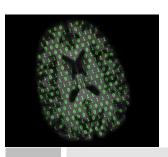


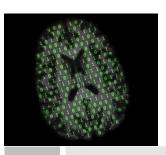
Getting ROI Anatomic Information from Non-Linear Registration

- We have the ROI in the template space
- We want to get information about its location
- This is possible if the atlas is labeled

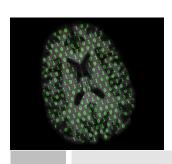


Reading in the Atlas and Look Up Table (LUT)

```
#### extracting JHU Eve atlas Type I and labels############
atlas = "JHU MNI SS WMPM Type-I"
txtfile = file.path(neurodir, "Template",
                    paste() (atlas, "SlicerLUT.txt"))
### read look up table (LUT)
jhut1.df = read.table(txtfile, stringsAsFactors=FALSE)
jhut1.df = jhut1.df[, 1:2]
colnames(jhut1.df) = c("index", "Label")
jhut1.df$index = as.numeric(jhut1.df$index)
jhut1.df[1:4,]
index
                              Label
                           Background
1
      1 SUPERIOR PARIETAL LOBULE left
3
                 CINGULATE GYRUS left
      3 SUPERIOR FRONTAL GYRUS_left
4
```

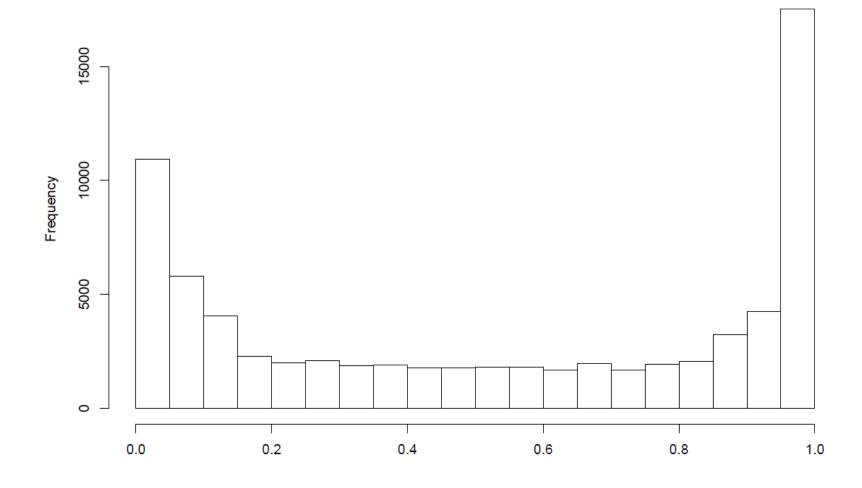


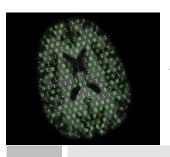
Reading in the Atlas and Look Up Table (LUT)



After Warping and Interpolation, the ROI is No Longer Binary

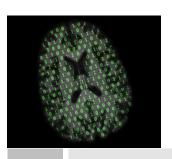
hist(c(syn_roi[syn_roi > 0]))



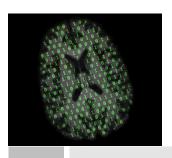


Ways of Dealing with Non-Binary ROI

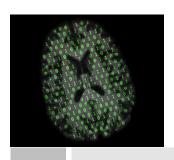
- Threshold the ROI to binary
- ☐ Use a weighted sum over the ROI



Quantifying the ROI Engagement by Region



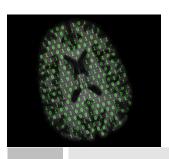
ROI engagement: Tumor by Region



ROI Engagement Results: Top 10 Regions

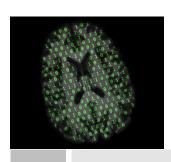
```
# Calculate the percent of the tumor engaging the region
label_pct = t(t(label_sums)/colSums(label_sums)) * 100
head(round(label_pct, 1), 10)
```

	sum_rol	sum_roi_thresh
PRECENTRAL_GYRUS_left	15.6	15.6
PRECENTRAL_WM_left	12.9	12.9
SUPERIOR_TEMPORAL_GYRUS	9.4	9.9
INSULAR	9.2	9.7
<pre>INFERIOR_FRONTAL_WM_left</pre>	7.4	7.5
Superior_longitudinal_fasciculus_left	7.2	7.3
POSTCENTRAL_GYRUS_left	6.9	6.9
<pre>INFERIOR_FRONTAL_GYRUS_left</pre>	6.0	6.2
SUPRAMARGINAL_GYRUS	5.7	6.0
MIDDLE_FRONTAL_GYRUS_left	4.8	4.3



ROI Engagement: Region by Tumor

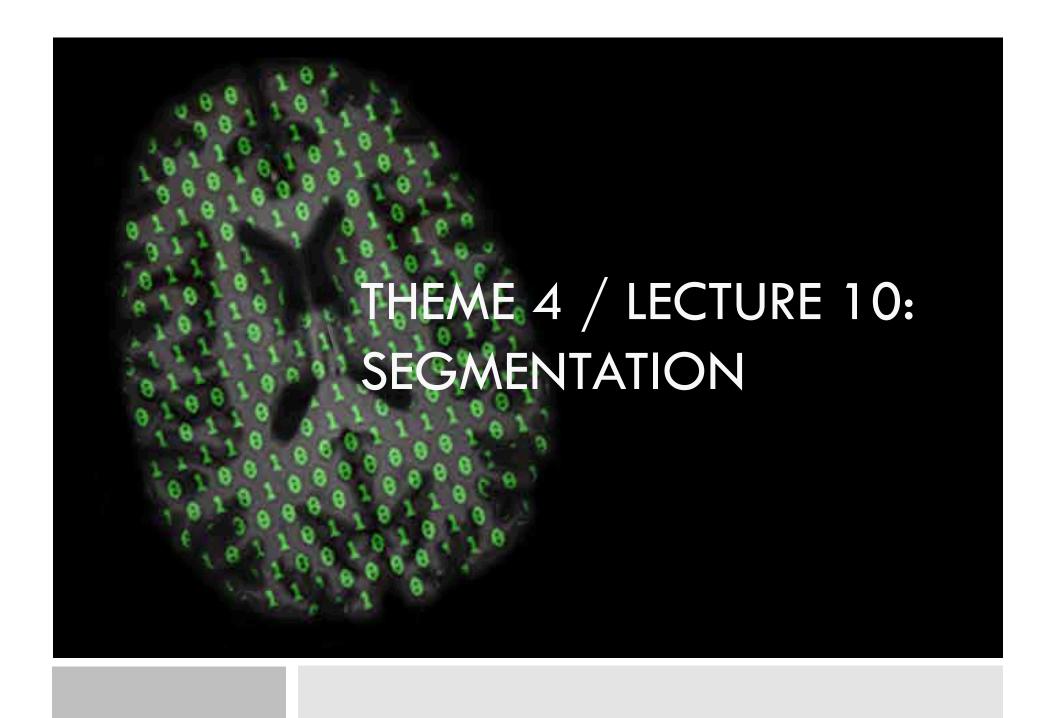
```
# Create a table that contains the number of voxels
#engaged in each labeled region
jhut1.tab = as.data.frame(table(c(jhut1.img)))
colnames(jhut1.tab) = c("index", "size")
# Merge the table of number of voxels per region with the
# table of number of voxels by ROI
region pct = merge(sums, jhut1.tab, by="index")
rownames (region pct) = sums$Label
# Calculate the percent of region engaged by the tumor
region pct$Label = region pct$index = NULL
region pct = region pct/region pct$size * 100
region pct$size = NULL
# Reorder regions from the most to the least engaged
region pct = region pct[ order(region pct$sum roi,
decreasing = TRUE), ]
```



ROI Engagement Results: Top 10 Regions

Calculate the percent of the tumor engaging the region
head(round(region_pct,1),10)

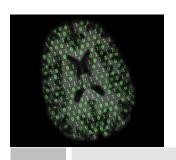
	sum_roi	sum_roi_thresh
Superior_longitudinal_fasciculus_left	46.3	46.4
PRECENTRAL_WM_left	34.9	34.5
INSULAR	31.2	32.4
External_capsule_left	27.5	27.5
<pre>INFERIOR_FRONTAL_WM_left</pre>	25.4	25.4
PRECENTRAL_GYRUS_left	23.8	23.6
SUPRAMARGINAL_GYRUS	12.6	13.0
SUPERIOR_TEMPORAL_GYRUS	11.3	11.6
<pre>INFERIOR_FRONTAL_GYRUS_left</pre>	11.1	11.3
PUTAMEN_left	11.1	10.2





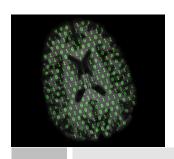
Segmentation

- We show how to use fslr to
 - segment CSF, GM, and WM
 - calculate the volume of CSF, GM, and WM



Functions Used in Segmentation

- readNifti: passes files as NIfTI objects
- bias_correct: wraps ANTsR bias field corrections
- fslbet_robust: performs skull-stripping
- fast: calls fast from FSL, does segmentation
- ortho2 does orthographic display



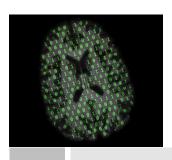
Data Preparation

```
mridir = file.path("/home/fsluser/Desktop/MOOC-2015/
kirby21/visit_1/113")
t1_path=file.path(mridir,"113-01-MPRAGE.nii.gz")

#Read the file
nim=readNIfTI(t1_path, reorient=FALSE)

#Conduct bias field correction
fast_img = fsl_biascorrect(nim, retimg=TRUE)

#Perform brain extraction
bet = fslbet(infile=fast img, retimg=TRUE)
```



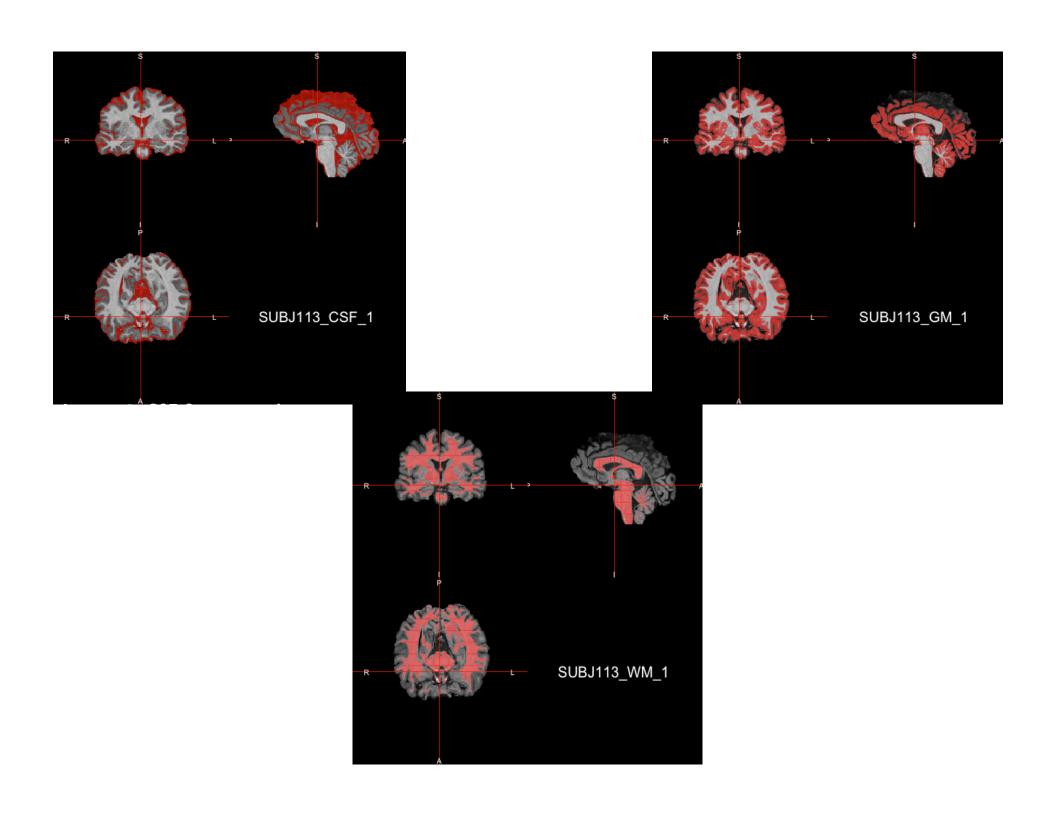
Brain Segmentation

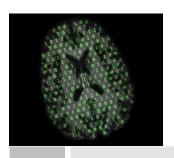
```
#Perform segmentation
fast=fast(file=bet_fast,outfile=file.path(paste0(mridir,"/
113-01-MPRAGE_biascorrected_BET_FAST.nii.gz")))

#Displays CSF segmentation
ortho2(bet,fast==1,col.y=alpha("red",
0.5),text="SUBJ113_CSF_1")

#Displays GM segmentation
ortho2(bet,fast==2,col.y=alpha("red",0.5),text="SUBJ113_GM_1")

#Displays WM segmentation
ortho2(bet,fast==3,col.y=alpha("red",0.5),text="SUBJ113_WM_1")
```



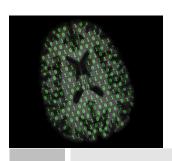


Output Files from fslr Segmentation

113-01-MPRAGEAST_mixeltype.nii.gz	Today, 9:03 PM	359 KB	GZip
113-01-MPRAGET_FAST_pve_0.nii.gz	Today, 9:03 PM	719 KB	GZip
113-01-MPRAGET_FAST_pve_1.nii.gz	Today, 9:03 PM	1.2 MB	GZip
113-01-MPRAGET_FAST_pve_2.nii.gz	Today, 9:03 PM	590 KB	GZip
113-01-MPRAGEFAST_pveseg.nii.gz	Today, 9:03 PM	316 KB	GZip
113-01-MPRAGE_N4_BET_FAST_seg.nii.gz	Today, 9:03 PM	307 KB	GZip
113-01-MPRAGE_N4_BET.nii.gz	Today, 8:17 PM	4.1 MB	GZip
113-01-MPRAGE_N4.nii.gz	Today, 8:10 PM	14.9 MB	GZip

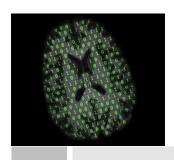
Output files:

- inhomogeneity correction file
- BET file
- probability maps for corresponding tissue classes: pve_0, pve_1, pve_2
- segmentation files



Output Files from fslr Segmentation

- pve files: probability maps where each voxel is assigned the probability of being in a particular tissue class.
 - pve_0 file corresponds to CSF
 - pve_1 file corresponds to grey matter
 - pve_2 file corresponds to white matter
- Values for the same voxel across the pve_0, pve_1, and pve_2
 files sum to 1

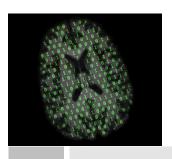


Loading in the pve Files

```
#Reads in the pve file for CSF
pve_CSF=readNIfTI(paste0(mridir,"/113-01-
MPRAGE_N4_BET_FAST_pve_0.nii.gz"))

#Reads in the pve file for GM
pve_GM=readNIfTI(paste0(mridir,"/113-01-
MPRAGE_N4_BET_FAST_pve_1.nii.gz"))

#Reads in the pve file for WM
pve_WM=readNIfTI(paste0(mridir,"/113-01-
MPRAGE_N4_BET_FAST_pve_2.nii.gz"))
```



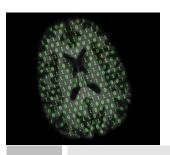
Calculating CSF, GM, and WM Volumes

```
#Reads in the pve file for CSF
threshold=0.33

#Calculate the product of voxel dimensions (volume)
vdim_CSF=prod(voxdim(pve_CSF))

#Reads in the pve file for WM
nvoxels_CSF=sum(pve_CSF>threshold)

#Calculate the volume of CSF in mL
vol_pveCSF=vdim_CSF*nvoxels_CSF/1000
```



Results

```
#CSF volume in mL
vol_pveCSF
[1] 349.9128

#GM volume in mL
vol_pveGM
[1] 914.67

#WM volume in mL
vol_pveWM
[1] 703.698
```

This would need to be redone using a better brain segmentation, using, for example the cog trick