPERIMENT/Analysis}
mv $HOME/$JOB_NAME.$JOB_ID.out $OUTDIR/$JOB_NAME.$JOB_ID.out
RETURNCODE=${RETURNCODE:-0}
exit $RETURNCODE
fi
#-- END POST USER--
```

To run this script, type:

qsub -v EXPERIMENT=Class.01 levelone.sh 1 1

on the command line. **Replace the first 1 in this case with your computer's number. Here the first 1 is the computer number and the second 1 is the run number.**

The job will then be submitted to a node on the cluster.

If you then type **qstat** you will see that job running on the cluster.

A folder called run01+.feat will appear. FSL does not overwrite folders, it just appends a '+' when a folder with the same name is created.

An output file (for example levelone.sh.####.out) will appear in your subject folder. This will tell you if something went wrong with the analysis. You can open this file in Wordpad.

You can also submit the jobs in a loop like this from the command line:

```
for RUN in 1 2 3 4 5;  
do
```

qsub -v EXPERIMENT=Class.01 levelone.sh 1 \$RUN

```
done
```

This same type of script can be set up for a second or third level analysis in basically the same way.

Key take home from the script tutorial: variables in the template file are replaced by full paths by the scripts.

Most errors are errors in defining the paths, all input and output paths must be carefully defined in the script.

Thank you!
