aMIDUS3

Analyzing Task fMRI data with FSL software package

(Best to Learn while Doing…Especially Scripting)

FSL Pipeline –Written by David Lee and Mike Kelly

**Processing Scripts:** /study3/midus3/processed\_data/scripts

**QA Scripts:** /study3/midus3/processed\_data/QA/scripts

**QA Spreadsheet:** /study3/midus3/processed\_data/QA/MIDUS3\_MRI\_processing\_checksheet.xlsx

**General Steps**

0. Check with each lab’s standard pre-processing preferences

e.g.) J’s pipeline - Do \*not\* do time slicing correction while smoothing

1. Look at Data ASAP
2. Pre-Processing
   1. Run **Master\_Script\_Midus3.py**
      1. 1\_conver\_to\_nifti.py
      2. 2\_fix\_orientation.py
      3. 3\_create\_onset.py
      4. 4\_seperate\_onset.py
   2. 5\_create\_txt\_onset.py (if needed) (Takes Argument)
   3. 6\_brain\_extraction.py or 6.1\_brain\_extraction\_with\_fslAnat.py (if needed)
      1. bet sub-###\_T1w.nii.gz sub-###\_T1w\_brain.nii.gz –R -m)
3. Behavioral QA
   1. Check # of onset files
4. Functional QA
   1. 1\_trim\_and\_motion\_assess\_MIDUS3.py (Takes Argument)
   2. 2\_check\_volumes\_MIDUS3.py
   3. Check Orientation & Labels (Visual Inspection)
5. Structural QA
   1. Visual inspection
   2. Vitamin E?
   3. Check Orientation & Labels
6. Level 1 & QA– within Run
7. Level 2 & QA – combining Runs within subject (\*not\* concatenating)
8. Level 3 & QA – group level

**Specific Steps**

1. **Pre-Processing**
   1. Master script performs #1~#4, takes 30Minutes per subject
   2. Modify **5\_create\_txt\_onset.py (takes argument)** to fit your need for FSL input
   3. Run **6\_brain\_extraction.py**

**Brain Extraction (Skull Stripping structural)**

Purpose: removing skull & eyeballs to help with Image Registration

Image registration algorithms assume that **skull &** **non-brains** are removed (so if leave eyeballs (bright parts, it interferes w/ I.R.)

Background Info:

Striping of functional, you can be little sloppier, because:

1. Contrast is much lower (brighter thing are less bright)

2. df for the I.R. of bold to structural template limited to 3~6 df (not much room to move around) Whereas, Striping of structural, registered to MNI template using non-linear registration, a lot more freedom.

**Using bet to run skull stripping**

* bet <input> <output>[options]
* Tessellated sphere (Ball) slowly creeps out from the center until it hits boundary
  + So center must be estimated well (-R flag can do that)
  + Name output file same as input file but add extension “\_brain because:
    - Handy when using fNIRT:
      * When using fNIRT (the non-linear registration), it needs both images. It’s the only way for bet do distinguish between stripped vs. non-stripped
    - i.e.) “highres001\_brain.nii.gz” for skull stripped file (brain only)
  + Common Options (Flags): (type in “bet” in terminal to see )
    - -m : generates a standard binary mask for QA step
    - -R : robust estimation of center (often this is most useful)
    - -f <f>: smaller f (a #) will yield larger brain outlines
    - -g <g>: vertical gradient: positive g gives larger outline at top of

the brain

1. **Behavioral QA**
   * + - 1. **Check for # of cond.txt (onset) files (BOLD:Functional)**

How can I make cond.txt’s?

- Onset Files (Make my own!)

cond00[1-4]\_go.txt (onset time files)

(1. go, 2. such stop, 3. fail stop, 4. junk)

- Under “BOLD/Onset” directory

- Each run will go into the model as a **regressor** (How ever many runs you have for a participant)

**-** Each run has 4 cond.txt files (4 conditions)

- Each cond.txt has 3 onset Columns

(Onset Time in Sec., duration of trial (stimulus) in Sec, parametric modulation (if you don’t have it put ‘1’s))

- Read “readme” and also do the following:

-Checking all subjects have cond.txt’s:

***ls /home/slee/Desktop/ds008/sub\*/model/model\*/onsets/task\*\_run\*/cond\*.txt***

Doing the same thing but putting it in 1page (pipe it into “more”):

***ls /home/slee/Desktop/ds008/sub\*/model/model\*/onsets/task\*\_run\*/cond\*.txt | more***

Count the total# of cond.txt’s:

***find sub\*/model/model\*/onsets/task\*\_run\*/ -type f -name "\*.txt" | wc -l***

**I can write a script for this!**

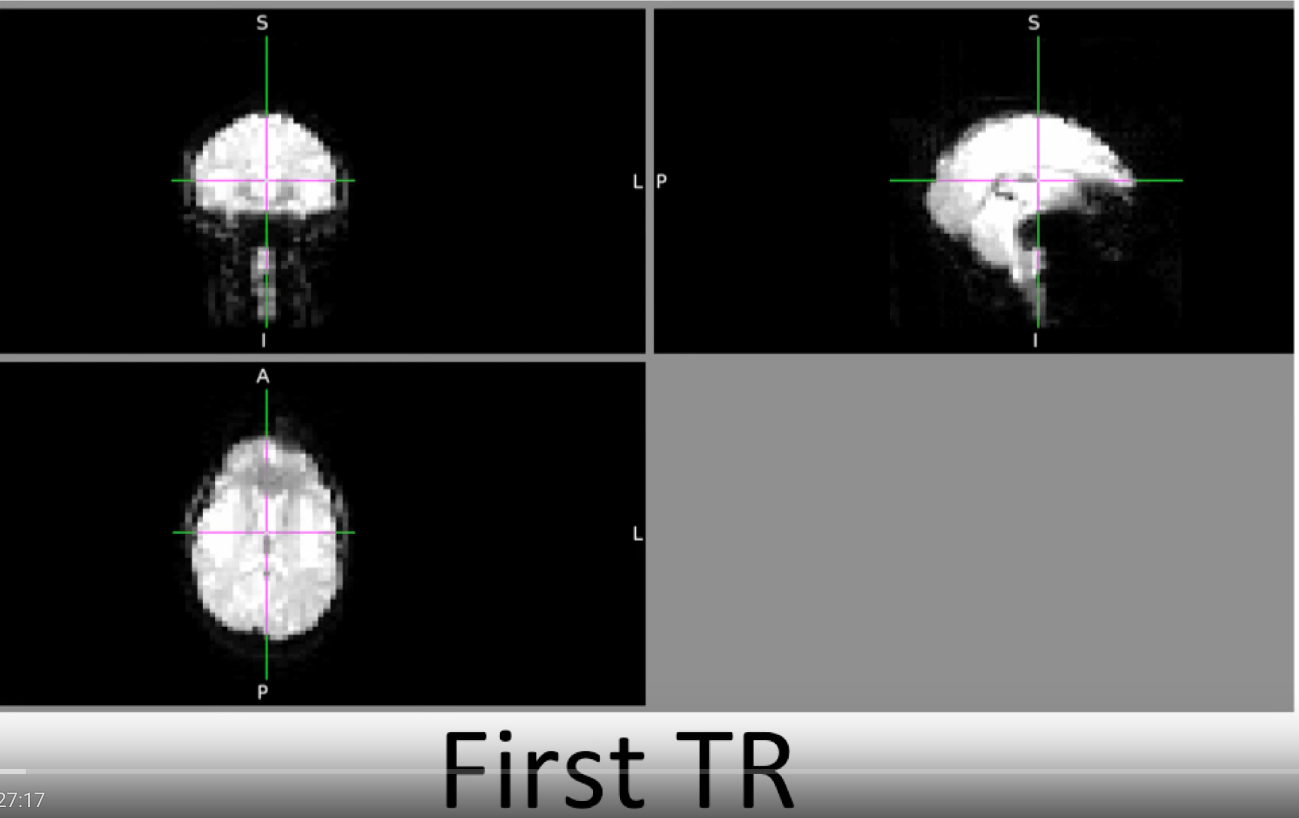
Need to watch J’s video on parametric modulation and onset files. Example: Gambling task: BOLD will fluctuate will fluctuate according to the amount won or loss.

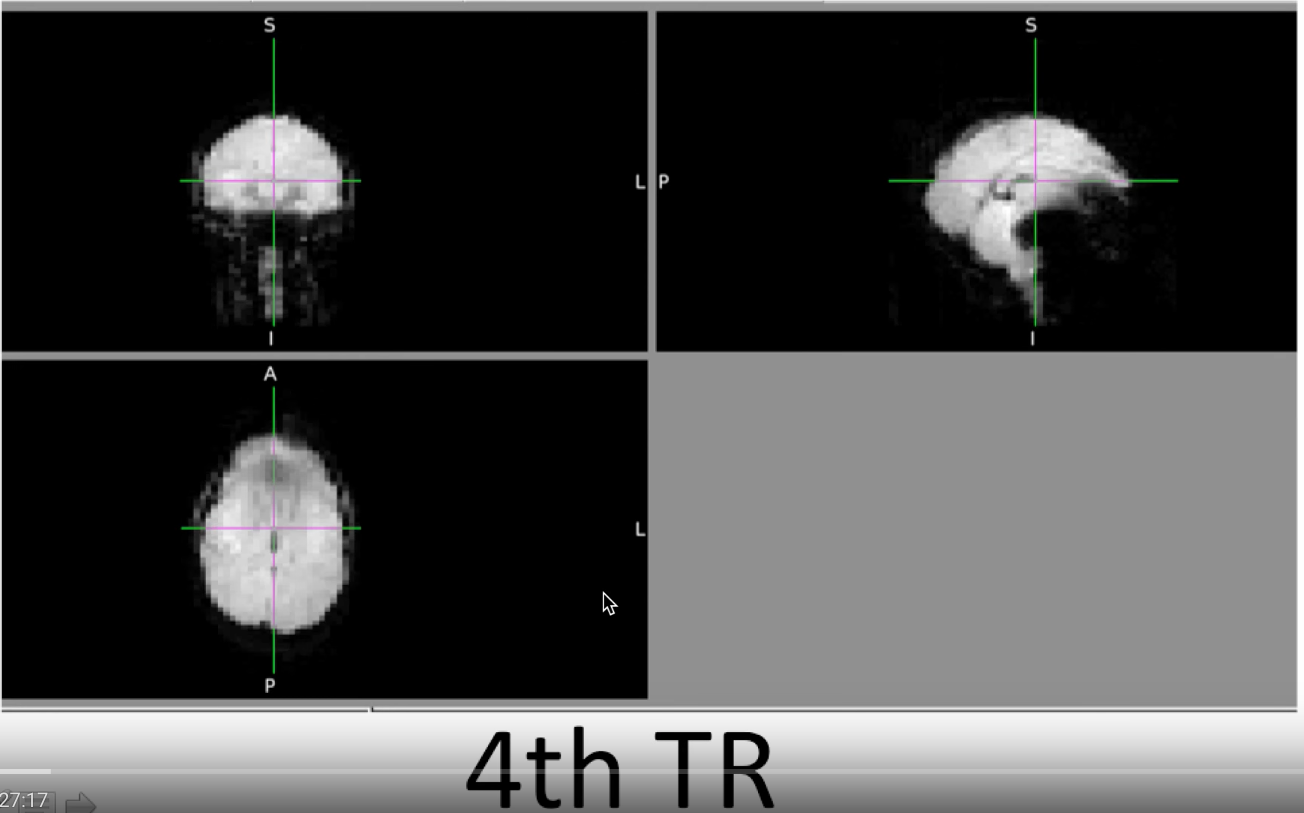
1. **Functional QA**
   1. Run **1\_trim\_and\_motion\_assess\_MIDUS3.py (Takes Argument)**
      1. **Trim volumes from the beginning (Scanner Warm up Artifact)**

- (Differ from lab to lab - some lab might dump this in the first place

- Jeanette’s code trims 4 volumes + no need to do it for DS008

Example: First TR is brighter than Fourth TR because scanner is warming up





* + 1. **Assess Motion**

Look at the html output that has all subjects

(Watch for really young or old (50+) population) **(BIGGEST!)**

There isn’t “best” method

But two\* Standards: just put motion parameters, but not enough!

1. Motion “Scrubbing”

- Delete bad time points via GLM (modeling out bad time points)

1. FSL or other ICA-based FIX algorithms

Semi-automated noise components selection

Take few subjects, run them through ICA, mark their bad components, FIX MACHINE LEARNING ALGORITHM to go through the rest of the subjects and mark the bad components! Take the time series from there and put it in the model

Either methods are good, but we do #1.

* FSL commands
  + **fslswapdim** - you could use fslreorient2std
  + **fslroi** - trim any dimensions (chop off extra volumes)
  + **fsl\_motion\_outliers** - scrubbing - createfile , we can dump into our first-level model
  1. **Check # of TimePoints(V) and Headers (BOLD:Functional)**
     1. Run **2\_check\_volumes\_MIDUS3.py** // (chmod +X) to make files executable
     2. Check the output file “check\_volume.txt”
     3. FSL Commands

**-fslnvols** bold.nii.gz : Check for correct **time points**! 176 or 182 Volumes!

-**fslinfo** bold.nii.gz: check headers of NIfTI

-**fslhd** bold.nii.gz: more detailed information (full headers)

* 1. Check Orientation and Labels
     1. Check Orientation (BOLD:Functional)

**- fslview bold.nii.gz**

* + 1. Robust FSL commands

**- fslorient2std:** changes orientation (Double - triple check the L-R afterwards regardless)

**- fslswapdim :** swap the dimensions (when fslorient2std does not do the job)

* + 1. Check Labels - Superior, Inferior, Anterior, Posterior (BOLD:Functional)

**fslview bold.nii.gz**

Left and Right Swaps are the worst!

Load and view one by one

Should already been fixed

By now, I should have excellent quality NIfTIs!

1. **Structural QA**
   1. **Visual Inspection**

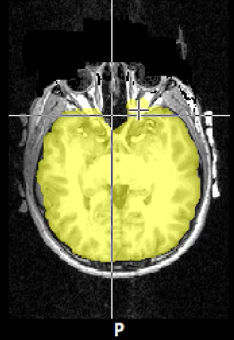
- Open one structural by one and QA

- Load the original Brain

- Load different bets on top of it, then manually check if any masks either

over/under cover the actual brain (by checking eyeballs etc)

Example of eye being covered (too much of non-brain :



-Choose one bet option that best covers the brain

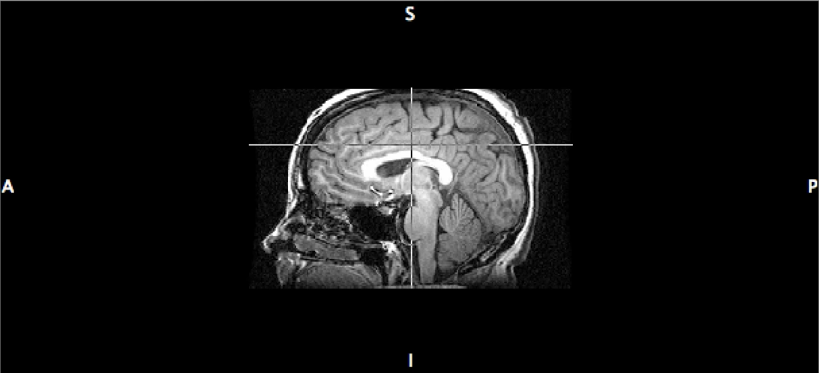
* 1. **Vitamin E**

Always on the Right Side

* 1. **Check Labels and Orientation**

For flipped Orientation: **use 1\_fix\_flip\_bet\_anayomical.py**

Example of sagittal having correct labels, but wrong orientation (Left-Right Swap):



\* In case this doesn’t go well, and in case there are ton of eyeballs:

- Use FREE SURFER (information is in “1\_fix\_flip\_run\_bet\_anatomical.py”)

\* Worst-case scenario: labor-intensive manual hand edit the mask…

By now, I should have excellent quality Structurals!

**Side Lecture: Frame-wise Displacement (FD)**

-2012 Neuroimage Paper - Resting State Data

- look at motion, exclude ones with excess motion, include motion parameters (n = 6)

- learned that its not enough (especially in children)

- **scrubbing** or Censoring

- nodes of interest, cluster them together into networks

- brief result summary: after scrubbing, learned motion was causing an artifact

- In TASK DATA, it’s LESS of an issue

- In Resting, it’s a big deal because the Model: (pick a part of the brain, take the time series, average over that part of that brain and use it as a regressor.) Not a good regressor if it has artifacts.

- In Task, regressors (convolved = HRF) are beautiful guess of what BOLD looks like (NO Noise!)

- Motion Regressors

- called McFlirt in FSL (Lev1 Analysis)

- 6 Time Series - aligning the data in BOLD time series over time to the middle time point

- 6 df image registration —> you get time series saying **how much each volume had to shift: translations & rotations in x,y,z**

**-** Raw motion Regressor

- add regressors to a model : it will do the job **only if the model works**

Assumptions:

**-** motion artifac**t** is not impacted by hemodynamic delay (instantaneous)

- motion artifact is linear

* 1. QA that
* Level 1 - within a Run within a Subject
* QA that
* Level 2 - combining Runs within a Subject (I guess if there is only one run, then skip leve2 ?)
* QA that
* Level 3 - group Level
* QA that

**5. Level 1: Within Run**

**Part A: Setting up the Feat GUI**

Ingredients:

1. BOLD Data
2. Onset Timing Files (for MS3 I have them as excel, do I need .txt?)
3. Field Map (Only if I am using them)
4. Slice Timing Info (Only if I care to do them)
5. Skull-Stripped T1w

Open up Feat gui in Terminal

Linux – Feat &

MAC – Feat\_gui &

Tab Settings

**First-level analysis & Full analysis**

**Misc**

Turn off progress watcher – to prevent html ambush

**Data**

**Click “Select 4D data”**

Select “BOLD” directory

Select “task001\_run001”

Select “bold.nii.gz”

Check TR, if  **\***not correct, use this to correct TR:

1\_Fix\_tr\_in\_nifti\_header.bold.py

The above script uses:

**fslinfo** bold.nii.gz

Things we want to change:

Pixdim1 dimension of the voxel (X)

Pixdim2 dimension of the voxel (Y)

Pixdim3 dimension of the voxel (Z)

Pixdim4 TR

The script uses **fslchpixdiw** to change the 4TH Pixdim (TR)

**Click “Output Directory”**

Select “MODEL” directory (for MS3, this I need to create)

\* If I don’t type anything else at the end – it’s going to create directory called “.feat” (invisible – so problem)

\* But we want something like run2, so type “run2” at the end. No need to do “run2.feat”

**Set High Pass Filter Cutoff(s) to 100**

Remove any variable frequencies such as scanner drifts that occur with cycles of 100S or greater. May remove information if single trial is 100S or longer.

**Pre-Stats**

**Motion correction: MCFLIRT**

Correct motion within bold data over time (Match all volumes to the middle volume within Run)

B0(Field Map) Unwarping: (only if doing field-map correction)

If interested in areas near locations that are susceptible to signal drop out or artifacts such as cortical patches near sinuses or OFC

Slice Time correction: (only if you are doing it)

Slices are NOT acquired instantaneously; Shift slices so it is as if all slices (entire volume) are acquired at a single time point

**Bet brain extraction (for functional)**

Important for future Co-Registration: functional runs to anatomical

**Set Spatial Smoothing FWHM(mm) to “5”**

Averaging data from nearby voxels, increasing signal to noise ratio b/c any present signal will be averaged and summated while noise, presumably should be random will be canceled out. Trade off: very large kernels increase signal to noise ratio. Smooth over large areas of cortex and subcortical regions, and eliminate any spatial resolution at mm below kernel size. i.e) if smoothing kernel is 10mm, will NOT be able to tease apart contributions from AMG as opposed to nearby areas.

Intensity normalization

Attempt to match intensities to certain preset constant. Intensity will be scaled, so that mean intensity across volume will match present constant. This can introduce biases b/c it can artificially deflate the amount of variance observed in data.

**Temporal Filtering (Highpass)**

Previously specified in “DATA” Tab

MELODIC ICA data exploration

Decompose data into several independent components. Isolate certain components as being associated with condition or effects of interest and/or used to remove artifacts or noise.

**Registration**

1. Co-registration: aligning all functional images into same orientation/space as anatomical image

2. Normalization: warping anatomical image to standardized space

**Click Main Structural Image**

Select “Anatomy Directory”

Select “highress001\_brain.nii.gz”

Main \*skull stripped\* structural Image

Select Linear “Normal Search” and “BBR”(Boundary Based Registration)

Type of registration to align bold data to subjects’ structural

Default: Linear Affine Registration (12 DOF)

May choose “Non Linear”, then use brain with skull

Turn on “Nonlinear” (FNIRT)

Important to put in \*brain with skull\* instead of \*skull stripped\*

**Stats : Set up Model**

**Skip** confound.txt if you do \*not need to scrub

Turn on “Standard Motion Parameters”

Standard: 6 motion parameters

Motion Parameters come from…Pre-stats tab 🡪 MCFLIRT = aligning the bold data time point by time point, it registers it to the middle image, and so for each time point, you get translation, rotation, and X-Y-Z direction.

Standard + Extended: 6 + squares (derivatives) : 24 full motion parameter

**Full Model Set up**

Turn on “Add additional confound EVs”

EV = Explanatory Variable

Putting in a txt in one column format that has scrubbing information

Select BOLD 🡪 Motion Assess 🡪 confound.txt

Either empty or column for each bad TR (all 0’s, and scrubbed TR is 1’s)

Click “Full Model Setup”

Cond 001 Go

Cond 002 Successful stop

Cond 003 Failed stop

Cond 004 junk

EVs

Number of Original EVs 🡪 4

EV Name 🡪 Go, succ stop,

Basic Shape 🡪 Custom (3 column format)

Filename 🡪 Model/onsets/task001\_run001/cond001.txt

Convolution 🡪 Double-Gamma HRF

Gamma overestimates activation

Turn on “Temporal derivative” – can soak up variability if your onset timing is off or the peak assumption of the time to peak in the double gamma HRF is off little bit.

Whatever you do to the data, do it to model as well

Turn on “Temporal filtering” – Hypass Filter used to remove Low frequency noise,

Pops up in 3 locations (want to be consistent in Data & Design)

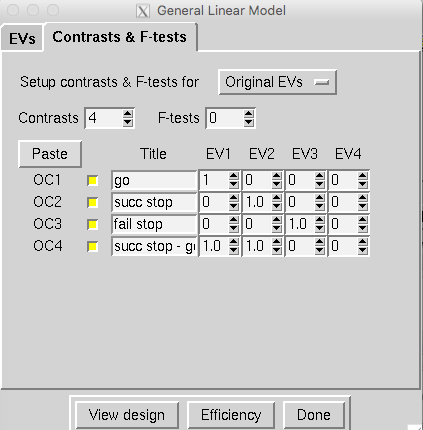
Data tab – “High pass filter cutoff (s)” 🡪 100 seconds

Pre-stats tab – Turn on “Highpass”

(And here) Stats tab – Turn on “Apply temporal filtering”

Contrast & F-tests

Contrasts 🡪 4



**Post – stats**

Select “None” for “Thresholding”

Turn off “Create time series plots” – to free up space

Click “Save” path/path/filename

Click “Go”

**Part B: Scripting Level 1 Analysis**

3\_make\_fsf\_lev1.py

**Part C: QA**

Useful Commands:

**Checking Errors:**

cd “analysis\_directory”

grep ‘Error’ sub-MRID-\*/model/run\*.feat/report\_log.html &&  
grep ‘ERROR’ sub-MRID-\*/model/run\*.feat/report\_log.html

**Check Z stats:**

cd “analysis\_directory”

ls sub-MRID-\*/model/run\*.feat/stats/zstat1.nii.gz

ls sub-MRID-\*/model/run\*.feat/stats/zstat1.nii.gz | wc -l

4\_QA\_all\_lev1s.py

Single html with all png files

Open up html.

Check if repressors stop wiggling.

**1. Checked Zstats**

- Use this command: ls sub\*/model/run\*\_fir.feat/stats/zstat1.nii.gz

check\_lev1\_output.py

# of zstats == # of contrasts

**2. Regressors**

- There seems to be 7 regressors for each pos, neut, neg - so total of 21.

Yes!  So, that's where I got the "21" I mentioned above.

**3. Pair-Wise Collinearity**

-Focused on the top left corner -->> to check the diagonal color contrast

Ignore the motion regressors, basically.  Actually, for this type of model, you do not need to worry about collinearity, so you're off the hook!

**4. Graphs**

 -You seem to emphasize the green line: large relative displacement

Question: Am I looking for weird peaks? Some guidance here would be helpful.

I don't recall what graphs you're referring to.  Motion?  If it is motion, you've already assessed this via the FD plots you guys looked at a while ago, so you can ignore there.

**5. Registration summaries (end result)**

Question: out of two lines, the top line has the MNI template as the outline & Brain as the bold data where as the bottom is vice versa. Am I looking for mismatch here? Again, some guidance what to look at would be helpful.

Yes, extreme mismatches.  You'll likely notice extreme things.

**6. Level 2**

Multiple Subjects – Multiple Runs per Subject

Purpose: Average the runs within subject

One sample T test with single contrast of 1

**Step A: GUI Setup (Template Set Up)**

Average the runs within subject

**Data**

Select “Higher-Level Analysis”

Select “Inputs are lower-level FEAT directories”

Set output directory

Input Feats

**Stats**

Select “Fixed Effects” – do NOT estimate any variance, so then GUI allows user to decrease FEAT to 2

Click “Full Model Setup”

EVs

Single column of 1s – averaging things together

Contrast & F-tests

“Group mean”

**Post-stats**

Turn off Thresholding to “None”

Uncheck “Create time series plots”

This creates design.fsf file!

When you have only one run: remember to put in same run twice, and put 0 into EV

**Step B: Python Setup**

5\_make\_lev2.py

Find all the. feat’s and replace…

**Step C: QA**

Check they ran without any flaws

Usual errors:

1. Big chunk of brain is missing
2. Make sure they are all yellow (RED is bad!)
3. **Level 3**

**Step A: GUI Set Up**

Average all subjects

**Misc**

Turn off “Balloon Help”

Turn off “Progress Watcher”

**Data**

Select “Higher-Level Analysis”

Select “Inputs are 3D cope images from FEAT Directories”

Input “Number of inputs” (How many subjects?)

Select “Select cope Images” 🡪 put paths in

ls copes in terminal 🡪 in .txt 🡪 copy/paste from there to GUI

Set output directory

**Stats**

Select “Mixed Effects FLAME1” – type 1 error rate is under 0.05

\*Could turn on “use automatic outlier de-weighing”, but time-consuming

Click “Full Model Setup”

Group Analysis – 1 Sample t-test

EVs

Single column of 1s – averaging things together

Contrast & F-tests

Title: “mean>0”; EV1: “mean<0”

Title: “mean<0

**Post-stats**

Turn on Thresholding to “Cluster”

Z threshold: 3.1

P threshold: 0.05

Uncheck “Create time series plots”

sub-###/model/lev2.gfeat/cope2.feat/stats/cope1.nii.gz

cope 2 – corresponds to 2nd contrast in the level 1 analysis

cope 1 – coreesponds to 1st contrast in the level 2 analysis

Average greater than 0 of the second cope…..

**Step B: QA**

Check the mask

fslview mask.nii.gz

load bg.image.nii.gz

Check Activation

Enter cope1.feat directory

fslview thresh\_zstat1.nii.gz -> corrected for multiple comparisons

if nothing met threshold:

Enter stats directory

fslview thresh\_zstat1 🡪 uncorrected

Check log and html output

PostStats – html

G

Flip through filtered\_func\_data

Group Analysis Steps

1. Make a list (.txt) of order of cope1’s to put in “Select Cope Images” in MISC tab

i.e) **suborder.lev.3.txt**

2. Set up Gui

A. Create a file (.txt) to put in “EVs” in STATS tab

**make.group\_mod\_covariates.R**

matching order of subjects with covariates for model

**install libraries packages**

move this script to FSL\_Pipeline directory!

B. Save as cope1\_XXXXXXXXXX.fsf (output directory)

3. Create a file (.condor)

i.e.) 3\_run\_lev\_3. condor

A. **run 3\_make\_fsf.py**

Change fsf template saved from step 2

B. Change email address in .stub file

4. Run .condor on condor server

5. Result

21 Group Feats

TR 1 ~ 7 for each TR -> Three 4D images (Neg Pos Neut)

Fslmerge –t 🡪 thresh Z stats

ls –d lists directories \*excluding subdirectories

From here – written by Mike Kelly

**General Steps**

0. Check with each lab’s standard pre-processing preferences

e.g.) J’s pipeline - Do \*not\* do time slicing correction while smoothing

0. Look at Data ASAP

Set up Automated pipeline to immediately following data collection

-Check that all files that should be there are actually there (scans & runs, onset files,etc.) !

-Open relevant files (e.g., onset files) and make sure they look like they should

-Make sure files and file structure (we will use BIDS formatting for MIDUS3) makes sense; read read-me files if necessary

Verify that functionals have the right number of time points w/ *checkvolumes.py* or fslnvols

1. Convert DICOM to NIfTI & Create Onset files
   1. Re: Onset files, do we need to have a ‘parametric modulation’ column? I doubt it, and if we did, I’m guessing it would be all 1s (Jeanette’s Step 3: Prepare your BOLD data video)
   2. Run Master\_script\_working\_MIDUS3\_Imaging.py
2. QA of NIfTIs
   1. Open *every* scan with fslview and do a quick inspection for issues, including:
      1. Making sure orientation and labels are ok. Take care to look for R/L swaps. **Look for vitamin E capsule in anatomical scans! Should be on R side.**
3. Skull Strip Structurals & QA
   1. Run bet: bet <input> <output> [options]
      1. **Example: *bet sub-001\_T1w.nii.gz sub-001\_T1w\_brain.nii.gz –R -m***
      2. Output = input\_brain
      3. –m : generates binary brain mask; -R : Robust center estimation
      4. Note: if we would like, we can adapt Jeanette’s script rather than run this via command line (1\_fix\_flip\_run\_bet\_anatomical.py)
         1. I’ve moved a copy of her script into QA/scripts, but it still needs some editing. If we are QAing 1 at a time, might be just as simple to run via command line. On the other hand, a script could eliminate potential mistakes by ensuring consistency. **Let’s chat when we’re both back**
   2. Load original structural and mask
      1. Make mask transparent enough to see brain beneath
      2. Check that mask covers entire brain and isn’t including skull/other parts we don’t want (carefully check area around eyes, as this is where problems most frequently occur).
4. Basic pre-processing to Functionals & QA
   1. Trim the junk volumes (a.k.a. warm-up scans, trim off first 4?-volumes)
      1. 2\_prep\_bold.py🡪I believe this needs subnum (e.g., 006) as an input
         1. Creates html file w/ QA information. Creates text file w/ bad subjects. *How does this compare to Sasha’s html files? If they’re comparable, may not be necessary to run Jeanette’s script. One difference is that Jeanette’s code creates 1 html file w/ all subjects, while Sasha’s makes 1 file per subject, but if we’re looking at these 1 at a time, it won’t help much to have an html file w/ everyone if data collection is still ongoing.*
         2. Trims unwanted volumes *(*uses **fslroi** command*; will need to change fd threshold to 0.5; may need to change script to make sure it only trims 4) – checked and it trims 4 TRs…,so 231 becomes 227*
         3. Assesses for motion and creates output file w/ motion info
            1. Throws out subjects when bad TRs reach a certain percentage of total TRs; make sure we decide on correct threshold to toss entire scans (I believe we had been using 25%, but we should confirm this)
   2. Check orientation
   3. Assess motion
5. Level 1 & QA– within Run
   1. Setting up the Feat GUI
      1. Need: BOLD Data, Onset timing files (make sure ours work!), fieldmaps (if we’re using them; are we? I thought not, but could be wrong), slice timing info (may not need?), skull stripped structural
         1. Steps:
            1. Type ‘Feat\_gui’ in the terminal
            2. In ‘Data’ tab:

Make sure ‘First-level analysis’ and ‘Full analysis’ are selected.

* + - * 1. ‘Misc’ tab:

Balloon help (optional) and Progress Watcher (optional—don’t use if submitting to condor)

* + - * 1. Back to ‘Data’ tab:

Click ‘Select 4D data’ and select one of the runs of functional data for a subject

If problems with ‘TR’:

Use fix\_tr\_in\_nifti\_header\_bold.py

Double-check TR time and make sure it is right before running script (TR hard-coded into script)! BE VERY CAREFUL WHEN CHANGING HEADER INFO!

Check that TR/# of volumes looks right

Change output directory by adding run number:

Ex: /…/model**/run2**

High pass filter cutoff: Change to ?100?

* + - * 1. Pre-stats

Motion correction: MCFLIRT

B0 unwarping: only if registrations need to be fixed w/ Field maps

Will likely leave slice timing correct at ‘None’

Make sure BET brain extraction is checked

Spatial Smoothing: ~2x voxel size. FSL default of 5 should be ok.

* + - * 1. Registration tab

Main structural:

Skull-stripped structural: …\_brain.nii.gz

Linear: Normal Search/BBR

Standard space

MNI template is input

Choose nonlinear for Standard space/registration

* + - * 1. Stats tab

Check ‘Use FILM prewhitening’

Choose ‘Standard + Extended Motion Parameters’

Check ‘Add additional confound EVs’

Select output from motion\_assess script: *confound.txt*

Click ‘Full model setup’:

Change # of EVs to # of conditions (for us is this 3??? EV name run1, run2, run3???)

Under ‘Basic Shape’, need to input onset file. In Jeanette’s video, she inputs ‘Custom (3 column format)’, but I think this will be different for us.

Input ‘cond001.txt’

Convolution: Double-Gamma HRF

Keep ‘temporal derivative’ checked, along with ‘apply temporal filtering’.

**Contrasts & F-tests**

Set up comparisons between conditions

Inspect design matrix

NOTE: I think our setup is very different from the setup used in Jeanette’s video (Onset files created differently; do we need to turn these into 3 column files; maybe separate files for IAPS & black and white faces?

* + - * 1. Post-stats tab

Change ‘Thresholding’ to ‘None’

Always turn off time series plots!

* + - * 1. Click ‘Go’!
  1. Scripting Level 1 analyses
     1. Create template: design.fsf
     2. Fill in wildcards to perform search/replace
        1. Note all things that will change in the data: Sub num Run num, # of time points, etc.
        2. Make sure your wildcards don’t match other text in fsf template!
     3. Write script to do search/replace
        1. Jeanette has ‘make\_fsf\_lev1.py; NEED TO ADAPT THIS TO OUR DATA!`
     4. Run with command “feat design.fsf”