

# Procesando fMRI con AFNI (single subject)

Voy a seguir los pasos de aqui,

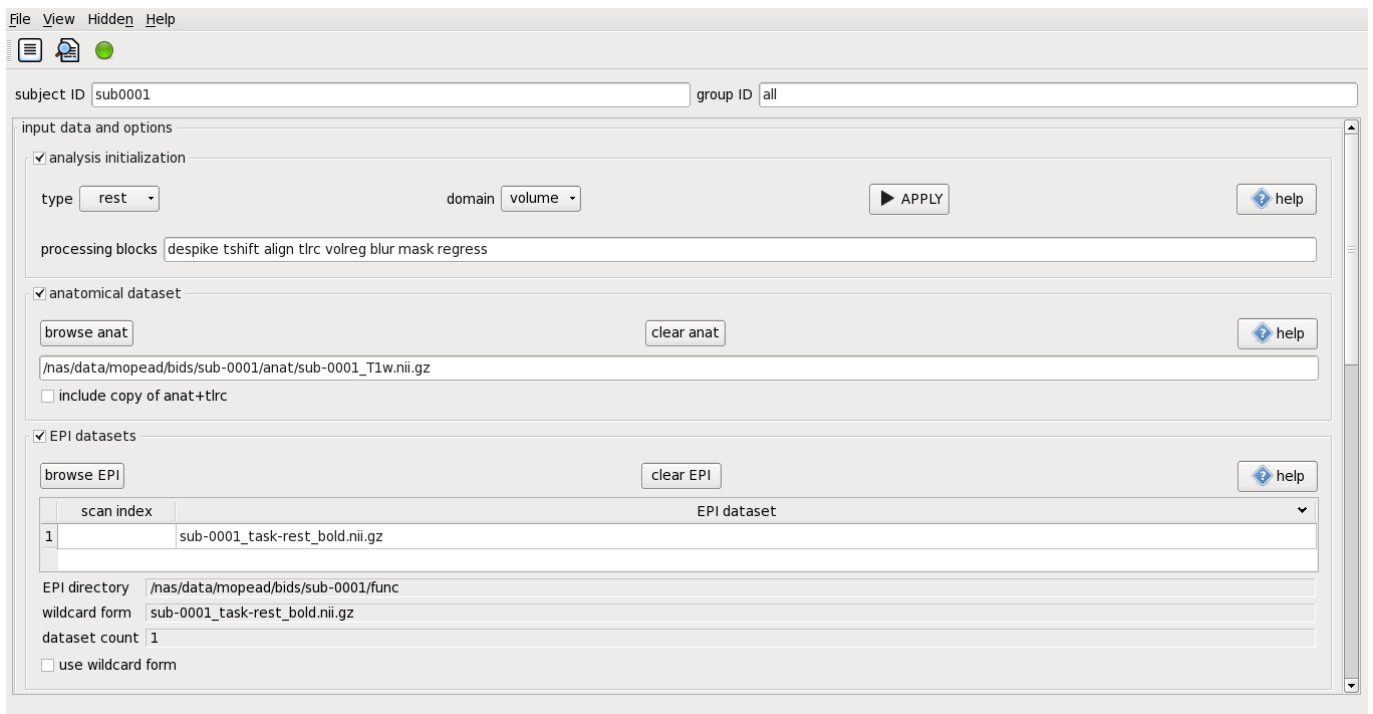
<https://blog.cogneurostats.com/2013/05/29/single-subject-analysis-in-afni/>

**Revisar aqui primero:**

[https://afni.nimh.nih.gov/pub/dist/doc/html/doc/programs/afni\\_proc.py\\_sphx.html](https://afni.nimh.nih.gov/pub/dist/doc/html/doc/programs/afni_proc.py_sphx.html)

## Construyendo el template

Siguiendo esta entrada del blog lanzo *uber\_subject.py* y lleno las opciones que deseo,



Pero no voy a correr el programa. En lugar de esto exporto los comandos que se correran,

```
#!/usr/bin/env tcsh

# created by uber_subject.py: version 0.36 (April 5, 2013)
# creation date: Fri Jan 24 16:33:30 2020

# set data directories
set top_dir = /nas/data/mopead/bids/sub-0001
set anat_dir = $top_dir/anat
set epi_dir = $top_dir/func

# set subject and group identifiers
set subj = sub_0001
set group_id = all
```

```
# run afni_proc.py to create a single subject processing script
afni_proc.py -subj_id $subj \
  -script proc.$subj -scr_overwrite \
  -blocks despiked tshift align tlrc volreg blur mask regress \
  -copy_anat $anat_dir/sub-0001_T1w.nii.gz \
  -tcats_remove_first_trs 0 \
  -dsets $epi_dir/sub-0001_task-rest_bold.nii.gz \
  -volreg_align_to third \
  -volreg_align_e2a \
  -volreg_tlrc_warp \
  -blur_size 4.0 \
  -regress_censor_motion 0.2 \
  -regress_censor_outliers 0.1 \
  -regress_bandpass 0.01 0.1 \
  -regress_apply_motion_types demean deriv \
  -regress_reml_exec \
  -regress_est_blur_errts
```

Este script lo voy a tomar de template para crear los scripts que necesite ejecutar.

### [afni\\_proc.template](#)

```
#!/usr/bin/env tcsh

# created by uber_subject.py: version 0.36 (April 5, 2013)
# creation date: Fri Jan 24 16:33:30 2020

# set data directories
set top_dir = /nas/data/<project>/bids/<subject>
set anat_dir = $top_dir/anat
set epi_dir = $top_dir/func

# set subject and group identifiers
set subj = <subject_mod>
set group_id = all

# run afni_proc.py to create a single subject processing script
afni_proc.py -subj_id $subj \
  -script proc.$subj -scr_overwrite \
  -blocks despiked tshift align tlrc volreg blur mask regress \
  -copy_anat $anat_dir/<subject>_T1w.nii.gz \
  -tcats_remove_first_trs 0 \
  -dsets $epi_dir/<subject>_task-rest_bold.nii.gz \
  -volreg_align_to third \
  -volreg_align_e2a \
  -volreg_tlrc_warp \
  -blur_size 4.0 \
  -regress_censor_motion 0.2 \
  -regress_censor_outliers 0.1 \
  -regress_bandpass 0.01 0.1
```

```
-regress_apply_mot_types demean deriv \
-regress_reml_exec \
-regress_est_blur_errts
```

Ojo que este script solo creara otro script, que es el que hay que ejecutar, el output de este escript es,

```
[osotolongo@detritus subj.sub_0001]$ ./cmd.ap.sub_0001
-- applying orig view as +orig
-- tcat: reps is now 200
-- volreg: base/sub indices are 0, 2
++ volreg: warp and align to isotropic 3 mm tlrc voxels
-- masking: group anat = 'TT_N27+tlrc', exists = 1
++ updating polort to 3, from run len 426.0 s
-- will apply motion types: demean, deriv
++ adding labels: []
-- will add 3dClustSim table to stats dset
-- using default: will not apply EPI Automask
  (see 'MASKING NOTE' from the -help for details)

-----
** warning have only 1 run to analyze
-----

--> script is file: proc.sub_0001
    consider the script execution command:

    tcsh -xef proc.sub_0001 |& tee output.proc.sub_0001
```

## Del template a los scripts

Primero escribimos el template con tags que sean simples de sustituir,

[afni\\_proc\\_rest.template](#)

```
#!/usr/bin/env tcsh

# created by uber_subject.py: version 0.36 (April 5, 2013)
# creation date: Fri Jan 24 16:33:30 2020

# set data directories
set top_dir    = /nas/data/<project>/bids/<subject>
set anat_dir   = $top_dir/anat
set epi_dir    = $top_dir/func

# set subject and group identifiers
set subj       = <subject_mod>
set group_id   = all
```

```
# run afni_proc.py to create a single subject processing script
afni_proc.py -subj_id $subj \
  -script proc.$subj -scr_overwrite \
  -blocks despiked tshift align tlrc volreg blur mask regress \
  -copy_anat <anat> \
  -tcats_remove_first_trs 0 \
  -dsets <bold_rest> \
  -volreg_align_to third \
  -volreg_align_e2a \
  -volreg_tlrc_warp \
  -blur_size 4.0 \
  -regress_bandpass 0.01 0.1 \
  -regress_apply_mot_types demean deriv \
  -regress_reml_exec \
  -regress_est_blur_errts
```

Ahora, para cada sujeto a analizar, debemos crear este script, ejecutarlo, tomar el script resultante y ejecutarlo, esto ultimo directamente en el directorio de output,

lo mas rapido es irlo haciendo on-the-fly,

[dfmriafni.pl](#)

```
#!/usr/bin/perl
use strict; use warnings;

use File::Find::Rule;
use NEUR04 qw(print_help load_project shit_done cut_shit check_subj
check_or_make);
use Data::Dump qw(dump);
use File::Remove 'remove';
use File::Basename qw(basename);
use File::Copy::Recursive qw(dircopy);

my $template = $ENV{'PIPEDIR'}.'./lib/afni_proc_rest.template';
my $cfile="";

@ARGV = ("-h") unless @ARGV;

while (@ARGV and $ARGV[0] =~ /^-/) {
  $_ = shift;
  last if /^--$/;
  if (/^-cut/) { $cfile = shift; chomp($cfile);}
  if (/^-h$/) { print_help $ENV{'PIPEDIR'}.'./doc/dfmriafni.hlp';
exit;}
}
my $study = shift;
unless ($study) { print_help $ENV{'PIPEDIR'}.'./doc/dfmriafni.hlp';
```

```

exit;}
my %std = load_project($study);
my $w_dir = $std{'WORKING'};
my $data_dir = $std{'DATA'};
my $bids_dir = $std{'BIDS'};
my $fsdir = $ENV{'SUBJECTS_DIR'};
my $db = $std{'DATA'}. '/' . $study . '_mri.csv';
my $fmriout_dir = $data_dir . '/afni_out';
check_or_make($fmriout_dir);
my $outdir = "$std{'DATA'}/slurm";
check_or_make($outdir);

my @subjects = cut_shit($db, $data_dir . "/" . $cfile);

foreach my $subject (@subjects) {
    my %nifti = check_subj($std{'DATA'}, $subject);
    if($nifti{'func'}){
        chdir($outdir);
        my $creator = $outdir . '/afni_proc_' . $subject . '.py';
        open TPF, "<$template" or die "No template file\n";
        open CRF, ">$creator" or die "Could not create
afni_proc\n";
        while(<TPF>){
            s/<project>/$study/;
            s/<subject>/sub-$subject/;
            s/<subject_mod>/sub_$subject/;
            s/<anat>/$nifti{'T1w'}/;
            s/<bold_rest>/$nifti{'func'}/;
            print CRF;
            print;
        }
        close CRF;
        close TPF;
        chmod 0755, $creator;
        system($creator);
        my $orderfile = $outdir . '/' . $subject . '_fmriafni.sh';
        open ORD, ">$orderfile";
        print ORD '#!/bin/bash' . "\n";
        print ORD '#SBATCH -J fmriafni_' . $study . "\n";
        print ORD '#SBATCH --time=72:0:0' . "\n"; #si no ha
terminado en X horas mata lo
        print ORD '#SBATCH --mail-
type=FAIL,TIME_LIMIT,STAGE_OUT' . "\n"; #no quieres que te mande email de
todo
        print ORD '#SBATCH -o ' . $outdir . '/fmriafni-%j' . "\n";
        print ORD '#SBATCH -c 16' . "\n";
        print ORD '#SBATCH -p fast' . "\n";
        print ORD '#SBATCH --mail-user=' . $ENV{'USER'} . "\n";
        print ORD 'srun ' . $outdir . '/proc.sub_' . $subject;
        close ORD;
        chdir($fmriout_dir);
    }
}

```

```
        system("sbatch $orderfile");
    }
}
my $orderfile = $outdir.'/fmriafni_end.sh';
open ORD, ">$orderfile";
print ORD '#!/bin/bash'. "\n";
print ORD '#SBATCH -J fmriafni_'. $study. "\n";
print ORD '#SBATCH --mail-type=END'. "\n"; #email cuando termine
print ORD '#SBATCH --mail-user='.$ENV{'USER'} "\n";
print ORD '#SBATCH -p fast'. "\n";
print ORD '#SBATCH -o '.$outdir.'/fmriprep_end-%j'. "\n";
print ORD ":\n";
close ORD;
my $order = 'sbatch --dependency=singleton '.$orderfile;
exec($order);
```

Es bastante rápido y deja el output en *afni\_out*,

```
[osotolongo@detritus mopead]$ ls afni_out/
sub_0001.results  sub_0014.results  sub_0029.results  sub_0042.results
sub_0056.results  sub_0069.results  sub_0083.results  sub_0096.results
sub_0109.results  sub_0122.results
sub_0002.results  sub_0015.results  sub_0030.results  sub_0043.results
sub_0057.results  sub_0070.results  sub_0084.results  sub_0097.results
sub_0110.results  sub_0123.results
sub_0003.results  sub_0016.results  sub_0031.results  sub_0044.results
sub_0058.results  sub_0071.results  sub_0085.results  sub_0098.results
sub_0111.results  sub_0124.results
sub_0004.results  sub_0017.results  sub_0032.results  sub_0045.results
sub_0059.results  sub_0072.results  sub_0086.results  sub_0099.results
sub_0112.results  sub_0125.results
sub_0005.results  sub_0019.results  sub_0033.results  sub_0046.results
sub_0060.results  sub_0073.results  sub_0087.results  sub_0100.results
sub_0113.results  sub_0126.results
sub_0006.results  sub_0020.results  sub_0034.results  sub_0047.results
sub_0061.results  sub_0074.results  sub_0088.results  sub_0101.results
sub_0114.results
sub_0007.results  sub_0021.results  sub_0035.results  sub_0048.results
sub_0062.results  sub_0075.results  sub_0089.results  sub_0102.results
sub_0115.results
sub_0008.results  sub_0022.results  sub_0036.results  sub_0049.results
sub_0063.results  sub_0076.results  sub_0090.results  sub_0103.results
sub_0116.results
sub_0009.results  sub_0023.results  sub_0037.results  sub_0050.results
sub_0064.results  sub_0077.results  sub_0091.results  sub_0104.results
sub_0117.results
sub_0010.results  sub_0024.results  sub_0038.results  sub_0051.results
sub_0065.results  sub_0078.results  sub_0092.results  sub_0105.results
sub_0118.results
```

```

sub_0011.results  sub_0025.results  sub_0039.results  sub_0052.results
sub_0066.results  sub_0079.results  sub_0093.results  sub_0106.results
sub_0119.results
sub_0012.results  sub_0026.results  sub_0040.results  sub_0053.results
sub_0067.results  sub_0081.results  sub_0094.results  sub_0107.results
sub_0120.results
sub_0013.results  sub_0028.results  sub_0041.results  sub_0055.results
sub_0068.results  sub_0082.results  sub_0095.results  sub_0108.results
sub_0121.results
[osotolongo@detritus mopead]$ ls afni_out/sub_0001.results/
all_runs.sub_0001+tlrc.BRIK      @epi_review.sub_0001
mask_group+tlrc.HEAD            pb02.sub_0001.r01.tshift+orig.BRIK
sub-0001_T1w_al_junk+orig.BRIK
all_runs.sub_0001+tlrc.HEAD      errts.sub_0001_REML+tlrc.BRIK
mat.r01.vr.aff12.1D              pb02.sub_0001.r01.tshift+orig.HEAD
sub-0001_T1w_al_junk+orig.HEAD
anat_final.sub_0001+tlrc.BRIK    errts.sub_0001_REML+tlrc.HEAD
mat.r01.warp.aff12.1D            pb03.sub_0001.r01.volreg+tlrc.BRIK
sub-0001_T1w_ns+orig.BRIK
anat_final.sub_0001+tlrc.HEAD    errts.sub_0001+tlrc.BRIK
motion_demean.1D                 pb03.sub_0001.r01.volreg+tlrc.HEAD
sub-0001_T1w_ns+orig.HEAD
bandpass_rall.1D                  errts.sub_0001+tlrc.HEAD
motion_deriv.1D                   pb04.sub_0001.r01.blur+tlrc.BRIK
sub-0001_T1w_ns+tlrc.BRIK
blur.err_reml.1D                  fitts.sub_0001_REML+tlrc.BRIK
motion_sub_0001_enorm.1D          pb04.sub_0001.r01.blur+tlrc.HEAD
sub-0001_T1w_ns+tlrc.HEAD
blur.errts.1D                     fitts.sub_0001_REML+tlrc.HEAD
out.cormat_warn.txt               @ss_review_basic
sub-0001_T1w_ns_WarpDrive.log
blur_est.sub_0001.1D              fitts.sub_0001+tlrc.BRIK
outcount.r01.1D                   @ss_review_driver
sub-0001_T1w_ns.Xaff12.1D
ClustSim.mask                     fitts.sub_0001+tlrc.HEAD
outcount_rall.1D                  @ss_review_driver_commands
sub-0001_T1w_ns.Xat.1D
ClustSim.NN1.1D                   full_mask.sub_0001+tlrc.BRIK   out.gcor.1D
stats.REML_cmd                    sub-0001_T1w+orig.BRIK
ClustSim.NN1.niml                  full_mask.sub_0001+tlrc.HEAD
out.mask_overlap.txt              stats.sub_0001_REMLvar+tlrc.BRIK
sub-0001_T1w+orig.HEAD
ClustSim.NN2.1D                   gmean.errts.unit.1D
out.pre_ss_warn.txt               stats.sub_0001_REMLvar+tlrc.HEAD
sum_ideal.1D
ClustSim.NN2.niml                  mask_anat.sub_0001+tlrc.BRIK
out.ss_review.sub_0001.txt         stats.sub_0001+tlrc.BRIK
TSNR.sub_0001+tlrc.BRIK
ClustSim.NN3.1D                   mask_anat.sub_0001+tlrc.HEAD
pb00.sub_0001.r01.tcat+orig.BRIK  stats.sub_0001+tlrc.HEAD
TSNR.sub_0001+tlrc.HEAD

```

```
ClustSim.NN3.niml          mask_epi_extents+tlrc.BRIK
pb00.sub_0001.r01.tcat+orig.HEAD  stimuli
X.jpg
dfile.r01.1D              mask_epi_extents+tlrc.HEAD
pb01.sub_0001.r01.despike+orig.BRIK
sub-0001_T1w_al_junk_e2a_only_mat.aff12.1D  X.stim.xmat.1D
dfile_rall.1D            mask_group+tlrc.BRIK
pb01.sub_0001.r01.despike+orig.HEAD  sub-0001_T1w_al_junk_mat.aff12.1D
X.xmat.1D
```

## Resultados

Revisar esto visualmente con AFNI tiene truco,

```
[osotolongo@detritus mopead]$ cd afni_out/sub_0001.results/
[osotolongo@detritus sub_0001.results]$ afni -dset all_runs.sub_0001+tlrc
```

Thanks go to J Evans for useful feedback

Initializing: X11.++ AFNI is detached from terminal.

```
[osotolongo@detritus sub_0001.results]$ . Widgets..... Input files:
dataset count = 1
Time series    = 22 files read
NLfit & NLerr= Optimizer (AFNI_NLFIM_METHOD) is SIMPLEX
NLfit & NLerr= Found 26 models
Plugins        = 52 libraries read
```

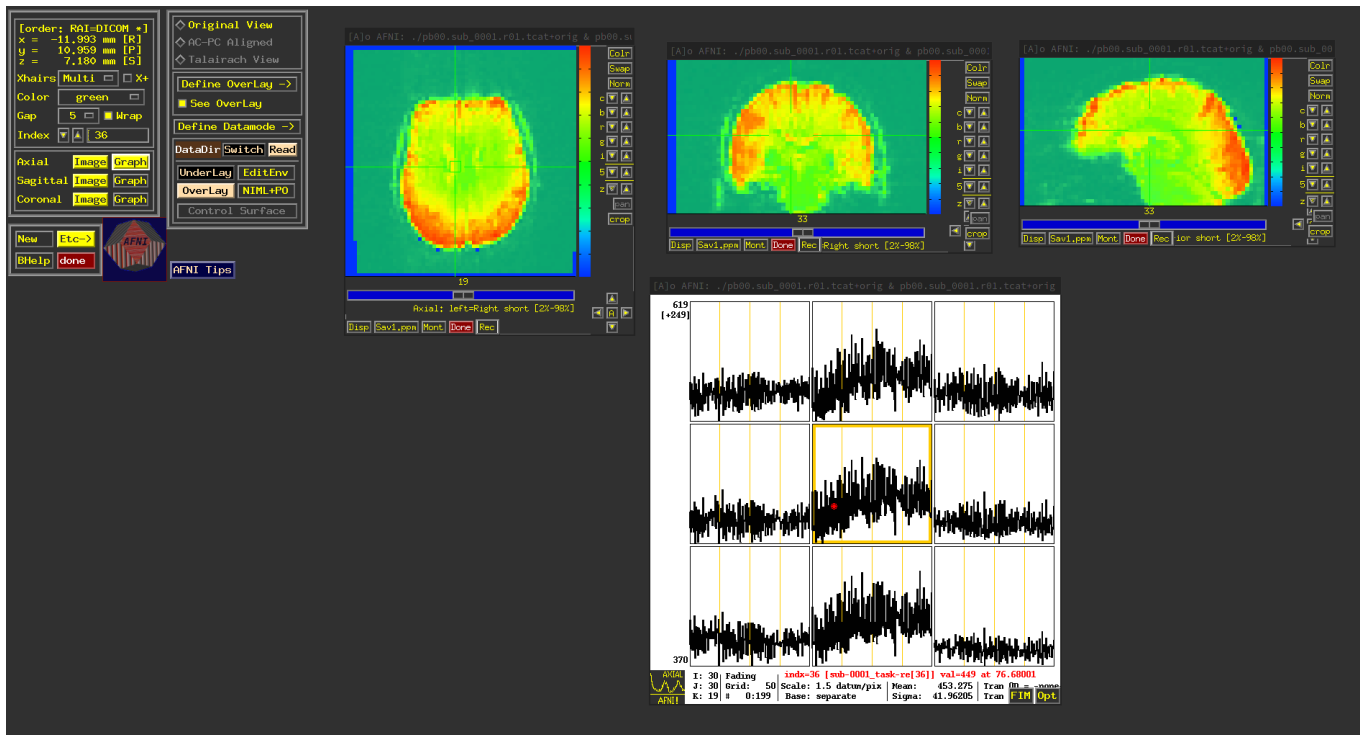
++ NOTE: This version of AFNI was built Sep 30 2013 ++

++ NOTE: 'Define Markers' is hidden: right-click 'DataDir' to see it

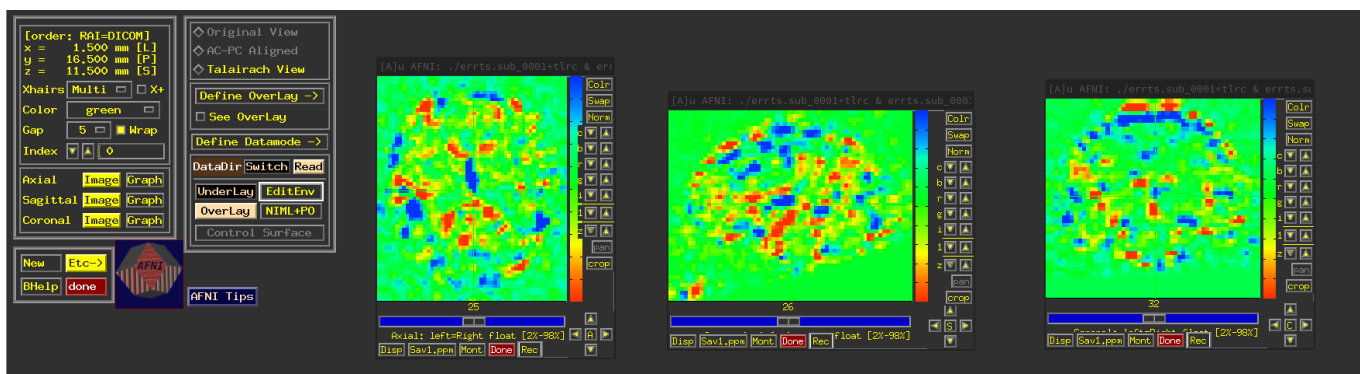
++ NOTE: you may want to consider creating a '.afnirc' file in your home directory, to control AFNI's setup. For more details, see

[http://afni.nimh.nih.gov/pub/dist/doc/program\\_help/README.environment.html](http://afni.nimh.nih.gov/pub/dist/doc/program_help/README.environment.html)





nota: Necesito otro monitor mas!

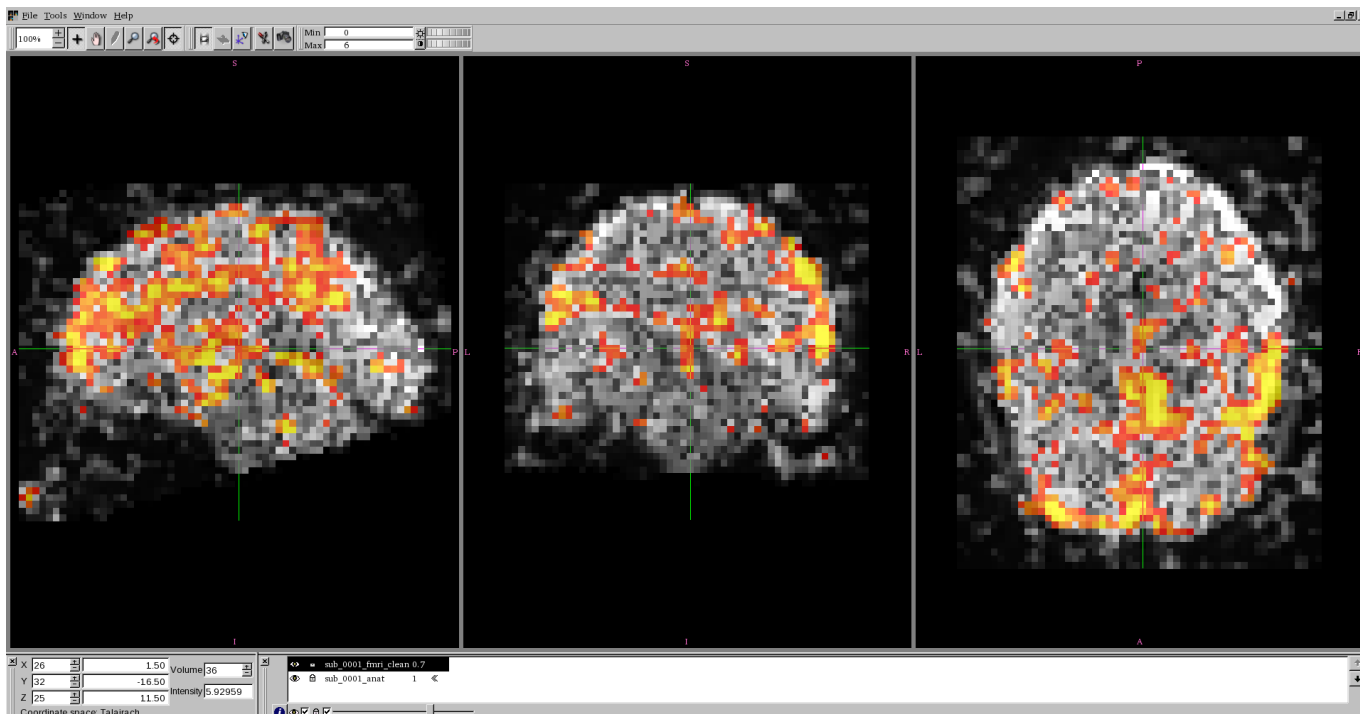


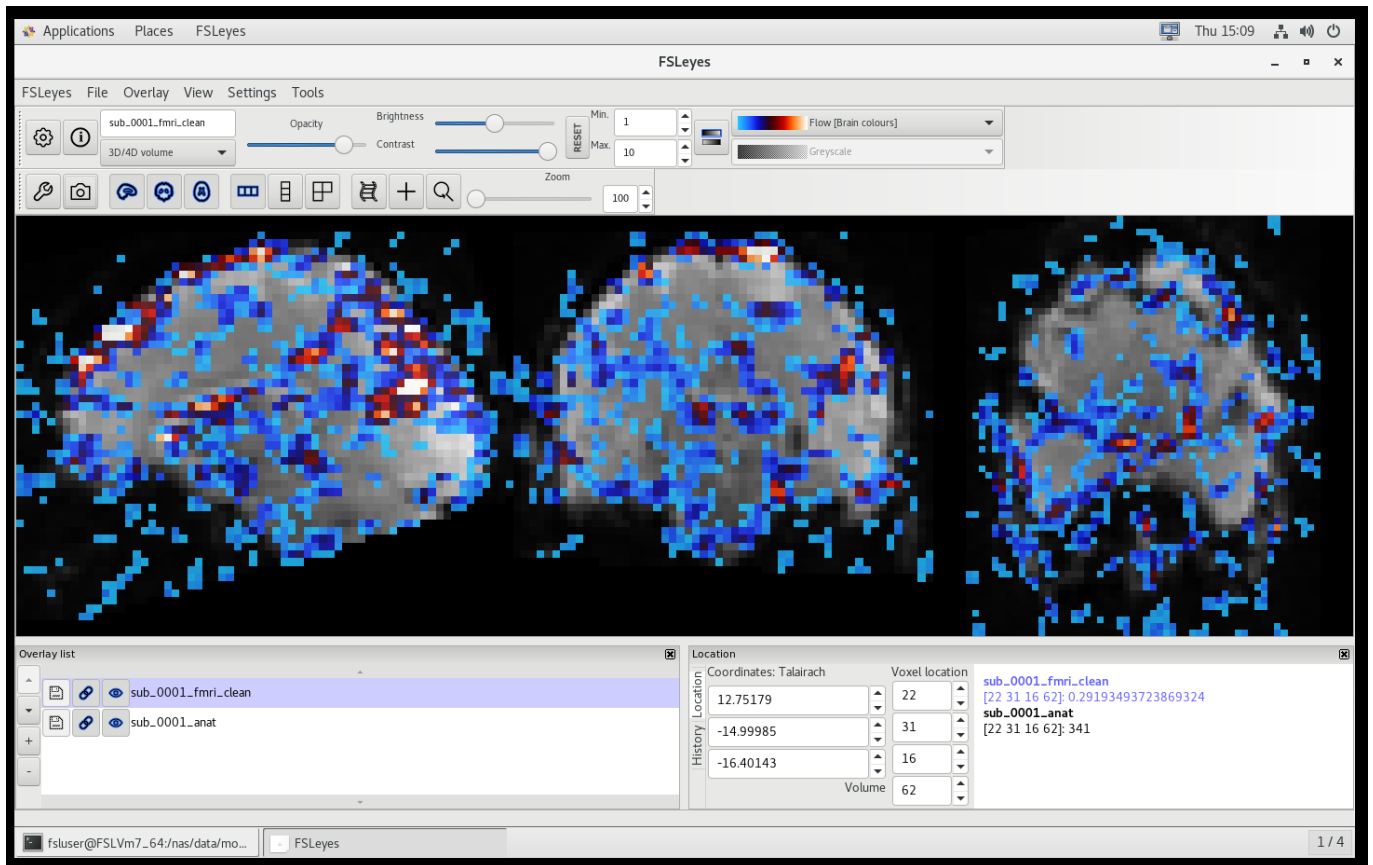
[https://afni.nimh.nih.gov/pub/dist/doc/html/doc/programs/afni\\_proc.py\\_sphx.html#main-outputs-many-datasets-are-created](https://afni.nimh.nih.gov/pub/dist/doc/html/doc/programs/afni_proc.py_sphx.html#main-outputs-many-datasets-are-created)

Para el *resting state*, el archivo de output con los datos limpios es *errts+sub\_XXXX+tlrc*. Puedo convertirlo a formato NIFTI facilmente,

```
[osotolongo@detritus sub_0001.results]$ 3dAFNItoNIFTI -prefix
sub_0001_fmri_clean errts.sub_0001+tlrc
++ 3dAFNItoNIFTI: AFNI version=AFNI_2011_12_21_1014 (Sep 30 2013) [64-bit]
[osotolongo@detritus sub_0001.results]$ fslinfo sub_0001_fmri_clean.nii
data_type      FLOAT32
dim1           54
dim2           64
dim3           50
dim4           200
datatype       16
pixdim1        3.000000
pixdim2        3.000000
```

```
pixdim3      3.000000
pixdim4      2.130000
cal_max      0.0000
cal_min      0.0000
file_type    NIFTI-1+
[osotolongo@detritus sub_0001.results]$ 3dAFNItoNIFTI -prefix sub_0001_anat
pb03.sub_0001.r01.volreg+tlrc
++ 3dAFNItoNIFTI: AFNI version=AFNI_2011_12_21_1014 (Sep 30 2013) [64-bit]
[osotolongo@detritus sub_0001.results]$ fslinfo sub_0001_anat.nii
data_type    INT16
dim1         54
dim2         64
dim3         50
dim4         200
datatype     4
pixdim1      3.000000
pixdim2      3.000000
pixdim3      3.000000
pixdim4      2.130000
cal_max      0.0000
cal_min      0.0000
file_type    NIFTI-1+
[osotolongo@detritus sub_0001.results]$ fslview_deprecated sub_0001_anat.nii
sub_0001_fmri_clean.nii -l "render1" -b 0,6 -t 0.7
```





[recording.mp4](#)

## y ahora?

Postprocessing,

- <https://blog.cogneurostats.com/2013/04/09/using-r-with-afni/>
- <https://blog.cogneurostats.com/2014/03/25/r-for-mri-lovers-part-1/>
- <https://blog.cogneurostats.com/2014/09/03/r-for-mri-lovers-part-2/>
- <https://blog.cogneurostats.com/2013/07/24/creating-volume-rois-in-afni/>

From:

<https://cortafuegos.fundacioace.com/wiki/> - **Detritus Wiki**

Permanent link:

[https://cortafuegos.fundacioace.com/wiki/doku.php?id=neuroimagen:afni\\_proc](https://cortafuegos.fundacioace.com/wiki/doku.php?id=neuroimagen:afni_proc)

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

