

Plantilla y datos para VBM

Tengo muchos problemas para montar un VBM con las imagenes 3T, en particular con el protocolo BIOFACE. No soy capaz de registrarlas con FSL, asi que el FSLVBM no funciona. He intentado algun *hack* pero el registro es tan malo que no hay manera.

Así que me voy a montar un VBM a medida usando [ANTs](#), que es lo que mejor me funciona para corregistro. Ver:

1. <https://github.com/asqwerty666/acevbm>
2. <https://gsanroma.github.io/posts/2019/01/vbm-data-preparation/>
3. https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FSLVBM/UserGuide#E_-_Obtaining_and_displaying_your_FSL-VBM_results

 Merge: Custom VBM pipeline 

[get_fsaseg.sh](#)

```
#!/bin/sh

fsid=$1
shift

id=$1
shift

dir=$1
shift

debug=0

#First get the freesurfer processed aseg.mgz
${FREESURFER_HOME}/bin/mri_label2vol --seg
${SUBJECTS_DIR}/${fsid}/mri/aseg.mgz --temp
${SUBJECTS_DIR}/${fsid}/mri/rawavg.mgz --o ${dir}/${id}_tmp_aseg.mgz --
regheader ${SUBJECTS_DIR}/${fsid}/mri/aseg.mgz
${FREESURFER_HOME}/bin/mri_convert --in_type mgz --out_type nii
${dir}/${id}_tmp_aseg.mgz ${dir}/${id}_tmp_aseg.nii.gz
${FSLDIR}/bin/fslreorient2std ${dir}/${id}_tmp_aseg ${dir}/${id}_aseg

if [ $debug = 0 ] ; then
    rm ${dir}/${id}_tmp*
fi
```

[get_lut.sh](#)

```
#!/bin/sh
```

```
id=$1
shift

dir=$1
shift

lut=$1
shift

debug=0

${FSLDIR}/bin/fslmaths ${dir}/${id}_aseg.nii.gz -uthr ${lut} -thr
${lut} -div ${lut} ${dir}/${id}_${lut}.nii.gz
```

mktpl.pl

```
#!/usr/bin/perl

# Copyright 2021 O. Sotolongo <asqwerty@gmail.com>

use strict; use warnings;
use File::Temp qw(tempdir);
use File::Find::Rule;
use Cwd qw(getcwd);
use SLURM qw(send2slurm);
use File::Basename qw(basename);
use Data::Dump qw(dump);
my $odir;
my $ilist;
@ARGV = ("-h") unless @ARGV;
while (@ARGV and $ARGV[0] =~ /^-/) {
    $_ = shift;
    last if /^--$/;
    if (/^-o/) {$odir = shift; chomp $odir;}
    if (/^-i/) {$ilist = shift; chomp $ilist;}
}

die "Should supply output directory\n" unless $odir;
die "Should supply file list\n" unless $ilist;

my @gmluts = (3, 10, 11, 12, 13, 17, 18, 26, 42, 49, 50, 51, 52, 53,
54, 58);
my @wmluts = (2, 16, 28, 41, 60, 77, 85, 251, 252, 253, 254, 255);
my %subjects;
my $cwdir = getcwd();
my $wdir = $cwdir.'/working';
my $slurm_dir = $cwdir.'/slurm';
unless (-d $wdir) {mkdir $wdir;}
unless (-d $slurm_dir) {mkdir $slurm_dir;}
my %tplrun;
```

```

my %r_jobs;
$tplrun{'job_name'} = 'reg2tpl';
$tplrun{'cpus'} = 4;
$tplrun{'time'} = '4:0:0';
open IDF, "<$ilist" or die "Could not open file\n";
while (<IDF>) {
    my ($sid, $fsid) = /(.*),(.*)/;
    $subjects{$sid} = $fsid;
    my $tdir = tempdir( CLEANUP => 1);
    $tplrun{'filename'} = $slurm_dir.'/'.$sid.'_prepare2tpl.sh';
    $tplrun{'output'} = $slurm_dir.'/prep2tpl_output_'.$sid;
    $tplrun{'command'} = $ENV{'PIPEDIR'}.'/bin/get_fsaseg.sh
'.'. $fsid.' '.'. $sid.' '.'. $tdir." \n";
    #my $order = $ENV{'PIPEDIR'}.'/bin/get_fsaseg.sh '.'. $fsid.'
'.'. $sid.' '.'. $tdir;
    #system($order);
    my $imlist = $ENV{'FSLDIR'}.'/bin/fslmaths ';
    my $first = 1;
    foreach my $roi (@gmluts){
        $imlist .= ($first?' ':' -add
'.'. $tdir.'/'.$sid.'_'.$roi.'.nii.gz';
        $first = 0;
        $tplrun{'command'} .= $ENV{'PIPEDIR'}.'/bin/get_lut.sh
'.'. $sid.' '.'. $tdir.' '.'. $roi." \n";
        #$order = $ENV{'PIPEDIR'}.'/bin/get_lut.sh '.'. $sid.'
'.'. $tdir.' '.'. $roi;
        #system($order);
    }
    $tplrun{'command'} .= $imlist.'
'.'. $wdir.'/'.$sid.'_GM.nii.gz'." \n";
    #$order = $imlist.' '.'. $wdir.'/'.$sid.'_GM.nii.gz';
    #print "$order\n";
    #system($order);
    $imlist = $ENV{'FSLDIR'}.'/bin/fslmaths ';
    $first = 1;
    foreach my $roi (@wmluts){
        $imlist .= ($first?' ':' -add
'.'. $tdir.'/'.$sid.'_'.$roi.'.nii.gz';
        $first = 0;
        $tplrun{'command'} .= $ENV{'PIPEDIR'}.'/bin/get_lut.sh
'.'. $sid.' '.'. $tdir.' '.'. $roi." \n";
        #$order = $ENV{'PIPEDIR'}.'/bin/get_lut.sh '.'. $sid.'
'.'. $tdir.' '.'. $roi;
        #system($order);
    }
    $tplrun{'command'} .= $imlist.'
'.'. $wdir.'/'.$sid.'_WM.nii.gz'." \n";
    #$order = $imlist.' '.'. $wdir.'/'.$sid.'_WM.nii.gz';
    #system($order);
    my $job_id = send2slurm(\%tplrun);
    $r_jobs{$sid} = $job_id;

```

```

}
close IDF;
my $tmp = tempdir( CLEANUP => 1);
my @twm; my @tgm;
my @t_jobs;
my $stoprint = '';
my $seg_file = 'seg_files.csv';
my $flist = $wdir.'/'. $seg_file;
open TPF,">$flist" or die "$!\n";
foreach my $ssid (sort keys %subjects){
    $tplrun{'filename'} = $slurm_dir.'/'. $ssid.'_prepare2tpl.sh';
    $tplrun{'output'} = $slurm_dir.'/prep2tpl_output_'. $ssid;
    $tplrun{'dependency'} = 'afterok:'. $r_jobs{$ssid};
    $tplrun{'command'} =
$ENV{'ANTS_PATH'}.'/antsRegistrationSyNQuick.sh -d 3 -f
'.$ENV{'FSLDIR'}.'/data/standard/MNI152_T1_2mm.nii.gz -m
'.$wdir.'/'. $ssid.'_GM.nii.gz -t a -o '.$tmp.'/'. $ssid.'_GM_init_'. "\n";
    #system($order);
    $tplrun{'command'} .= $ENV{'ANTS_PATH'}.'/antsApplyTransforms -
d 3 -r '.$ENV{'FSLDIR'}.'/data/standard/MNI152_T1_2mm.nii.gz -i
'.$wdir.'/'. $ssid.'_GM.nii.gz -t
'.$tmp.'/'. $ssid.'_GM_init_0GenericAffine.mat -o
'.$tmp.'/'. $ssid.'_GM0.nii.gz'. "\n";
    #system($order);
    push @tgm, $tmp.'/'. $ssid.'_GM0.nii.gz';
    $tplrun{'command'} .=
$ENV{'ANTS_PATH'}.'/antsRegistrationSyNQuick.sh -d 3 -f
'.$ENV{'FSLDIR'}.'/data/standard/MNI152_T1_2mm.nii.gz -m
'.$wdir.'/'. $ssid.'_WM.nii.gz -t a -o '.$tmp.'/'. $ssid.'_WM_init_'. "\n";
    #system($order);
    $tplrun{'command'} .= $ENV{'ANTS_PATH'}.'/antsApplyTransforms -
d 3 -r '.$ENV{'FSLDIR'}.'/data/standard/MNI152_T1_2mm.nii.gz -i
'.$wdir.'/'. $ssid.'_WM.nii.gz -t
'.$tmp.'/'. $ssid.'_WM_init_0GenericAffine.mat -o
'.$tmp.'/'. $ssid.'_WM0.nii.gz'. "\n";
    #system($order);
    push @twm, $tmp.'/'. $ssid.'_WM0.nii.gz';
    $stoprint .=
$wdir.'/'. $ssid.'_GM.nii.gz','.$wdir.'/'. $ssid.'_WM.nii.gz'. "\n";
    my $job_id = send2slurm(\%tplrun);
    push @t_jobs, $job_id;
}
my %tplmk;
$tplmk{'job_name'} = 'merge_all';
$tplmk{'cpus'} = 4;
$tplmk{'time'} = '4:0:0';
$tplmk{'dependency'} = 'afterok:'. join(', ', @t_jobs);
$tplmk{'output'} = $slurm_dir.'/merge2tpl_output';
$tplmk{'filename'} = $slurm_dir.'/merge2tpl.sh';
my $aux = join ' ', @tgm;

```

```

$tplmk{'command'} = $ENV{'FSLDIR'}.'/bin/fslmerge -t
'.$tmp.'/GM0template.nii.gz '$aux.'"\\n";
#system($order);
$tplmk{'command'} .= $ENV{'FSLDIR'}.'/bin/fslmaths
'.$tmp.'/GM0template.nii.gz '$tmp.'/avg_GM.nii.gz'.'"\\n";
#system($order);
$aux = join ' ',@twm;
$tplmk{'command'} .= $ENV{'FSLDIR'}.'/bin/fslmerge -t
'.$tmp.'/WM0template.nii.gz '$aux.'"\\n";
#system($order);
$tplmk{'command'} .= $ENV{'FSLDIR'}.'/bin/fslmaths
'.$tmp.'/WM0template.nii.gz '$tmp.'/avg_WM.nii.gz';
#system($order);
my $mjob = send2slurm(\%tplmk);
open TPF,">$flist" or die "$!\n";
    print TPF $toprint;
close TPF;
my $order = $ENV{'ANTS_PATH'}.'/waitForSlurmJobs.pl 0 600 '$mjob;
system($order);

$order = 'cd
'.$wdir.'; '$ENV{'ANTS_PATH'}.'/antsMultivariateTemplateConstruction2.s
h -d 3 -a 0 -b 0 -c 5 -u 1:0:0 -e 1 -g 0.25 -i 4 -k 2 -w 1x1 -q
70x50x30x10 -f 6x4x2x1 -s 3x2x1x0 -n 0 -o antsTPL_ -r 0 -l 1 -m CC -t
SyN -y 0 -z '$tmp.'/avg_GM.nii.gz -z '$tmp.'/avg_WM.nii.gz
'.$seg_file;
print "$order\\n";
system($order);
foreach my $sid (sort keys %subjects){
    my $order = $ENV{'ANTS_PATH'}.'/antsApplyTransforms -d 3 -i
'.$wdir.'/'.$sid.'_GM.nii.gz -r '$wdir.'/antsTPL_template0.nii.gz -o
'.$wdir.'/'.$sid.'_fulltransf.nii.gz -t
'.$wdir.'/antsTPL_'.$sid.'_GM*1Warp.nii.gz -t
'.$wdir.'/antsTPL_'.$sid.'_GM*0GenericAffine.mat --float || true';
    print "$order\\n";
    system($order);
    $order = $ENV{'ANTS_PATH'}.'/CreateJacobianDeterminantImage 3
'.$wdir.'/'.$sid.'_fulltransf.nii.gz '$wdir.'/'.$sid.'_jacobian.nii.gz
0 1 || true';
    print "$order\\n";
    system($order);
    $order = $ENV{'FSLDIR'}.'/bin/fslmaths
'.$wdir.'/'.$sid.'_fulltransf.nii.gz -mul
'.$wdir.'/'.$sid.'_jacobian.nii.gz '$wdir.'/'.$sid.'_GM2temp_mod';
    print "$order\\n";
    system($order);
}
my @regoks = find(file => 'name' => "*_fulltransf.nii.gz", in =>
$wdir);
@regoks = sort @regoks;
my @fsums;

```

```
my $nomodsums = join ' ', @regoks;
(my $modsums = $nomodsums) =~ s/fulltransf/GM2temp_mod/g;
my $statsdir = $cwdir.'/stats';
unless (-d $statsdir) {mkdir $statsdir;}
$order = $ENV{'FSLDIR'}.'/bin/fslmerge -t '.$statsdir.'/GM_merg
'.$nomodsums;
print "$order\n";
system($order);
$order = $ENV{'FSLDIR'}.'/bin/fslmerge -t '.$statsdir.'/GM_mod_merg
'.$modsums;
print "$order\n";
system($order);
open ROF, ">niceregister.list";
foreach my $regok (@regoks){
    $regok = basename $regok;
    $regok =~ s/(.*)_.*/$1/;
    print ROF "$regok\n";
}
close ROF;
```

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

Permanent link: https://mail.fundacioace.com/wiki/doku.php?id=neuroimagen:vbm_ants

Last update: **2021/05/05 09:21**

