

Integrating FreeSurfer and FSL FEAT

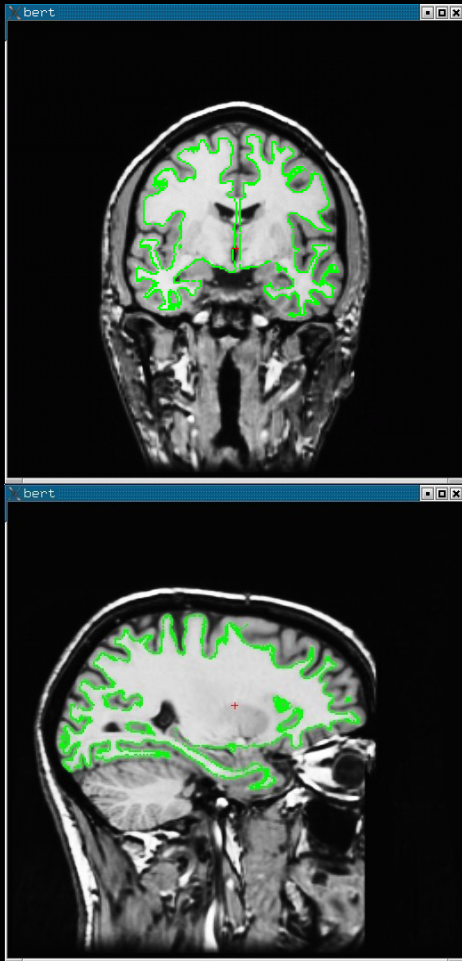


Outline

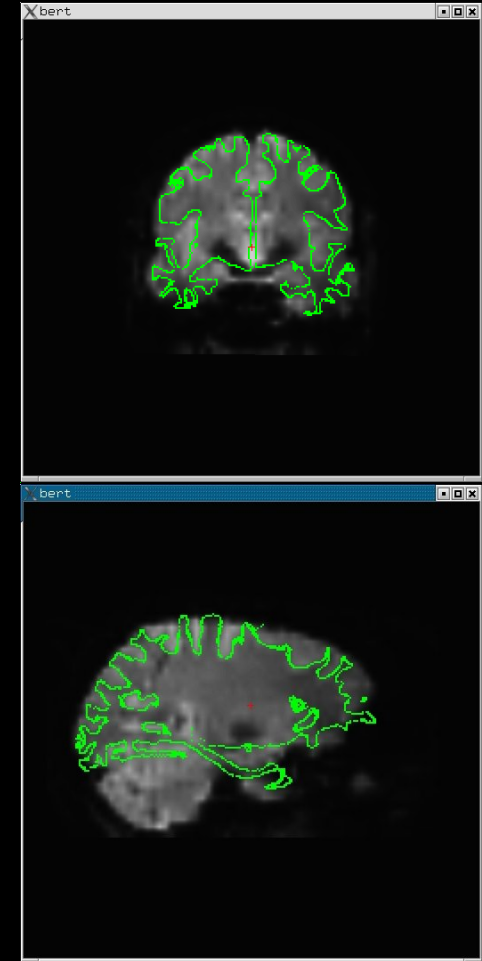
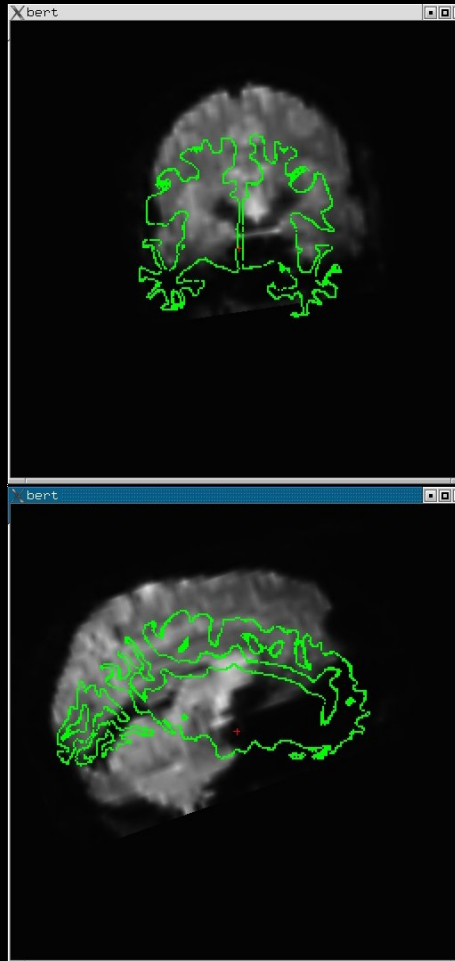
- Registering FEAT → FreeSurfer Anatomical
 - Automatic (FLIRT)
 - Manual (tkregister2)
- Viewing FEAT output on Anatomical
- Sampling FEAT output on the surface
 - Individual
 - Common Surface Space (Atlas/fsaverage)
- Mapping FreeSurfer Segmentations to FEAT
- Group Analysis

Registration

FreeSurfer Anatomical (orig)



Template Functional



Note: Registering the template functional volume to the anatomical volume is sufficient to register the template to the surface.

FreeSurfer Registration

FreeSurfer Subject-Specific

- Volumes
- Surfaces
- Thickness
- ROIs



Your Data/Software

- fMRI (FSL, etc)
- EEG/MEG
- DTI
- ...

Registration Matrix

- Affine 4x4
- As many as 12 DOF (usually 6)
- Text file

Automatic Registration

First: analyze your data with FEAT (No Smoothing)

```
reg-feat2anat --featdir fbert1.feats --subject bert
```

Uses FLIRT to perform 6DOF registration:

fbert1.feats/example_func.nii.gz

\$SUBJECTS_DIR/bert/mri/brainmask.mgz

Creates FreeSurfer registration matrix:

fbert1.feats/reg/freesurfer/anat2exf.register.dat

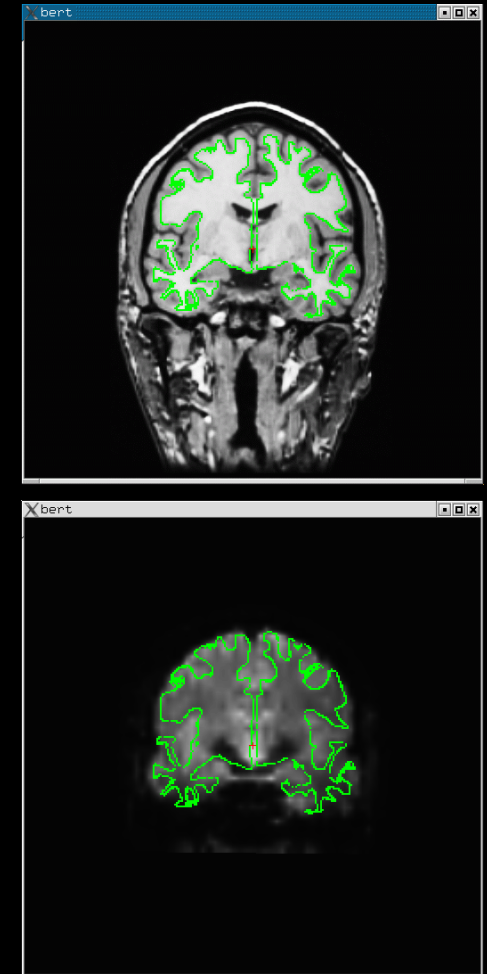
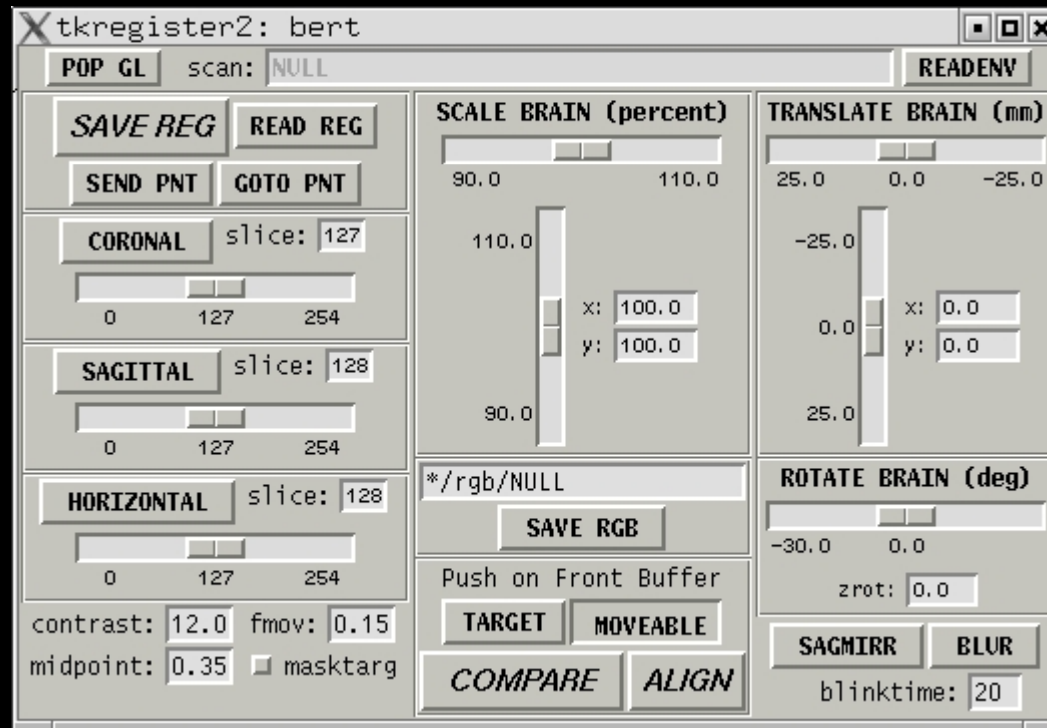
Also creates: fbert1.feats/reg/freesurfer/anat2std.register.dat

This matrix maps standard space to anatomical space and can be used when combining data within subject across data sets

reg-feat2anat --help

Manual Registration

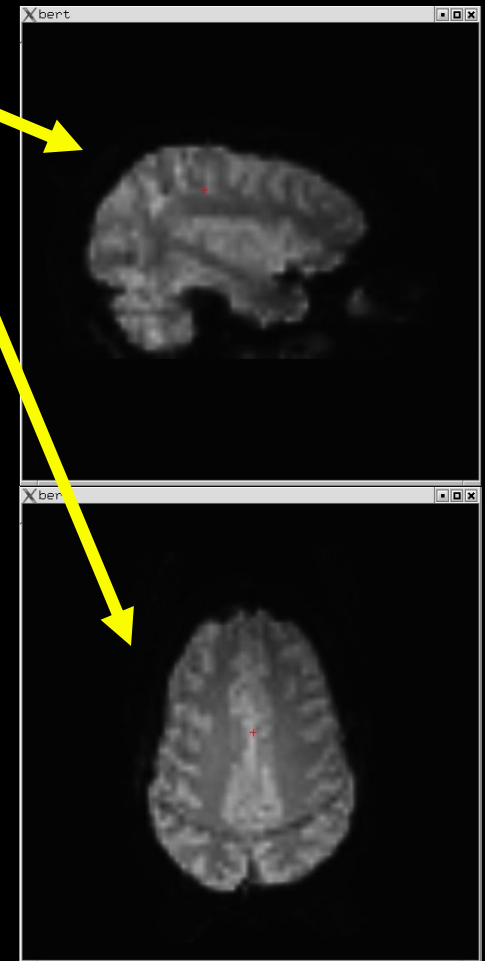
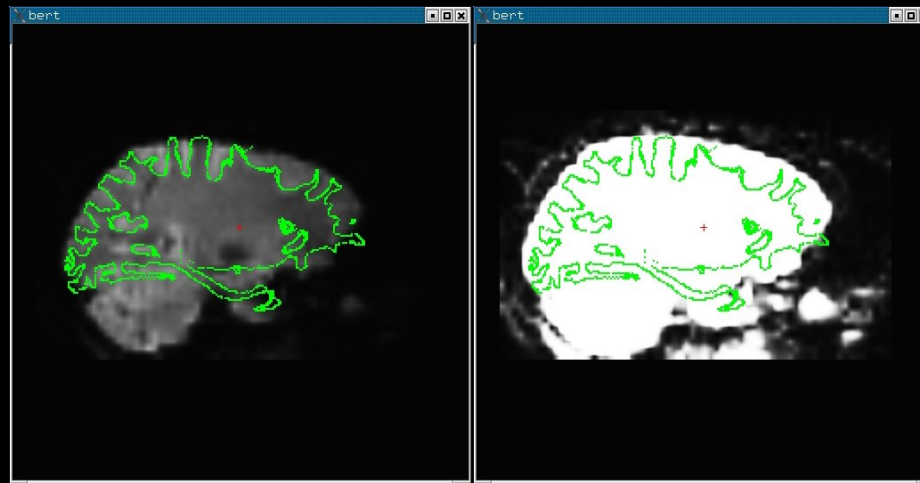
- `reg-feat2anat -feat fbert.feats --manual`
- Visually inspect registration
- Manually edit registration (6 DOF)
- Cf Manual Talairach registration



`tkregister2 --help`

Tips

- Rigid = 6 DOF = No stretching
- Use CSF to get a sense of where the folds are
- Avoid using B0 distortion regions
- Avoid using ventricles
- Warning about “edge” of the brain
- Same Subject, Left-Right Flips



Viewing Functional Activation on the Volume

```
tkmedit bert orig.mgz -aux brain.mgz
```

```
-overlay ./fbert1.feats/stats/zstat1.nii.gz
```

```
-overlay-reg ./fbert1.feats/reg/freesurfer/anat2exf.register.dat
```

```
-fthresh 2.3 -fmax 4.3 -seg aparc+aseg.mgz
```

Visual, Auditory, Motor

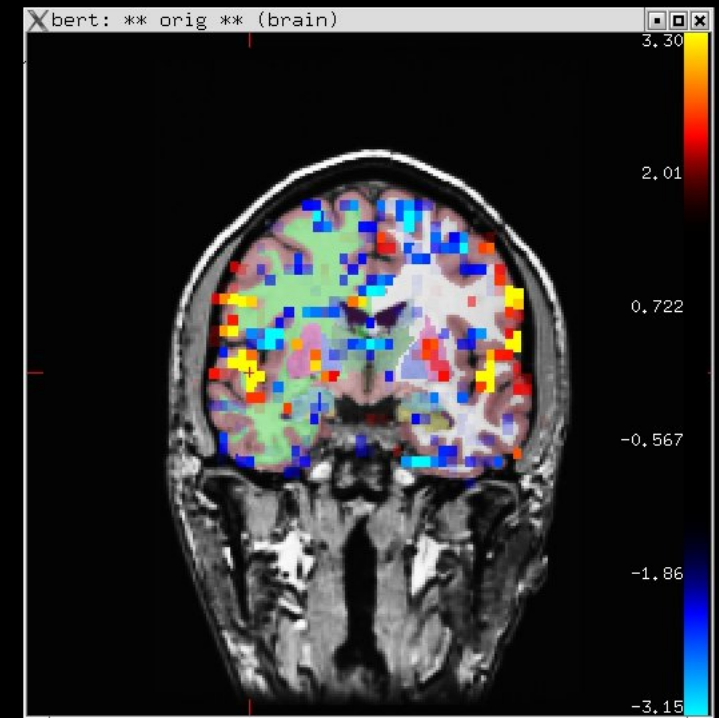
Thresholds depend on the nature of the data,

Eg, for zstat image, 2.3 means $z > 2.3$

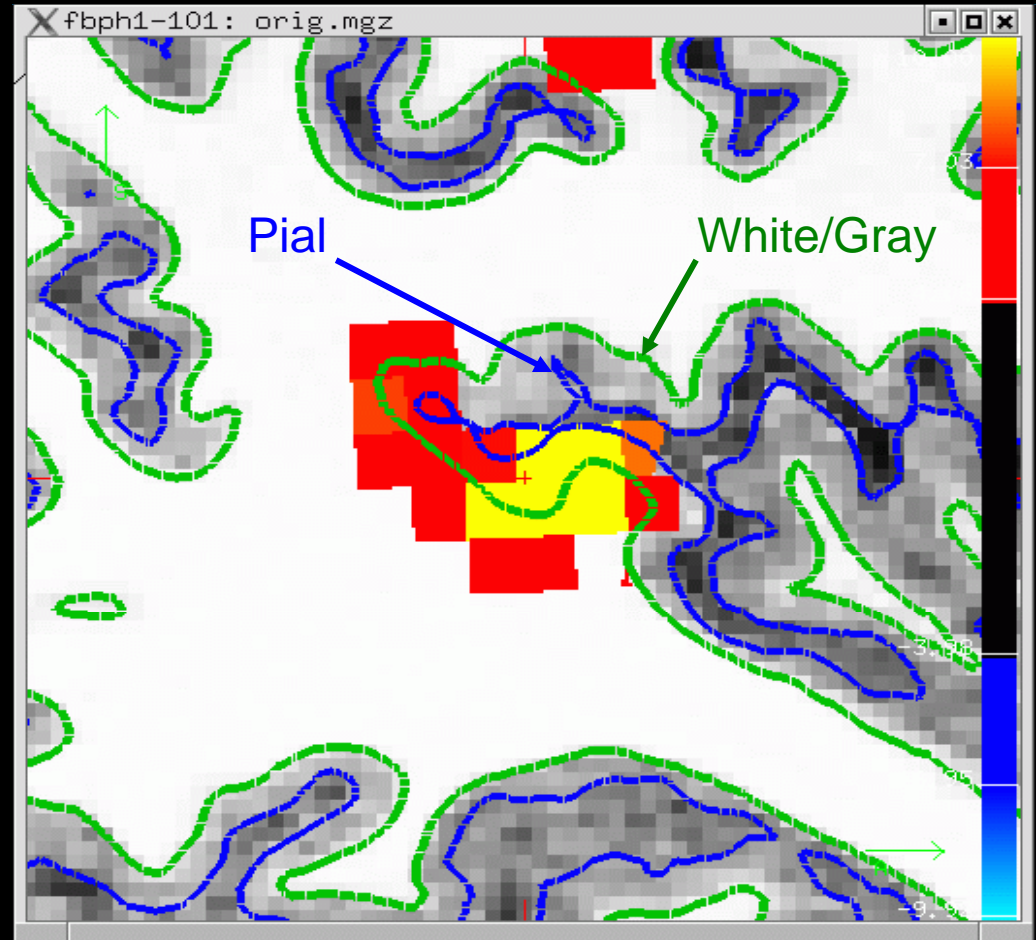
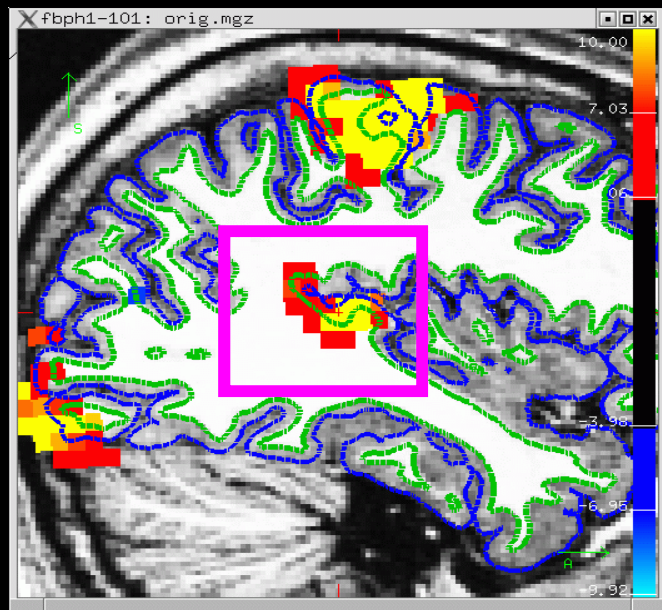
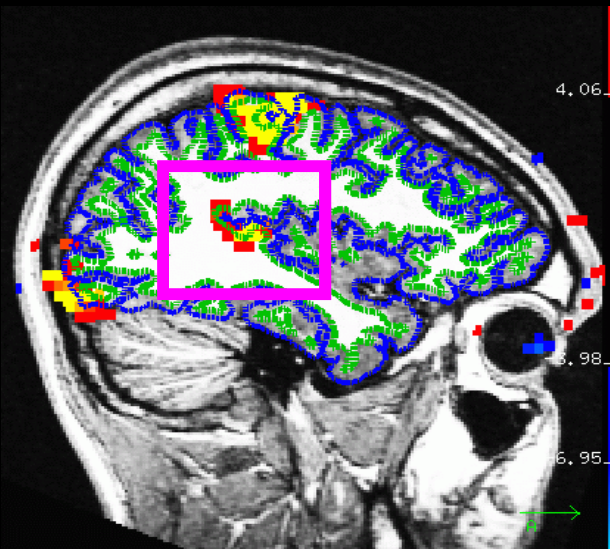
Can be changed with

View->Configure->FunctionalOverlay

Can display any functional data, eg,
zstat, fzstat, cope, pe, etc



Sampling on the Surface

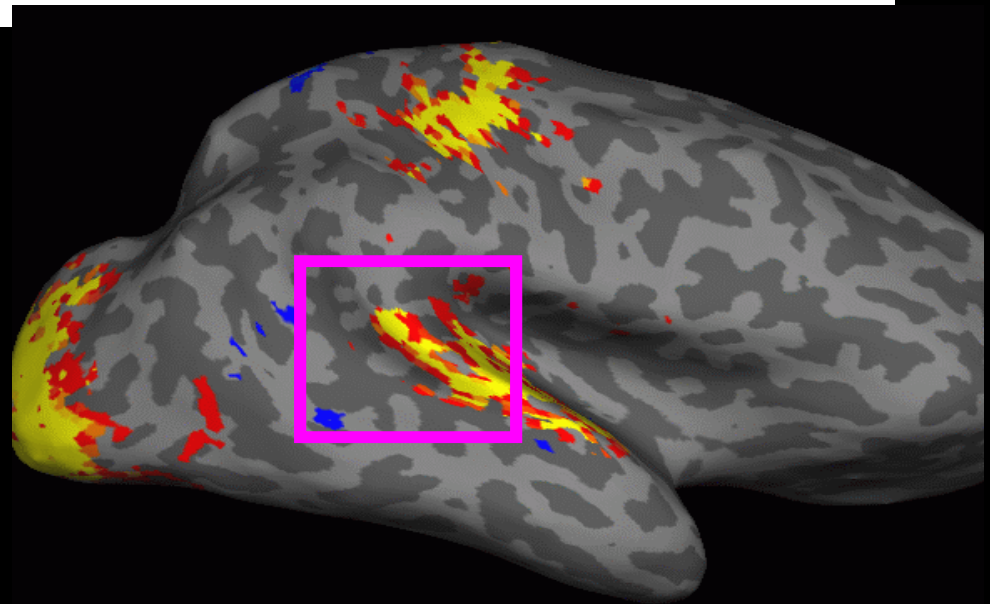
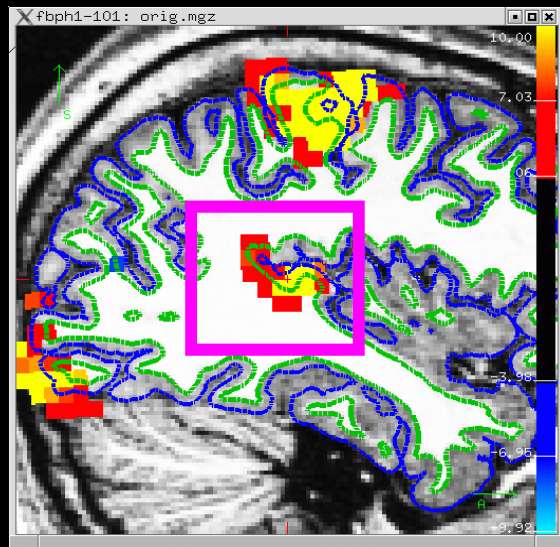
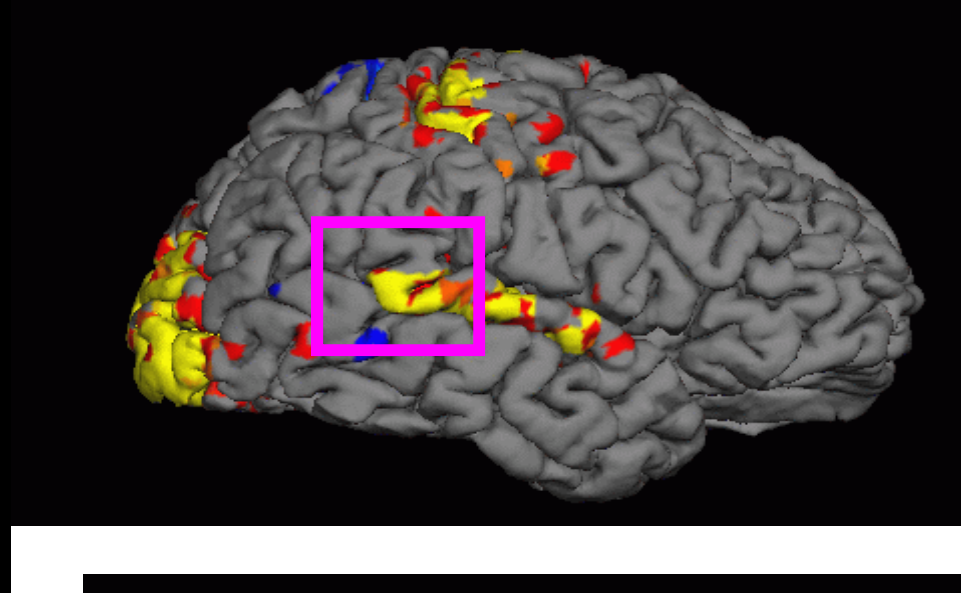
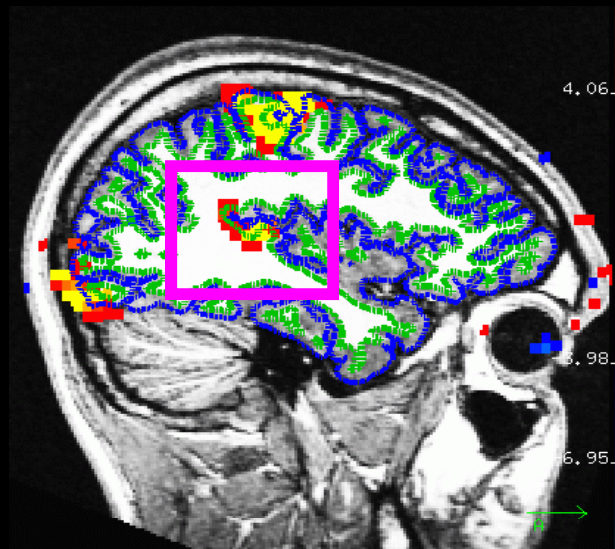


- White/Gray
- Pial
- Half Way
- Average

Projection Fraction
--projfrac 0.5

Applying FreeSurfer Tools to FSL fMRI Analysis

Sampling on the Surface



Viewing FEAT Stats on the Surface

tksurfer bert rh inflated

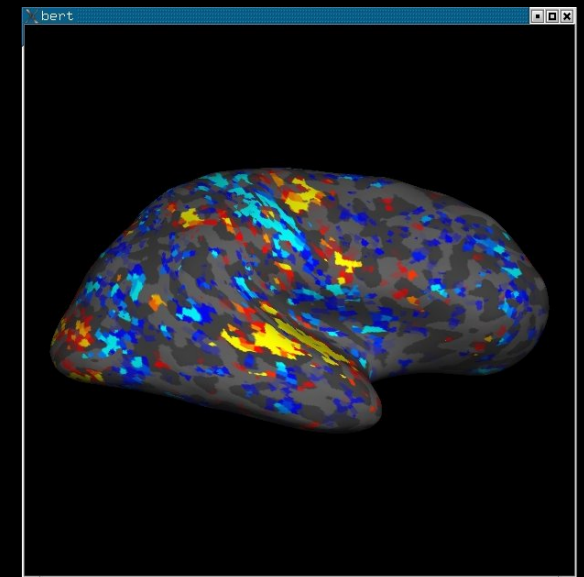
-overlay ./fbert1.feats/stats/zstat1.nii.gz

-overlay-reg ./fbert1.feats/reg/freesurfer/anat2exf.register.dat

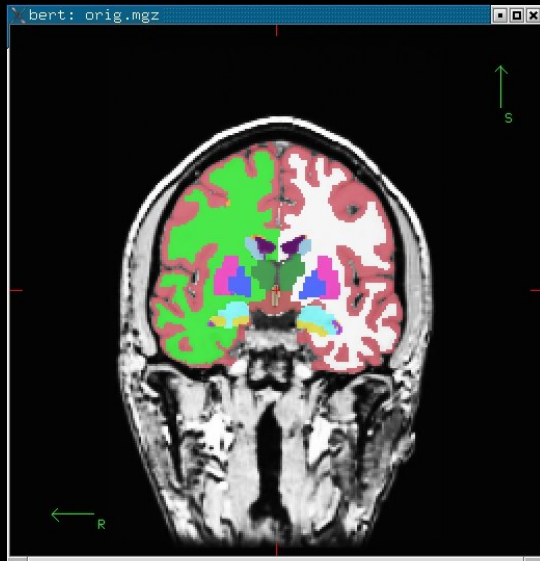
-fthresh 2.3 -fmid 3.3 -fslope 1

Can display any functional data, eg,
zstat, fzstat, cope, pe, etc

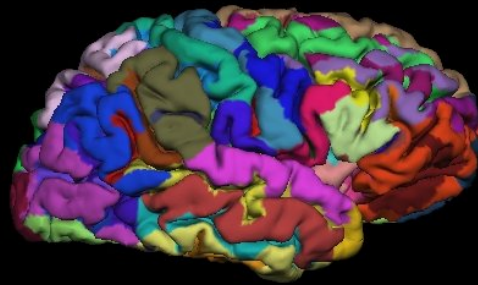
Visual, Auditory, Motor



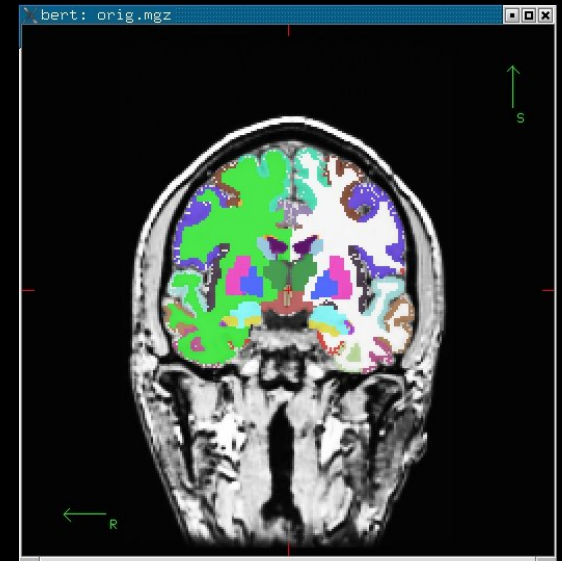
Mapping Automatic Segmentations



aseg.mgz



lh.aparc.annot



aparc+aseg.mgz

`$FREESURFER_HOME/FreeSurferColorsLUT.txt`

Mapping Automatic Segmentations

```
aseg2feat --feat fbert.feats --aseg aparc+aseg
```

In the functional FOV, creates:

fbert1.feats/reg/freesurfer/aparc+aseg.nii.gz

Create Text Summary Table (nvox, mean cope, std cope, etc)

```
mri_segstats --seg fbert1.feats/reg/freesurfer/aparc+aseg.nii.gz  
--nonempty --ctab-default --in fbert.feats/stats/cope1.nii.gz  
--sum fbert1.sum.txt
```

Can summarize any functional data, eg,
zstat, fzstat, cope, pe, etc

Make and View ROI

Make a binary mask of the left putamen:

Note: 12 = Left Putamen,

see \$FREESURFER_HOME/FreeSurferColorsLUT.txt

fslmaths

```
./fbert1.feats/reg/freesurfer/aparc+aseg.nii.gz
```

```
-thr 12 -uthr 12
```

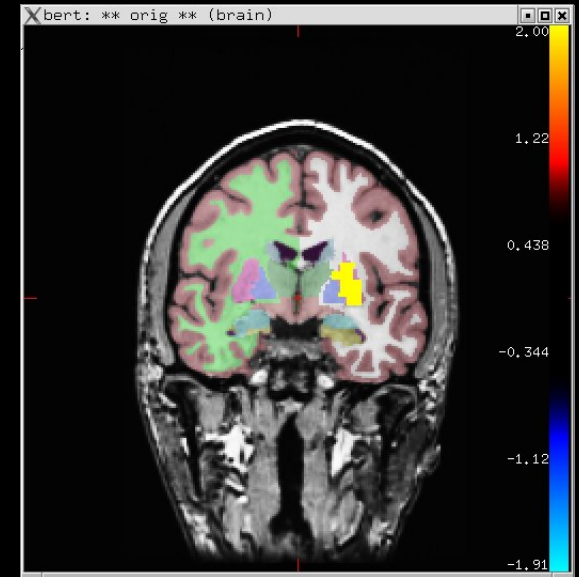
```
./fbert1.feats/reg/freesurfer/lh.putamen.nii.gz
```

```
tkmedit bert orig.mgz -aux brain.mgz
```

```
-overlay ./fbert1.feats/reg/freesurfer/lh.putamen.nii.gz
```

```
-overlay-reg ./fbert1.feats/reg/freesurfer/anat2exf.register.dat
```

```
-fthresh 0.5 -seg aparc+aseg.mgz
```



Within-Subject, Cross-Run Analysis

- Analyze each run with FEAT (dataX.feats)
- Combine runs with GFEAT (standard space)
- mean_func.nii.gz – avg of example_func in std space
- Register each run (not .gfeats) with reg-feat2anat.
- dataX.feats/reg/freesurfer/anat2std.register.dat
- All runs (X) should be very close

Verify the registration

```
tkregister2 -mov data.gfeats/mean_func.nii.gz  
--reg anat2std.register.dat --surf
```

Use anat2std.register.dat the way you would anat2exf.register.dat

Surface-based Group Analysis

```
mris_preproc --out lh.cope1.nii.gz --target fsaverage --hemi lh
--iv fbert.feats/stats/cope1.nii.gz fbert.feats/reg/freesurfer/anat2exf.register.dat
--iv fgreg.feats/stats/cope1.nii.gz fgreg.feats/reg/freesurfer/anat2exf.register.dat
--iv fsally.feats/stats/cope1.nii.gz fsally.feats/reg/freesurfer/anat2exf.register.dat
--iv fpat.feats/stats/cope1.nii.gz fpat.feats/reg/freesurfer/anat2exf.register.dat
```



Volumes

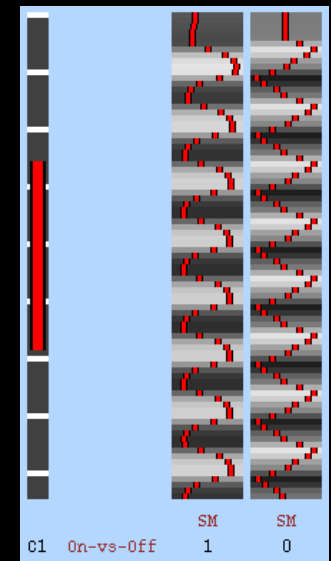
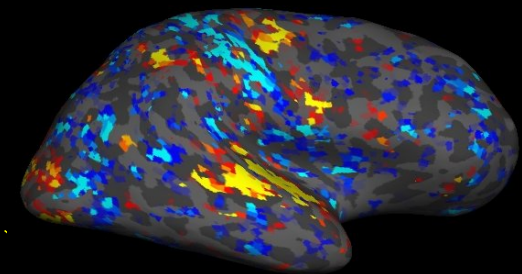
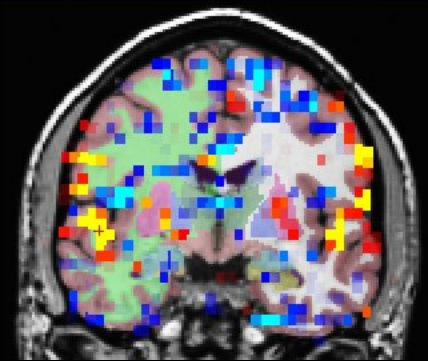


Registrations

- lh.cope1.nii.gz – “volume-encoded surface file”: $\text{dim } 1974 \times 1 \times 83 = 163842$
- dim nframes = number of subjects
- Can map any functional data, eg, zstat, fzstat, cope, pe, etc
- fsaverage – defines common space (spherical surface)
- mri_glmfit (FreeSurfer), randomise (FSL), flame (FSL), melodic (FSL)

Practical Data

- Sensory-motor study
- Blocked Design (15sec OFF, 15 sec ON)
- One subject – “bert”
- Two runs (TR=3, 85 time points)
- FEAT has been run on both runs (FWHM=5)
- Combined with GFEAT
 - FFX
 - One-Sample Group Mean (OSGM)
- Actually, all analysis steps already performed!



Practical

- Automatic registration (<5 min)
- Manual registration
- View FEAT results on subject's anatomy (aparc+aseg)
- View FEAT results with tksurfer
- Map aparc+aseg to Functional Space
- Verify GFEAT registration
- View GFEAT results in volume and on surface
- “Higher-Level” analysis with mri_glmfit
 - Cross-Run
 - Fixed-Effects (FFX), One-Sample Group Mean (osgm)