



How to make a composite ROI (and use it!)

So you read that paper when they choose some regions of interest and put all together to make a single composite ROI and determine the mean value of FDG activity there. And you really want to make the same shit.

Ok, let's see how. First you need to know what regions you want. Take, for example the HO Atlas. If the desired composite ROI includes:

right and left angular gyri, bilateral posterior cingulate gyrus, left middle/inferior temporal gyrus and right middle/inferior temporal gyrus

then the following ROIs:

- Angular_Gyrus
- Cingulate_Gyrus_posterior_division
- Middle_Temporal_Gyrus_anterior_division
- Middle_Temporal_Gyrus_posterior_division
- Middle_Temporal_Gyrus_temporooccipital_part
- Inferior_Temporal_Gyrus_anterior_division
- Inferior_Temporal_Gyrus_posterior_division
- Inferior_Temporal_Gyrus_temporooccipital_part

must be extracted from HO atlas and joined into a composite single ROI.

Put your ROIs together into the same mask

Luckily the HO Atlas (see [Pipeline HO Atlas individual ROI files for MNI standard space](#)) is already separated into every single ROI at *pipeline* lib directory. Just make,

```
$ cd /opt/neuro.dev/lib/  
$ fslmaths oxford/Cortical_21.nii.gz -add oxford/Cortical_30.nii.gz -add  
oxford/Cortical_11.nii.gz -add oxford/Cortical_12.nii.gz -add  
oxford/Cortical_13.nii.gz -add oxford/Cortical_14.nii.gz -add  
oxford/Cortical_15.nii.gz -add oxford/Cortical_16.nii.gz wroi.nii.gz
```

in order to get you composite ROI.

Use it

At this point you need to have all your PET images corregistered into the MNI space ([aquí hay una pista](#)). Then get into working directory and make something like this:

```
$ for x in *_pet_inMNI*; do a=`echo $x | awk -F"_" '{print $1}' | sed
's/mci//;s/hc//`; b=`fslstats $x -k
/opt/neuro.dev/lib/mni_cerebellum.nii.gz -M`; c=`fslstats $x -k
/opt/neuro.dev/lib/wroi.nii.gz -M`; echo $a, `echo $c/$b | bc -l` ; done >
../wroi.csv
```

Voilà! You got it. 😊

What about batches?

```
$ for y in /opt/neuro.dev/lib/composites/*.nii.gz; do z=`basename $y
.nii.gz`; for x in *_pet_inMNI*; do a=`echo $x | awk -F"_" '{print $1}' | sed
's/mci//;s/hc//`; b=`fslstats $x -k
/opt/neuro.dev/lib/mni_cerebellum.nii.gz -M`; c=`fslstats $x -k $y -M`; echo
$a, `echo $c/$b | bc -l` ; done > ../${z}.csv; done
```

😊 good luck

Imaging problems huh

There are some images with incomplete scans. In particular some of those images have incomplete around cerebellum. What happen then if there is an image with half cerebellum? To avoid those problem I will assume that the mean value of the signal in the scanned portion of cerebellum is equal to the mean value in the whole cerebellum.

This way:

```
for y in /opt/neuro.dev/lib/composites/*.nii.gz; do z=`basename $y .nii.gz`;
for x in *_pet_inMNI*; do a=`echo $x | awk -F"_" '{print $1}' | sed
's/mci//;s/hc//`; fslmaths $x -thr 0 ${x%.nii.gz}_tmp; b=`fslstats
${x%.nii.gz}_tmp -k /opt/neuro.dev/lib/mni_cerebellum.nii.gz -M`;
c=`fslstats ${x%.nii.gz}_tmp -k /opt/neuro.dev/lib/composites/$z -M`; echo
$a, `echo $c/$b | bc -l`; done > ../${z}.csv; rm *_tmp*; done;
```

What about FS?

mmmm, se puede implementar pero hay que mirar esto:

- <http://surfer.nmr.mgh.harvard.edu/fswiki/FsAnat-to-NativeAnat>
- http://surfer.nmr.mgh.harvard.edu/fswiki/mri_label2vol
- <http://brainybehavior.com/neuroimaging/2010/05/converting-cortical-labels-from-freesurfer-to-v>

[olometric-masks/](#)

Primeramente hay que mirar como [aplicar los labels en FS](#).

The ADNI way (thanks to S. Landau)

⚠ Esto es muy parecido pero usando las zonas correctas. Ahi vamos:

```
for x in *_pet_inMNI*; do a=`echo $x | awk -F"_" '{print $1}' | sed
's/mci//;s/hc//'; b=`fslstats $x -k
/opt/neuro.dev/lib/bin_rpons_vermis.nii.gz -M`; c=`fslstats $x -k
/opt/neuro.dev/lib/ADNI_Composite.nii.gz -M`; echo $a, `echo $c/$b | bc -l`
; done > ../slandau.csv
```

From:
<http://mail.fundacioace.com/wiki/> - **Detritus Wiki**

Permanent link:
<http://mail.fundacioace.com/wiki/doku.php?id=neuroimagen:xrois>

Last update: **2020/08/04 10:58**

