SNSS Structural Module: Tutorial 2 – Exercises

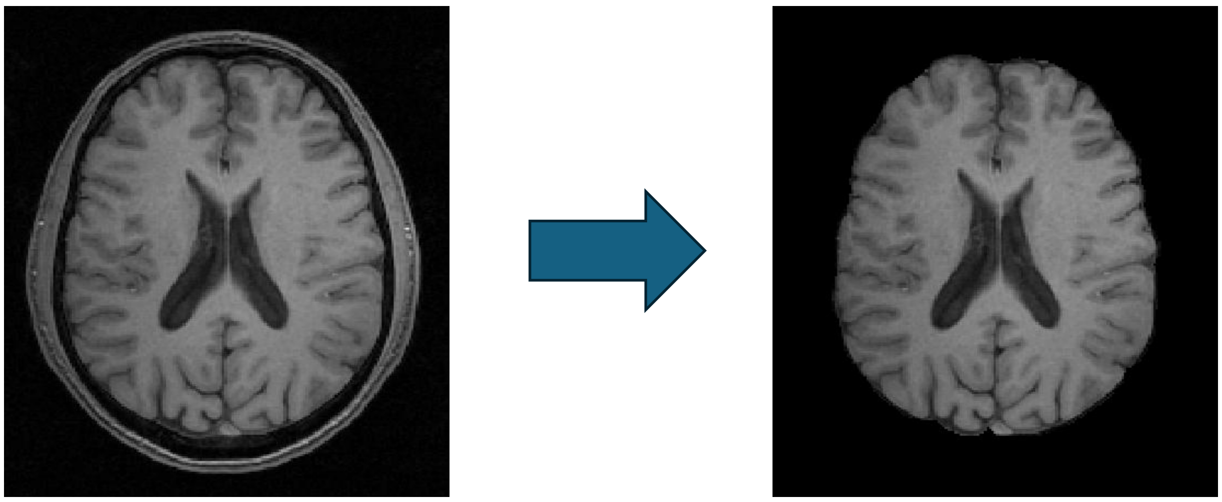
Tissue Segmentation and Getting Tissue Metrics

By: Peter Truong

Reference Information provided from FSL courses from FMRIB at Oxford (FSL, FSL Course Material) and MIT (FSL).

Brain Extraction

The presence of skull, fat, and skin will negatively impact any quantitative data we wish to obtain. Thus, it's best to work with a brain that has any non-brain removed from the images.



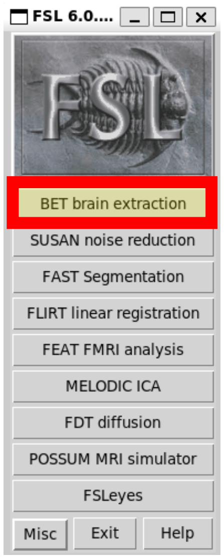
Exercise 1.1: Running BET (GUI)

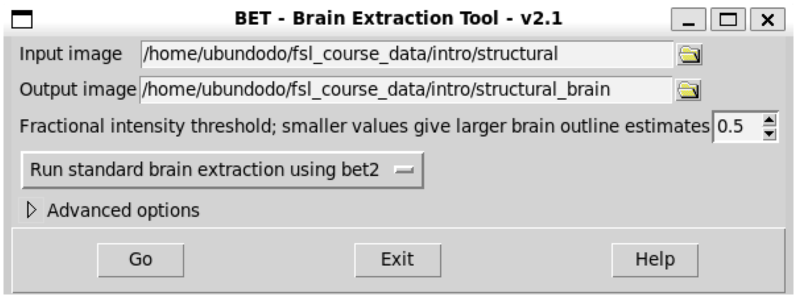
There are two ways to run FSL's Brain Extraction Tool (BET). You can run it using a graphical user interface or through command line. Either way will work, but depending on your overall processing and analysis pipeline, it might be more efficient to run things via command line.

First you want to open **terminal** (you can press "CTRL+ALT+T"), then you want to navigate to the directory (if using the VM):

/home/user/Desktop/distribution_data/tutorial2_data/exercise1_BRAIN-EXTRACT

Open up FSL by typing in "**fsl &**" in the terminal. From the FSL selection window, click on the "**BET brain extraction**" box. This will open a new window for **BET.**




BET - Brain Extraction Tool - v2.1
Run standard brain extraction using bet2
Advanced options
Go
Exit
Help

Input image /home/ubundodo/fsl_course_data/intro/structural

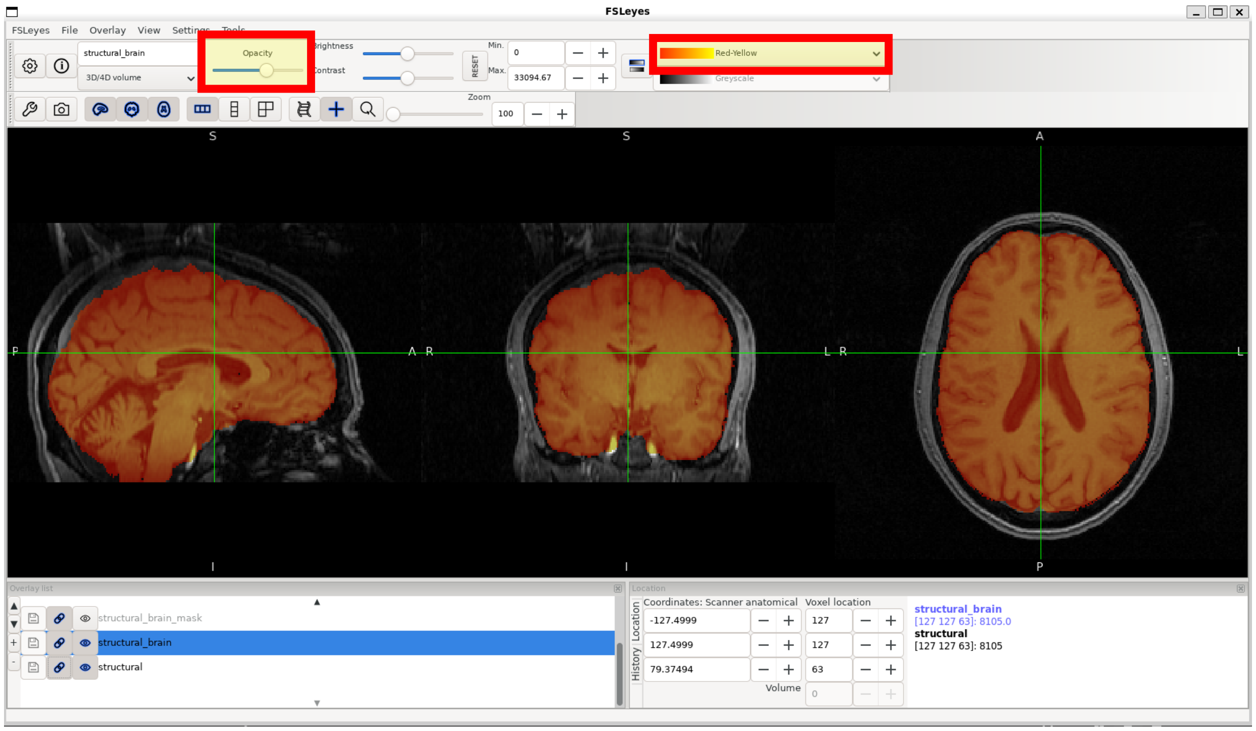
Output image /home/ubundodo/fsl_course_data/intro/structural_brain

Fractional intensity threshold; smaller values give larger brain outline estimates 0.5

For the input image, select **structural.nii.gz**. This will automatically set the output image with the name **structural_brain**. You can change the name, along with the output directory, to your choosing.

After BET finishes, you can verify the results by overlaying it onto the structural image in **FSLeyes.**

<u>Note</u>: the setting "**Fractional intensity threshold**", allows you to adjust how much is removed. If you see that there's too much non-brain in your output, increase this value. If you notice that it's removing too much brain, lower this value.



structural_brain
3D/4D volume
Opacity
Brightness
Contrast
Min. 0
Max. 33094.67
Red-Yellow
Greyscale
Zoom 100

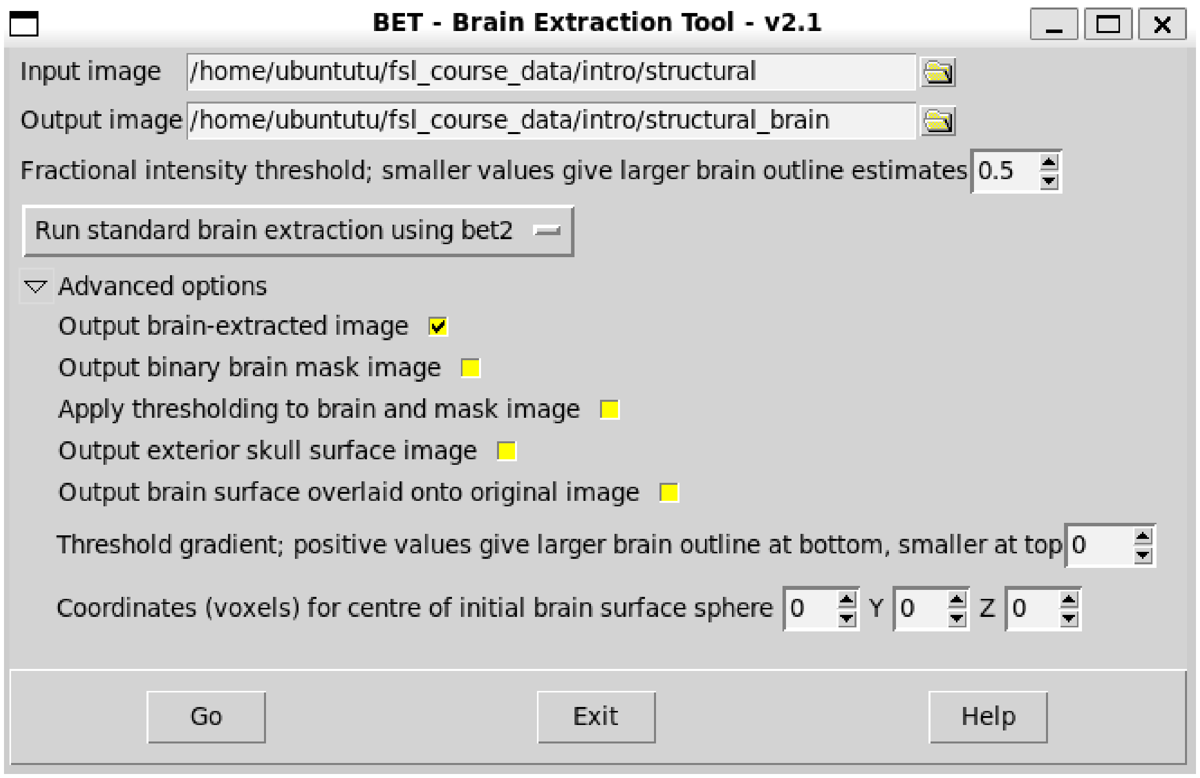
structural_brain_mask
structural_brain
structural

Coordinates: Scanner anatomical | Voxel location
-127.4999 | 127
127.4999 | 127
79.37494 | 63
Volume | 0

structural_brain
[127 127 63]: 8105.0
structural
[127 127 63]: 8105

<u>Note:</u> By selecting "**structural_brain**" in the **Overlay list** window (bottom left), you can change the opacity and colour of the image. Here I turned the opacity down so the image is translucent and changed the color from Greyscale to Red-Yellow.

By default, it will output the brain extracted image. However, there are a few more options you can enable by clicking on **"Advanced options"**.



BET - Brain Extraction Tool - v2.1

Input image /home/ubuntutu/fsl_course_data/intro/structural

Output image /home/ubuntutu/fsl_course_data/intro/structural_brain

Fractional intensity threshold; smaller values give larger brain outline estimates 0.5

Run standard brain extraction using bet2

Advanced options

Output brain-extracted image ✓

Output binary brain mask image

Apply thresholding to brain and mask image

Output exterior skull surface image

Output brain surface overlaid onto original image

Threshold gradient; positive values give larger brain outline at bottom, smaller at top 0

Coordinates (voxels) for centre of initial brain surface sphere 0 Y 0 Z 0

The additional outputs you can generate are:

- **Binary brain mask image**
- **Exterior skull surface image**
- **Brain surface overlaid onto original image**

For this portion of the exercise, we won't be running BET in GUI to generate these outputs. We will try to enable these outputs through running BET in command line.

Exercise 1.2: Running BET (Command Line)

In terminal, you can type in:

```
bet <input> <output> [options]
```

You can see all the options by entering "**bet -h**". Have a look through to see which settings we need to enable to output: **binary brain mask**, **brain surface overlaid onto original image**, and **skull surface image**. Also, we will set the output name to be "**structural_brain_CL**" to differentiate it from the previous results.

Some of the options corresponding to the ones shown above are:

-o: generate brain surface outline overlaid onto original image

-m: generate binary brain mask

-s: generate approximate skull image

-f <value>: fractional intensity threshold (0->1)

With default settings you can run it using:

```
bet structural structural_brain_CL
```

Adding in the optional outputs:

```
bet structural structural_brain_CL -f 0.5 -o -m -t -s
```

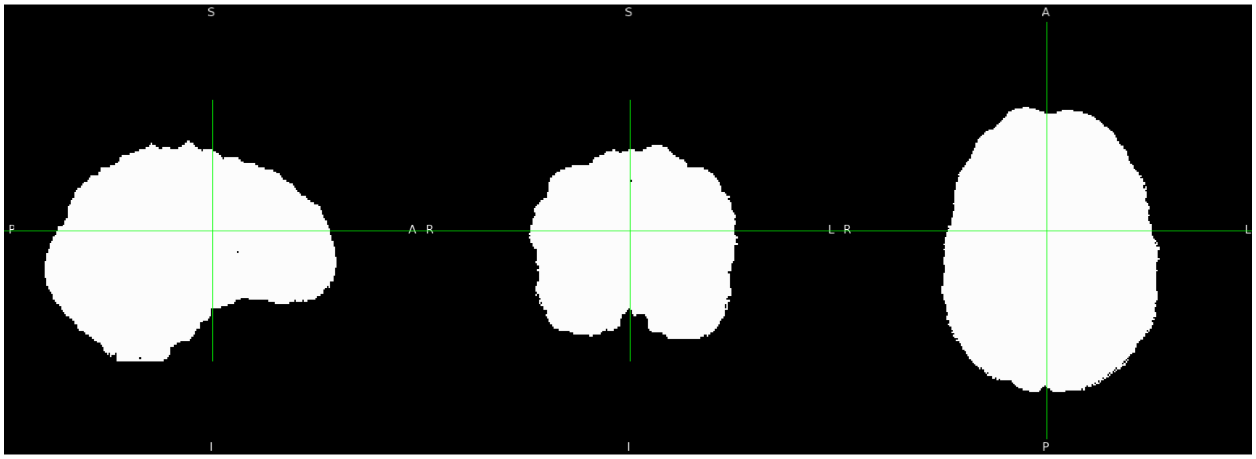
To overlay the brain extracted image onto the input image:

```
fsleyes structural structural_brain_CL &
```

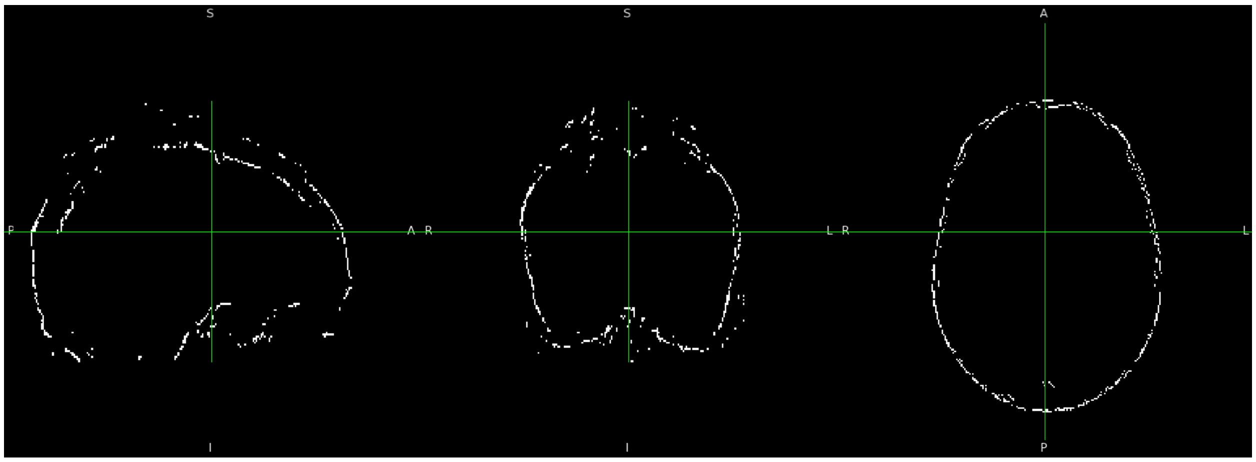
Have a look through the additional outputs as well.

Other BET Outputs:

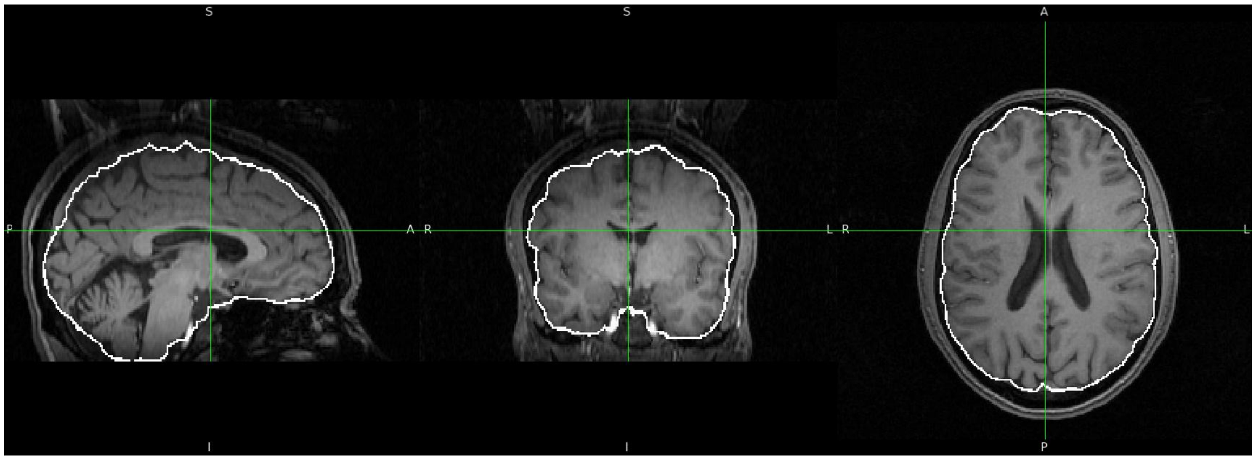
Structural_brain_mask.nii.gz: a binary image showing areas where there is brain (set as 1) and non-brain (set as 0)



Structural_brain_skull.nii.gz: estimate of the exterior surface of the skull



structural_brain_overlay.nii.gz: output image will be a copy of the input image, but with the outline of the area where the brain has been extracted. This will be similar to overlaying the brain-extracted image onto structural image, where you can see how BET set the boundary between brain and non-brain.



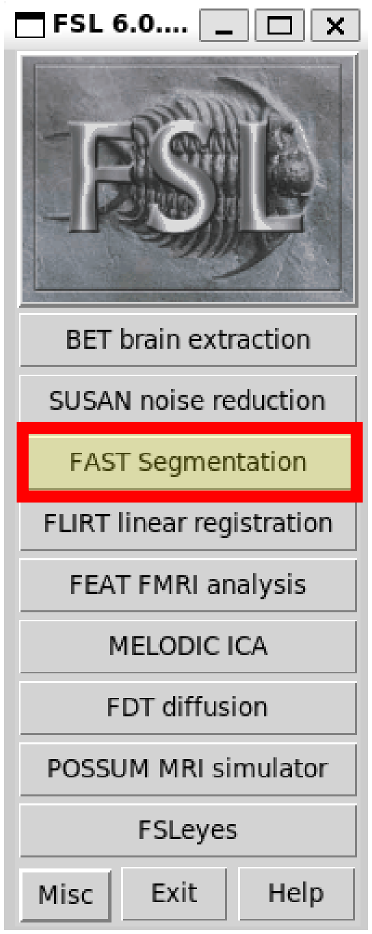
Tissue Segmentation

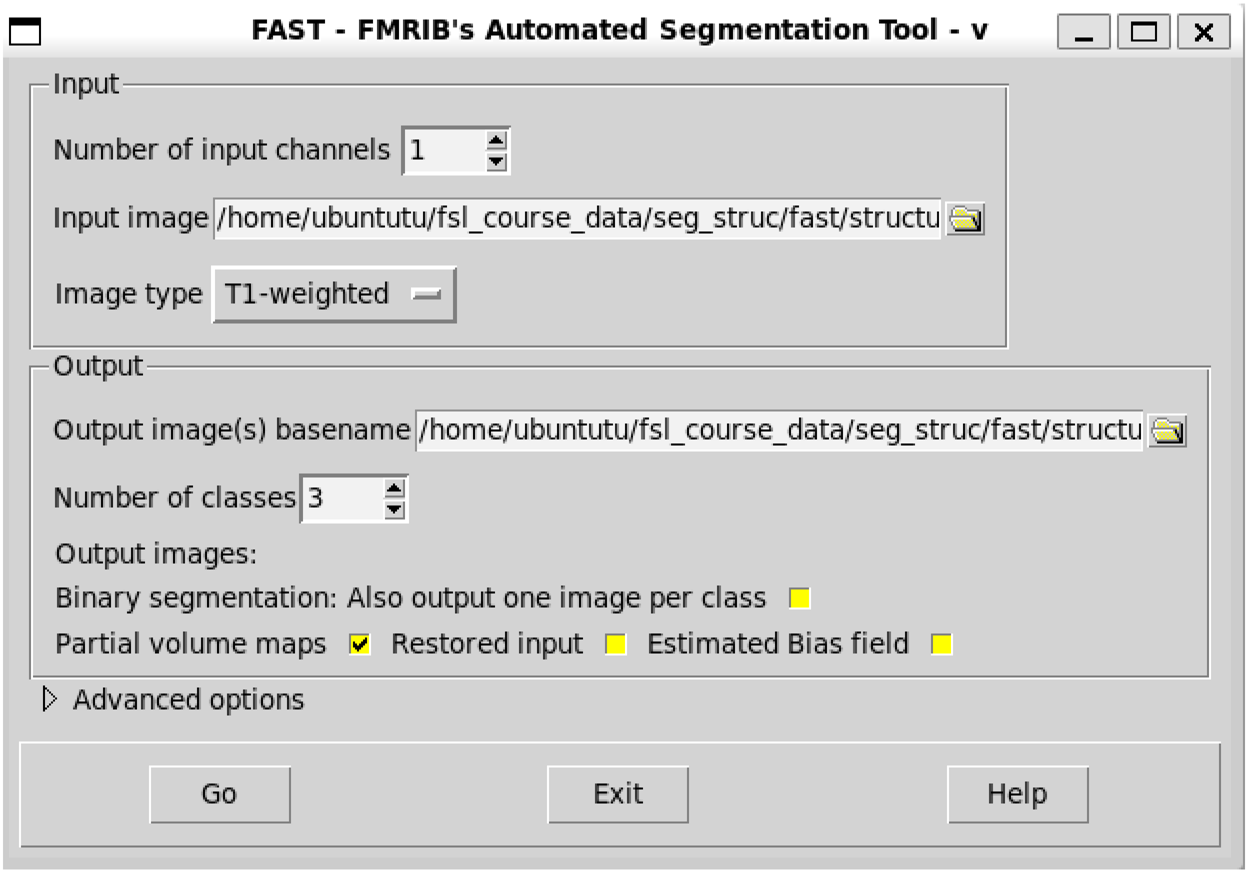
Now that we have only brain, we can separate it further based on tissue type. Launch FSL (**fsl &**) from the directory on the VM:

/home/user/Desktop/distribution_data/tutorial2_data/exercise2_SEGMENT

Before running we segment the image, we must first run **BET**. Tissue segmentation uses each voxel's intensity value to characterise its corresponding tissue(s) type. By having the skull, skin, and fat present in the image, segmentation will include them into the characterization, giving you incorrect results.

Select **FAST Segmentation** option in the FSL menu.





FAST - FMRIB's Automated Segmentation Tool - v

Input

Number of input channels 1

Input image /home/ubuntutu/fsl_course_data/seg_struc/fast/structu

Image type T1-weighted

Output

Output image(s) basename /home/ubuntutu/fsl_course_data/seg_struc/fast/structu

Number of classes 3

Output images:

Binary segmentation: Also output one image per class ☐

Partial volume maps ☑ Restored input ☐ Estimated Bias field ☐

The segmentation routine takes around 5 to 10 minutes for a full image. For the purposes of this tutorial, we will trim down **structural_brain.nii.gz**, so we can test out a few options of **FAST segmentation**.

Exercise 2.1: Runing FAST (GUI)

BETing The Image

The first thing we want to do is to run **BET** on our structural image.

Reduce the Image Size using FSLROI

FSL has a function called **fslroi** that allows us to trim down an image. For us to accurately run this function, we need to know the image's dimensions. You can use **fslsize <input file>**.

From inputting "**structural_brain**", we can see the dimensions are:

dim1 (x size) is **174**

dim2 (y size) is **192**

dim3 (z size) is **192**

Let's trim down the z size to **5 slices**, starting at **position 100**. The inputs for **fslroi** are as follows:

```
fslroi <input file> <output name> <xmin> <xsize> <ymin> <ysize> <zmin> <zsize>
```

In the terminal, run:

```
fslroi structural_brain structural_brain_roi 0 174 0 192 100 5
```

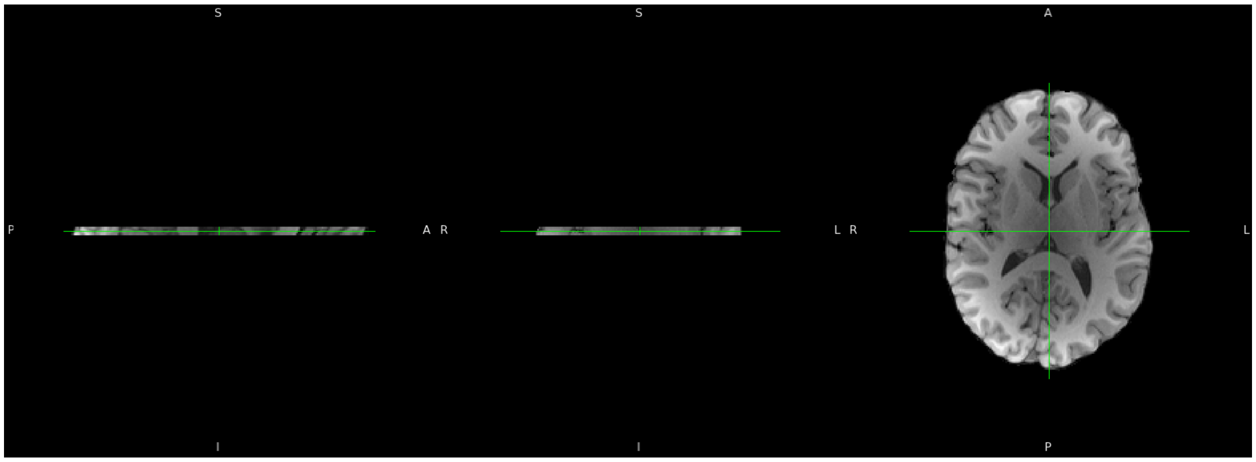
Let's look at the size of **structural_brain_roi**. We can see that it now has the dimensions:

dim1 (x size) is **174**

dim2 (y size) is **192**

dim3 (z size) is **5**

Load "**structural_brain_roi**" in **FSLeyes** to see the trimmed-down image.



Segmentation:

Now, back to segmentation, select “**structural_brain_roi.nii.gz**” as the input image. This image type is a T1 image, but you could also input T2 and PD image types as well for segmentation.

Press “**Go**” and the segmentation routine will run.

There are Advanced options as well, but it’s recommended to just use the default settings.

The outputs from **FAST segmentation** are:

- **structural_brain_roi_seg.nii.gz**: the binary segmentation image has the GM, WM, and CSF segmented
- **structural_brain_roi_pve_0.nii.gz:** partial volume estimate image of the CSF
- **structural_brain_roi_pve_1.nii.gz:** partial volume estimate image of the GM
- **structural_brain_roi_pve_2.nii.gz:** partial volume estimate image of the WM
- **structural_brain_roi_pveseg.nii.gz:** best binary segmentation image following partial volume estimates. Use for quick assessment of segmentation.
- **structural_brain_roi.mixeltype.nii.gz**: image which classifies pure and mixed voxels represented by different values in voxels with mixed voxel type

Have a look at the different images and/or try overlaying the partial volume maps.

Exercise 2.2: Running FAST in Command Line & Getting Tissue Volume Estimates

FAST in Command Line:

The basic structure to use FAST on command line is:

fast [options] file(s)

The equivalent command line command is:

fast -t 1 -n 3 -H 0.1 -I 4 -l 20.0 -o <output file> <input file>

The options correspond to:

[-t 1]: Image contrast type – either 1 (T1), 2 (T2), or 3 (PD); default is 1

[-n 2]: Number of tissue-type classes to be segmented; default is 3 for GM, WM, and CSF output. For T2 images you might have to output 4.

[-H]: Segmentation spatial smoothness; default is 0.1

[-I 4] (uppercase i): Number of main-loop iterations during bias-field removal; default is 4

[-l 20.0] (lowercase L): Bias field smooth extent (FWHM) in mm; default is 20.0

[-o <output file>]: filename/path of the output file; default is the same name as input file with typical output file suffix

However, a few of these options are defaults already. Although it is good to include them to ensure proper functioning of your command, we can simplify this further:

f**ast <input file>**

FSLSTATS for getting Tissue Volumes:

By using **fslstats**, we can get tissue volume estimates from the partial volume images. This is solely done on command line. Looking at the options for **fslstats** there are two that are of interest for us: <**-V**>, to output the volume (number of non-zero voxels), and <**-M**>, to output the mean of the voxel's intensities. By multiplying two, we get the volume of tissue in the image.

vol=`$FSLDIR/bin/fslstats structural_brain_CL_pve_0 -V | awk '{print $2}'`

This command runs FSLSTATS, getting the total number of non-zero voxels in the image. Then saves the second output under the variable "**vol**".

Please note that the <` > character is the grave accent or backtick. In standard keyboards, this key is to the left of the "**1**" in the top number row.

The output from running **"fslstats -V**" it will output two numbers. The first number will output the volume with respect to the number of voxels. The second number will output the volume in mm^3.

By using "**print $2**", it ensures that we are outputting the volume in mm^3. If you want to output volume with respect to number of voxels, use "**print $1**".

mean=`$FSLDIR/bin/fslstats structural_brain_CL_pve_0 -M`

Here, the output from “**fslstats -M**” will be saved under the variable “**mean**”.

tissuevol=`echo “$mean * $vol” | bc -l`

The variable “**tissuevol**” will be the multiplication of variables “**mean**” and “**vol**”**.** This uses a calculator to perform floating-point math “**bc -l**”.

echo $tissuevol

We use “**echo**” to output the value of “**tissuevol**” on command line.

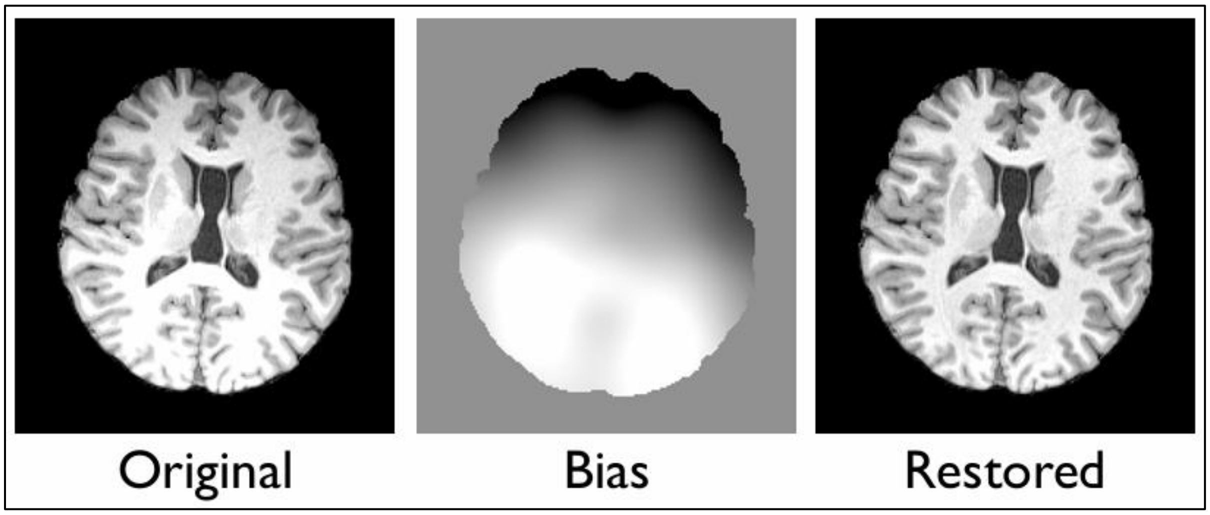
This can also be condensed into one line via:

fslstats structural_brain_roi_pve_0 -M -V | awk ‘{ print $1 * $3 }’

Try this out and see what volumes you get for the GM, WM, and CSF.

Bias Fields

Bias fields occur due to inhomogeneous RF transmission. It results in uneven intensities/contrast throughout the image. This can be due to the RF coil penetration depth, poor shim, and is exacerbated when going to higher field strengths.



By default, running FAST via GUI and command line will automatically have bias field correction enabled, which for most cases is preferable. To see the estimated bias field as an output, you can enable "**Estimated Bias field**" as one of the output images. As well, you can output a bias field corrected image of the input image by enabling "**Restored input**".

Let's have a look at the effect of correcting and not corrected for the bias field in our data. This will serve as a good way for us to practice using **FAST** and **FSLSTATS**.

Exercise 3: Bias Fields and FAST with 7T Data

The data for this exercise is in:

/home/user/Desktop/distribution_data/tutorial2_data/exercise3_BIAS

Within the same folder, you can see the structural image at 7T: "**structural_brain_7T.nii.gz**". This image has already been skull stripped.

Open "**structural_brain_7T.nii.gz**" in FSLeyes. Notice the difference in intensity and contrast between white and grey matter in the anterior region of the brain versus the posterior region. You can see that the posterior region is brighter compared to the anterior region of the 7T image.

Once again, let's use **fslroi** to trim down the image, so we can quickly run through **fast**. Trim down the axial dimension (dim3) to 5 slices starting at position 100, call the output "**structural_brain_7T_roi**".

Now using **fast GUI** segment “**structural_brain_7T_roi.nii.gz**” with “**Restored input**” and “**Estimated Bias field**” enabled.

Once segmentation is complete, compare “**structural_brain_7T_roi.nii.gz**” with “**structural_brain_7T_roi_restored.nii.gz**”. See the difference in image contrast and intensity. Open “**structural_brain_7T_roi_bias.nii.gz**”, to see the image of the estimated bias field.

Since bias field correction is automatically performed using FAST, let’s perform segmentation without it to see its effect on the data. The only way to turn this step off is through command line.

In terminal, run fast, but this time set option **<-N>** for no bias field correction.

```
fast -N -o structural_brain_7T_wbias structural_brain_7T_roi
```

Get the tissue volume estimates from “**structural_brain_7T_roi**” and “**structural_brain_7T_wbias**”.

You can see a large disparity in the outputs, highlighting the importance of bias field correction.

<u>Subcortical Segmentation</u>

If you were primarily interested in subcortical structures, such as the hippocampus, thalamus, or putamen, FSL has the capability to segment these structures and from these results you can perform vertex and volumetric analysis.

Exercise 4.1: Subcortical segmentation with FIRST

FSL supplies a script called **run_first_all** which performed all the necessary routines and optimizations needed. This script sequentially runs a few functions such as: **first_flirt**, **run_first**, and **first**. By default, it will reference your input image to the "**MNI152_T1_1mm**" standard located in "**$FSLDIR/data/standard**".

run_first_all [options] -i <input image> -o <output name>

*Only T1-weighted images can be used

Options:

<**-m**>: boundary correction method; default is **auto**. Other options are: **fast**; **thresh**; or **none**

<**-s**>: specifies which subcortical structure(s) is segmented. Options are:

- Putamen ([L/R]_Puta)
- Caudate nucleus ([L/R]_Caud)
- Nucleus accumbens ([L/R]_Accu)
- Globus pallidus ([L/R]_Pall)
- Hippocampus ([L/R]_Hipp)
- Amygdala ([L/R]_Amyg)
- Thalamus ([L/R]_Thal)
- Brainstem (BrStem)

<**-b**>: the input image is already brain-extracted.

<**-a**>: input affine registration. Input the affine matrix, if you previously ran **first_flirt** for subcortical registration.

The practice data is located on the VM in:

/home/user/Desktop/distribution_data/tutorial2_data/exercise4_SUBCORTICAL

We will run the command:

```
run_first_all -i con0047_brain -o con0047 -a con0047_brain_to_std_sub.mat
```

The option **<-a>** allows you to supply the affine registration. This will help speed up running this script. Without this option, the script will run affine registration automatically using **MNI152_1mm** as the standard, which will take longer to complete.

A few of the outputs you will get after running **run_first_all**:

con0047_all_fast_firstseg.nii.gz: Single imaging showing all segmented output of the structure(s). Structures are boundary corrected to ensure no overlap. Each structure is labelled with a specific integer label, which can be used afterwards to get volumes of desired subcortical structures.

con0047_all_fast_origsegs.nii.gz: 4D image containing the individual structure segmentations. Each structure will have its own 3D image, and you will be able to cycle through all the individual segmented structures. Structures are not corrected for boundary overlap.

con0047_*_first.vtk: mesh representation of the final segmentation

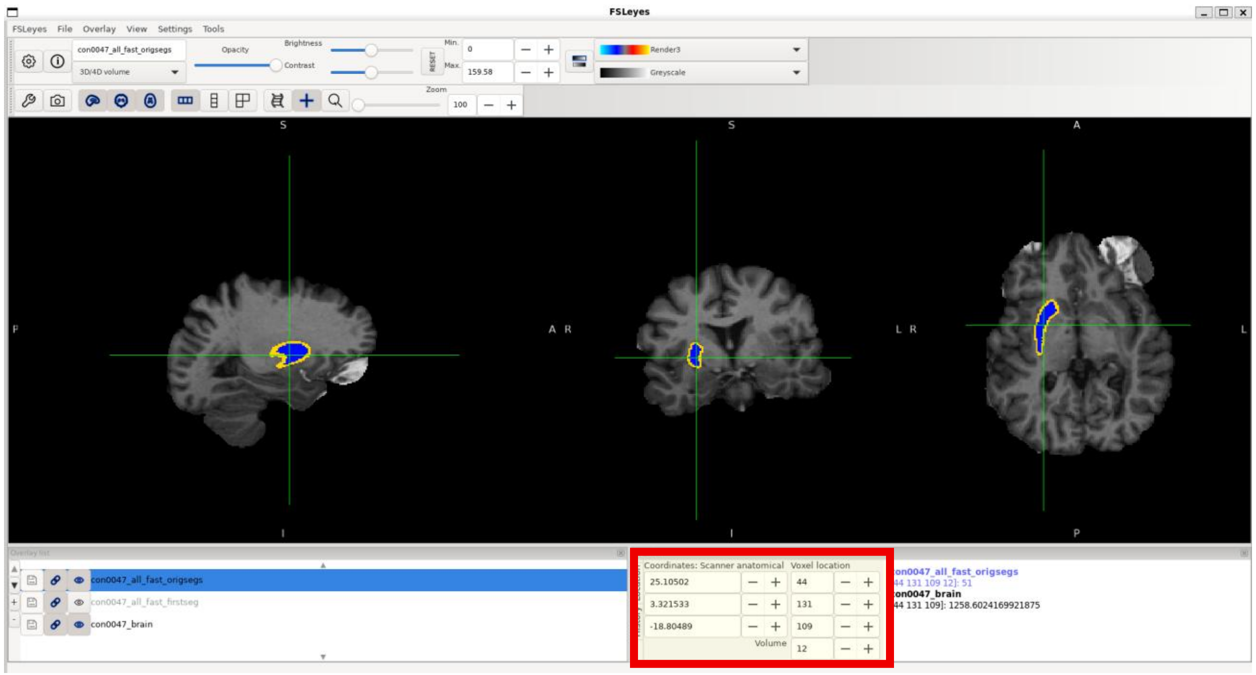
con0047_*_first_bvars: Don't delete this file. Contains the mode parameters and the model used. This along with the model files, can be used to reconstruct the other outputs. This can be used later to perform vertex analysis or as a shape prior to segment other shapes.

To verify that the registration and segmentation were successful, type in the command line:

```
cat *.logs/*.e*
```

If no output is shown, there are no errors.

Open the original structural image “**con0047_brain.nii.gz**” and overlay it with the newly created “**con0047_all_fast_origsegs.nii.gz**”. You can switch between different structures by changing the “**volume**” selection in the middle-bottom “**coordinates**” window.



con0047_all_fast_origsegs
3D/4D volume
Min. 0
Max. 159.58
Render3
Greyscale
Zoom 100

con0047_all_fast_origsegs
con0047_all_fast_firstseg
con0047_brain

Coordinates: Scanner anatomical	Voxel location
25.10502	44
3.321533	131
-18.80489	109
Volume	12

con0047_all_fast_origsegs
[44 131 109 12]: 51
con0047_brain
[44 131 109]: 1258.6024169921875

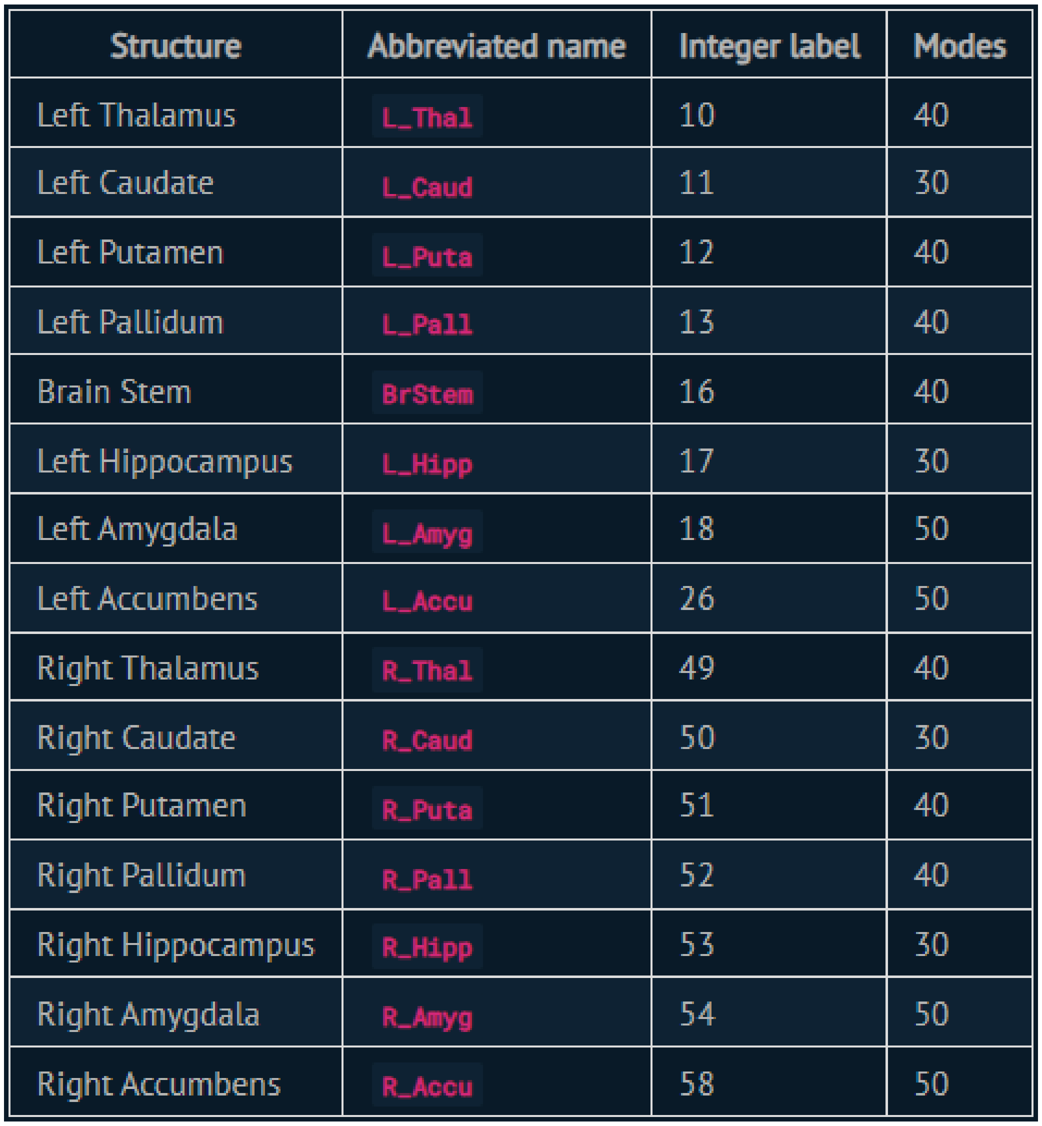
Note: it is critical to check the registration results as **run_first_all** will run and generate outputs even if the registration stage failed in the routine!

Exercise 4.2: Volumetric Analysis of Subcortical Structures

We can use **fslstats** to get volume metrics from each subcortical structure that was segmented. Here we will be using **fslstats** on **con0047_all_fast_firstseg.nii.gz**. You might remember that this is a 4D image, with each subcortical structure given a different label. We can use this label number to specify which structure we will analyze.

```
fslstats con0047_all_fast_firstseg.nii.gz -l 16.5 -u 17.5 -v
```

Options **-l** and **-u** adjust the threshold to select a single label number. In this example, the label number is 17 for the left hippocampus, so we set the lower and upper threshold to be 16.5 and 17.5, respectively.



Structure	Abbreviated name	Integer label	Modes
Left Thalamus	L_Thal	10	40
Left Caudate	L_Caud	11	30
Left Putamen	L_Puta	12	40
Left Pallidum	L_Pall	13	40
Brain Stem	BrStem	16	40
Left Hippocampus	L_Hipp	17	30
Left Amygdala	L_Amyg	18	50
Left Accumbens	L_Accu	26	50
Right Thalamus	R_Thal	49	40
Right Caudate	R_Caud	50	30
Right Putamen	R_Puta	51	40
Right Pallidum	R_Pall	52	40
Right Hippocampus	R_Hipp	53	30
Right Amygdala	R_Amyg	54	50
Right Accumbens	R_Accu	58	50