

TRACULA, integrando en el cluster

<https://surfer.nmr.mgh.harvard.edu/fswiki/Tracula>

<https://surfer.nmr.mgh.harvard.edu/fswiki/trac-all>

dmrirc

El primer paso ha de ser escribir el archivo de configuracion adecuado. Ha de tener una forma como esta,

```
[tester@detritus fstest]$ cat dmri.rc
set subjlist = ( fstest_0001 fstest_0027 fstest_0106 fstest_0119
fstest_0187 )
set dcmlist = ( /nas/data/fstest/bids/sub-0001/dwi/sub-0001_dwi.nii.gz
/nas/data/fstest/bids/sub-0027/dwi/sub-0027_dwi.nii.gz
/nas/data/fstest/bids/sub-0106/dwi/sub-0106_dwi.nii.gz
/nas/data/fstest/bids/sub-0119/dwi/sub-0119_dwi.nii.gz
/nas/data/fstest/bids/sub-0187/dwi/sub-0187_dwi.nii.gz )
set bveclist = ( /nas/data/fstest/bids/sub-0001/dwi/sub-0001_dwi.bvec
/nas/data/fstest/bids/sub-0027/dwi/sub-0027_dwi.bvec
/nas/data/fstest/bids/sub-0106/dwi/sub-0106_dwi.bvec
/nas/data/fstest/bids/sub-0119/dwi/sub-0119_dwi.bvec
/nas/data/fstest/bids/sub-0187/dwi/sub-0187_dwi.bvec )
set bvallist = ( /nas/data/fstest/bids/sub-0001/dwi/sub-0001_dwi.bval
/nas/data/fstest/bids/sub-0027/dwi/sub-0027_dwi.bval
/nas/data/fstest/bids/sub-0106/dwi/sub-0106_dwi.bval
/nas/data/fstest/bids/sub-0119/dwi/sub-0119_dwi.bval
/nas/data/fstest/bids/sub-0187/dwi/sub-0187_dwi.bval )
```

Teniendo la lista de sujetos y los archivos en formato **BIDS** es bastante simple escribir este archivo.

Update: Asi que construimos un script que lo haga, asi como de que verifique y/o ejecute las segmentacion del *thalamus* ([ctrac_dmri.pl](#))

[Ejemplo de dmri.rc](#)

[Escribir el archivo es bastante simple](#)

```
my @jobs;
my $dob_line = 'set dob0 = 2';
my $pedir_line = 'AP PA ';
my @subjects = cut_shit($db, $data_dir."/".$cfile);
my $dmrirc = $data_dir.'/dmri.rc';
my $b0vec = $ENV{'PIPEDIR'}.'/lib/b0.bvec';
my $b0val = $ENV{'PIPEDIR'}.'/lib/b0.bval';
unless (-e $dmrirc && -r $dmrirc){
    my $subjlist = 'set subjlist = ( ';
```

```

my $dmclist = 'set dcmlist = ( ';
my $bveclist = 'set bveclist = ( ';
my $bavllist = 'set bvallist = ( ';;
my $pedir = 'set pedir = ( ';
foreach my $subject (@subjects) {
    my %nifti = check_subj($std{'DATA'},$subject);
    if($nifti{'dwi'} and $nifti{'dwi_sbref'}){
        $subjlist.=$study.'_'. $subject.'
'.'. $study.'_'. $subject.' ' ';
        $dmclist.=$nifti{'dwi'}.' ' '. $nifti{'dwi_sbref'}.' ' ' ';
        (my $bvec = $nifti{'dwi'}) =~ s/nii\.gz$/bvec/;
        (my $bval = $nifti{'dwi'}) =~ s/nii\.gz$/bval/;
        $bveclist.=$bvec.' ' '. $b0vec.' ' ';
        $bavllist.=$bval.' ' '. $b0val.' ' ';
        $pedir.=$pedir_line;
    }
}
$subjlist.=')';
$dmclist.=')';
$bveclist.=')';
$bavllist.=')';
$pedir.=')';
open CIF, ">$dmrirc" or die "Couldnt open dmrirc file for
writing\n";
print CIF
"$dob_line\n$subjlist\n$dmclist\n$bveclist\n$bavllist\n$pedir\n";
print CIF 'set echospacing = '."$echospacing\n". 'set epifactor =
'."$epifactor\n";
close CIF;
}else{
    die "There is a previous dmrirc file\nYou should delete it and try
again\n";
}

```

Pero hay mas,

Al incorporar la posibilidad de ejecutar *topup*, ahora tenemos que suministrar ls *bvec* y *bval* no solo de nuestro DTI (sentido AP), sino del B0 que se usa para corregir (sentido PA). En el caso del DTI completo, estos se crean a partir del DICOM. En el caso del B0 individual, este es facil de construir. Los he puesto como plantillas en el pipeline y lo que hago es añadirlos en cada caso.

b0.bvec

```
0 0 0
```

b0.bval

```
0
```

Y todavía hay mas,

El nuevo TRACULA busca la segmentacion del Thalamus dentro del directorio de Freesurfer. Asi que para cada sujeto, he de verificar que exista el archivo *mri/ThalamicNuclei.v12.T1.FSvoxelSpace.mgz* o en caso contrario ejecutar algo como

```
$ segmentThalamicNuclei.sh subject
```

Esto por supuesto he de lanzarlo integrado en el cluster pues demora un rato. Dentro del script sería algo como,

```

                                unless ( -e
$ENV{'SUBJECTS_DIR'}. '/' . $study . '_' . $subject . '/mri/ThalamicNuclei.v12.T1.FSvoxelSpace.mgz' ){
                                my %ptask = ( 'filename' =>
$outdir . '/segthalamus_' . $subject . '.sh',
                                'job_name' => 'segthalamus_' . $study,
                                'cpus' => 2,
                                'time' => '2:0:0',
                                'mem_per_cpu' => '4G',
                                'partition' => 'fast',
                                'output' => $outdir . '/segthalamus',
                                'command' =>
'segmentThalamicNuclei.sh ' . $study . '_' . $subject,
                                );
                                my $jobid = send2slurm(\%ptask);
                                push @jobs, $jobid;
                                }

```

y despues aviso cuando terminen las segmentaciones,

```

if (scalar(@jobs)) {
    my $ljobs = join(',', @jobs);
    $ljobs = 'afterok:' . $ljobs;
    my %hey = (
        'filename' => $outdir . '/segthalamus_end.sh',
        'job_name' => 'segthalamus_' . $study,
        'output' => $outdir . '/segthalamus_end',
        'dependency' => $ljobs,
    );
    send2slurm(\%hey);
}

```

Procesos a correr

Ha cuatro pasos distintos que han de ejecutarse,

<https://surfer.nmr.mgh.harvard.edu/fswiki/trac-all#Processingstepoptions>

Pero el objetivo no es ejecutar directamente los pasos sino tomarlos por separado y enviarlos a slurm. *trac-all* tiene la opción *-jobs* que en lugar de ejecutar las ordenes las escribe en un archivo de texto y nosotros somos los responsables de enviarlas al cluster.

-prep

El primer paso es el *pre-processing*. Ejecutamos el comando apropiado,

```
[tester@detritus fstest]$ trac-all -prep -c dmri.rc -jobs trac_step1.txt
INFO: SUBJECTS_DIR is /nas/data/subjects
INFO: Diffusion root is /nas/data/subjects
Actual FREESURFER_HOME /nas/usr/local/opt/freesurfer
```

y obtenemos la lista de comandos a ejecutarse,

```
[tester@detritus fstest]$ cat trac_step1.txt
trac-preproc -c /nas/data/subjects/fstest_0001/scripts/dmrirc.local -log
/nas/data/subjects/fstest_0001/scripts/trac-all.log -cmd
/nas/data/subjects/fstest_0001/scripts/trac-all.cmd
trac-preproc -c /nas/data/subjects/fstest_0027/scripts/dmrirc.local -log
/nas/data/subjects/fstest_0027/scripts/trac-all.log -cmd
/nas/data/subjects/fstest_0027/scripts/trac-all.cmd
trac-preproc -c /nas/data/subjects/fstest_0106/scripts/dmrirc.local -log
/nas/data/subjects/fstest_0106/scripts/trac-all.log -cmd
/nas/data/subjects/fstest_0106/scripts/trac-all.cmd
trac-preproc -c /nas/data/subjects/fstest_0119/scripts/dmrirc.local -log
/nas/data/subjects/fstest_0119/scripts/trac-all.log -cmd
/nas/data/subjects/fstest_0119/scripts/trac-all.cmd
trac-preproc -c /nas/data/subjects/fstest_0187/scripts/dmrirc.local -log
/nas/data/subjects/fstest_0187/scripts/trac-all.log -cmd
/nas/data/subjects/fstest_0187/scripts/trac-all.cmd
```

Luego, es tan sencillos como hacer la plantilla, ejecutar un comando simple y luego lanzar el contenido del archivo output en el cluster.

[Esto mas o menos se hace asi,](#)

[ctrac_prep.pl](#)

```
#!/usr/bin/perl

# Copyright 2020 O. Sotolongo <asqwerty@gmail.com>

# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
```

```

# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
use strict; use warnings;

use File::Find::Rule;
use NEURO4 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 $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/ctrac_prep.hlp';
exit;}
}
my $study = shift;
unless ($study) { print_help $ENV{'PIPEDIR'}.'/doc/ctrac_prep.hlp';
exit;}
my %std = load_project($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 $outdir = "$std{'DATA'}/slurm";
check_or_make($outdir);

my @subjects = cut_shit($db, $data_dir."/".$cfile);
my $dmrirc = $data_dir.'/dmri.rc';
my $tmp_orders = 'trac_step1.txt';
unless (-e $dmrirc && -r $dmrirc){
    my $subjlist = 'set subjlist = ( ';
    my $dmclist = 'set dcmlist = ( ';
    my $bveclist = 'set bveclist = ( ';
    my $bavllist = 'set bvallist = ( ';

    foreach my $subject (@subjects) {
        my %nifti = check_subj($std{'DATA'},$subject);
        if($nifti{'dwi'}){
            $subjlist.= $study.'_'. $subject.' ';
            $dmclist.= $nifti{'dwi'}.' ';
        }
    }
}

```

```

        (my $bvec = $nifti{'dwi'}) =~ s/nii\.gz$/bvec/;
        (my $bval = $nifti{'dwi'}) =~ s/nii\.gz$/bval/;
        $bveclist.= $bvec.' ';
        $bavllist.= $bval.' ';
    }
}
$subjlist.=')';
$dmclist.=')';
$bveclist.=')';
$bavllist.=')';
open CIF, ">$dmrirc" or die "Couldnt open dmrirc file for
writing\n";
print CIF "$subjlist\n$dmclist\n$bveclist\n$bavllist\n";
close CIF;
}
my $pre_order = 'trac-all -prep -c '.$dmrirc.' -jobs '.$tmp_orders;
system($pre_order);
my $tmp_orders = 'trac_step1.txt';
unless (-e $dmrirc && -r $dmrirc){
    my $subjlist = 'set subjlist = ( ';
    my $dmclist = 'set dcmlist = ( ';
    my $bveclist = 'set bveclist = ( ';
    my $bavllist = 'set bvallist = ( ';

    foreach my $subject (@subjects) {
        my %nifti = check_subj($std{'DATA'}, $subject);
        if($nifti{'dwi'}){
            $subjlist.= $study.'_'.$subject.' ';
            $dmclist.= $nifti{'dwi'}.' ';
            (my $bvec = $nifti{'dwi'}) =~ s/nii\.gz$/bvec/;
            (my $bval = $nifti{'dwi'}) =~ s/nii\.gz$/bval/;
            $bveclist.= $bvec.' ';
            $bavllist.= $bval.' ';
        }
    }
}
$subjlist.=')';
$dmclist.=')';
$bveclist.=')';
$bavllist.=')';
open CIF, ">$dmrirc" or die "Couldnt open dmrirc file for
writing\n";
print CIF "$subjlist\n$dmclist\n$bveclist\n$bavllist\n";
close CIF;
}
my $pre_order = 'trac-all -prep -c '.$dmrirc.' -jobs '.$tmp_orders;
system($pre_order);

```

y deberia lanzarse,

```
$ ctrac_prep.pl [-cut <cut_file.txt>] proj
```

siendo *cut_file.txt* la lista de sujetos que queremos analizar exclusivamente (en caso de no ser el proyecto completo) y *proj* el nombre del proyecto.

-bedp

Este proceso es similar, se lanza el script,

```
[tester@detritus fstest]$ trac-all -bedp -c dmri.rc -jobs trac_step2.txt
```

Pero tiene una dificultad añadida, el output son tres archivos con ordenes. Cada archivo debe ejecutarse despues que el anterior haya terminado.

```
[tester@detritus fstest]$ ls trac_step2*
trac_step2.post.txt  trac_step2.pre.txt  trac_step2.txt
[tester@detritus fstest]$ cat trac_step2.pre.txt
bedpostx_preproc.sh /nas/data/subjects/fstest_0001/dmri
bedpostx_preproc.sh /nas/data/subjects/fstest_0027/dmri
bedpostx_preproc.sh /nas/data/subjects/fstest_0106/dmri
bedpostx_preproc.sh /nas/data/subjects/fstest_0119/dmri
bedpostx_preproc.sh /nas/data/subjects/fstest_0187/dmri
[tester@detritus fstest]$ cat trac_step2.txt
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 0 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 1 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 2 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 3 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 4 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 5 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 6 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 7 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 8 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 9 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 10 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 11 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 12 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
```



```
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 13 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 14 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 15 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 16 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 17 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 18 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0001/dmri 19 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 0 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 1 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 2 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 3 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 4 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 5 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 6 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 7 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 8 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 9 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 10 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 11 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 12 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 13 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 14 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 15 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 16 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 17 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 18 --nf=2 --
```



```
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0027/dmri 19 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 0 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 1 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 2 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 3 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 4 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 5 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 6 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 7 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 8 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 9 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 10 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 11 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 12 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 13 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 14 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 15 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 16 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 17 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 18 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0106/dmri 19 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 0 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 1 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 2 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 3 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
```

```
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 4 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 5 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 6 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 7 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 8 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 9 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 10 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 11 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 12 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 13 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 14 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 15 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 16 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 17 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 18 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0119/dmri 19 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 0 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 1 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 2 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 3 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 4 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 5 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 6 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 7 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 8 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 9 --nf=2 --
```

```

fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 10 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 11 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 12 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 13 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 14 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 15 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 16 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 17 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 18 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
bedpostx_single_slice.sh /nas/data/subjects/fstest_0187/dmri 19 --nf=2 --
fudge=1 --bi=1000 --nj=1250 --se=25 --model=1 --cnonlinear
[tester@detritus fstest]$ cat trac_step2.post.txt
bedpostx_postproc.sh /nas/data/subjects/fstest_0001/dmri
bedpostx_postproc.sh /nas/data/subjects/fstest_0027/dmri
bedpostx_postproc.sh /nas/data/subjects/fstest_0106/dmri
bedpostx_postproc.sh /nas/data/subjects/fstest_0119/dmri
bedpostx_postproc.sh /nas/data/subjects/fstest_0187/dmri

```

Esto lo voy a hacer aprovechando el sistema de dependencias de SLURM. Lanzo el primer grupo, pongo una bandera que dependa de que todos terminen, lanzo el segundo grupo, haciendo que dependa de la bandera y así sucesivamente. 😊

Tiene su truco pero no es tan complicado

[ctrac_bedp.pl](#)

```

#!/usr/bin/perl

# Copyright 2020 O. Sotolongo <asqwerty@gmail.com>

# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#

```

```

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 $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/ctrac_prep.hlp';
exit;}
}
my $study = shift;
unless ($study) { print_help $ENV{'PIPEDIR'}.'./doc/ctrac_prep.hlp';
exit;}
my %std = load_project($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 $outdir = "$std{'DATA'}/slurm";
check_or_make($outdir);

my @subjects = cut_shit($db, $data_dir."/".$cfile);
my $tmp_orders = 'trac_step2.txt';
my $dmrirc = $data_dir.'/dmri.rc';
unless (-e $dmrirc && -r $dmrirc){
    my $subjlist = 'set subjlist = ( ';
    my $dmclist = 'set dcmlist = ( ';
    my $bveclist = 'set bveclist = ( ';
    my $bavllist = 'set bvallist = ( ';

    foreach my $subject (@subjects) {
        my %nifti = check_subj($std{'DATA'},$subject);
        if($nifti{'dwi'}){
            $subjlist.=$study.'_'.'$subject.' ' ';
            $dmclist.=$nifti{'dwi'}.' ' ';
            (my $bvec = $nifti{'dwi'}) =~ s/nii\.gz$/bvec/;
            (my $bval = $nifti{'dwi'}) =~ s/nii\.gz$/bval/;
            $bveclist.=$bvec.' ' ';
            $bavllist.=$bval.' ' ';

```

```

    }
}
$subjlist.=')';
$dmclist.=')';
$bveclist.=')';
$bavllist.=')';
open CIF, ">$dmrirc" or die "Couldnt open dmrirc file for
writing\n";
print CIF "$subjlist\n$dmclist\n$bveclist\n$bavllist\n";
close CIF;
}
my $pre_order = 'trac-all -bedp -c '.$dmrirc.' -jobs '.$tmp_orders;
system($pre_order);
#run pre-bedp first
(my $pre_tmp_orders = $tmp_orders) =~ s/\.txt/\.pre\.txt/;
open CORD, "<$pre_tmp_orders" or die "Could find orders file";
while (<CORD>){
    (my $subj) = /subjects\/$study\_(.*)\/dmri/;
    my $cpath =
$fsdir.'/'.$study.'_'.$subj.'/dmri.bedpostX/logs/monitor';
    system('mkdir -p '.$cpath);
    my $orderfile = $outdir.'/'.$subj.'_trac_pre_bedp.sh';
    open ORD, ">$orderfile";
    print ORD '#!/bin/bash'\n";
    print ORD '#SBATCH -J trac_pre_bedp_'. $study.\n";
    print ORD '#SBATCH --time=8:0:0'\n";
    print ORD '#SBATCH --mail-type=FAIL,TIME_LIMIT,STAGE_OUT'\n";
    print ORD '#SBATCH -o '.$outdir.'/trac_prep-%j'\n";
    print ORD '#SBATCH -c 8'\n";
    print ORD '#SBATCH --mem-per-cpu=4G'\n";
    print ORD '#SBATCH -p fast'\n";
    print ORD '#SBATCH --mail-user='.$ENV{'USER'}\n";
    print ORD;
    close ORD;
    system("sbatch $orderfile");
}
close CORD;
my $orderfile = $outdir.'/trac_pre_bedp_end.sh';
open ORD, ">$orderfile";
print ORD '#!/bin/bash'\n";
print ORD '#SBATCH -J trac_pre_bedp_'. $study.\n";
print ORD '#SBATCH --mail-type=END'\n"; #email cuando termine
print ORD '#SBATCH --mail-user='.$ENV{'USER'}\n";
print ORD '#SBATCH -p fast'\n";
print ORD '#SBATCH -o '.$outdir.'/trac_pre_bedp_end-%j'\n";
print ORD ":\n";
close ORD;
my $order = 'sbatch --dependency=singleton '.$orderfile;
my $jobid = ` $order `;
$jobid = ( split ' ', $jobid )[-1 ];
#run bedp now

```

```

my $count = 0;
open CORD, "<$tmp_orders" or die "Could find orders file";
while (<CORD>){
    (my $subj) = /subjects\/$study\_(.*)\/dmri/;
    my $orderfile = $outdir.'/'.$subj.'_'.$count.'_trac_bedp.sh';
    open ORD, ">$orderfile";
    print ORD '#!/bin/bash'."\n";
    print ORD '#SBATCH -J trac_bedp_'.$study."\n";
    print ORD '#SBATCH --time=12:0:0'."\n";
    print ORD '#SBATCH --mail-type=FAIL,TIME_LIMIT,STAGE_OUT'."\n";
    print ORD '#SBATCH -o '.$outdir.'/trac_bedp-%j'."\n";
    print ORD '#SBATCH -c 8'."\n";
    print ORD '#SBATCH --mem-per-cpu=4G'."\n";
    print ORD '#SBATCH -p fast'."\n";
    print ORD '#SBATCH --mail-user='.$ENV{'USER'}"\n";
    print ORD 'srun ';
    print ORD;
    close ORD;
    system('sbatch --depend=afterok:'.$jobid.' '$orderfile);
    $count++;
}
close CORD;
$orderfile = $outdir.'/trac_bedp_end.sh';
open ORD, ">$orderfile";
print ORD '#!/bin/bash'."\n";
print ORD '#SBATCH -J trac_bedp_'.$study."\n";
print ORD '#SBATCH --mail-type=END'."\n"; #email cuando termine
print ORD '#SBATCH --mail-user='.$ENV{'USER'}"\n";
print ORD '#SBATCH -p fast'."\n";
print ORD '#SBATCH -o '.$outdir.'/trac_bedp_end-%j'."\n";
print ORD ":\n";
close ORD;
$order = 'sbatch --dependency=singleton '$orderfile;
$jobid = ` $order `;
$jobid = ( split ' ', $jobid )[-1 ];
#and post.bedp later
(my $post_tmp_orders = $tmp_orders) =~ s/\.txt /\.post\.txt/;
open CORD, "<$post_tmp_orders" or die "Could find orders file";
while (<CORD>){
    (my $subj) = /subjects\/$study\_(.*)\/dmri/;
    my $orderfile = $outdir.'/'.$subj.'_'_trac_post_bedp.sh';
    open ORD, ">$orderfile";
    print ORD '#!/bin/bash'."\n";
    print ORD '#SBATCH -J trac_post_bedp_'.$study."\n";
    print ORD '#SBATCH --time=12:0:0'."\n";
    print ORD '#SBATCH --mail-type=FAIL,TIME_LIMIT,STAGE_OUT'."\n";
    print ORD '#SBATCH -o '.$outdir.'/trac_post-%j'."\n";
    print ORD '#SBATCH -c 8'."\n";
    print ORD '#SBATCH --mem-per-cpu=4G'."\n";
    print ORD '#SBATCH -p fast'."\n";
    print ORD '#SBATCH --mail-user='.$ENV{'USER'}"\n";

```



```

#s/bedpostx_postproc\.sh/bedpostx_postproc_gpu.sh/; #It doesnt
work :-(
    print ORD;
    close ORD;
    system('sbatch --depend=afterok:'. $jobid.' '$orderfile);
}
close CORD;
$orderfile = $outdir.'/trac_post_bedp_end.sh';
open ORD, ">$orderfile";
print ORD '#!/bin/bash'."\n";
print ORD '#SBATCH -J trac_post_bedp_'. $study."\n";
print ORD '#SBATCH --mail-type=END'."\n"; #email cuando termine
print ORD '#SBATCH --mail-user='.$ENV{'USER'}"\n";
print ORD '#SBATCH -p fast'."\n";
print ORD '#SBATCH -o '$outdir.'/trac_post_bedp_end-%j'."\n";
print ORD ":\n";
close ORD;
$order = 'sbatch --dependency=singleton '$orderfile;
exec($order);

```

y ya hemos hecho el bedpost.

OJO que esta parte demora bastante pero no mas que lanzar los bedpost con FSL.

-path

Y ahora el probtrack.

```

[tester@detritus fstest]$ trac-all -path -c dmri.rc -jobs trac_step3.txt
INFO: SUBJECTS_DIR is /nas/data/subjects
INFO: Diffusion root is /nas/data/subjects
Actual FREESURFER_HOME /nas/usr/local/opt/freesurfer

```

Son unas pocas lineas ahora,

```

[tester@detritus fstest]$ cat trac_step3.txt
trac-paths -c /nas/data/subjects/fstest_0001/scripts/dmrirc.local -log
/nas/data/subjects/fstest_0001/scripts/trac-all.log -cmd
/nas/data/subjects/fstest_0001/scripts/trac-all.cmd
trac-paths -c /nas/data/subjects/fstest_0027/scripts/dmrirc.local -log
/nas/data/subjects/fstest_0027/scripts/trac-all.log -cmd
/nas/data/subjects/fstest_0027/scripts/trac-all.cmd
trac-paths -c /nas/data/subjects/fstest_0106/scripts/dmrirc.local -log
/nas/data/subjects/fstest_0106/scripts/trac-all.log -cmd
/nas/data/subjects/fstest_0106/scripts/trac-all.cmd
trac-paths -c /nas/data/subjects/fstest_0119/scripts/dmrirc.local -log
/nas/data/subjects/fstest_0119/scripts/trac-all.log -cmd
/nas/data/subjects/fstest_0119/scripts/trac-all.cmd

```



```
trac-paths -c /nas/data/subjects/fstest_0187/scripts/dmrirc.local -log
/nas/data/subjects/fstest_0187/scripts/trac-all.log -cmd
/nas/data/subjects/fstest_0187/scripts/trac-all.cmd
```

Esto vuelve a ser sencillo, solo hay que reaprovechar un poco el script inicial,

[ctrac_path.pl](#)

```
#!/usr/bin/perl

# Copyright 2020 O. Sotolongo <asqwerty@gmail.com>

# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
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 $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/ctrac_prep.hlp';
exit;}
}
my $study = shift;
unless ($study) { print_help $ENV{'PIPEDIR'}.'/doc/ctrac_prep.hlp';
exit;}
my %std = load_project($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 $outdir = "$std{'DATA'}/slurm";
check_or_make($outdir);

my @subjects = cut_shit($db, $data_dir."/".$file);
my $dmrirc = $data_dir.'/dmri.rc';
my $tmp_orders = 'trac_step3.txt';
unless (-e $dmrirc && -r $dmrirc){
    my $subjlist = 'set subjlist = ( ';
    my $dmclist = 'set dcmlist = ( ';
    my $bveclist = 'set bveclist = ( ';
    my $bavllist = 'set bvallist = ( ';

    foreach my $subject (@subjects) {
        my %nifti = check_subj($std{'DATA'},$subject);
        if($nifti{'dwi'}){
            $subjlist.=$study.'_'.'$subject.' ' ';
            $dmclist.=$nifti{'dwi'}.' ' ';
            (my $bvec = $nifti{'dwi'}) =~ s/nii\.gz$/bvec/;
            (my $bval = $nifti{'dwi'}) =~ s/nii\.gz$/bval/;
            $bveclist.=$bvec.' ' ';
            $bavllist.=$bval.' ' ';
        }
    }
    $subjlist.=')';
    $dmclist.=')';
    $bveclist.=')';
    $bavllist.=')';
    open CIF, ">$dmrirc" or die "Couldnt open dmrirc file for
writing\n";
    print CIF "$subjlist\n$dmclist\n$bveclist\n$bavllist\n";
    close CIF;
}
my $pre_order = 'trac-all -path -c '.$dmrirc.' -jobs '.$tmp_orders;
system($pre_order);
open CORD, "<$tmp_orders" or die "Could find orders file";
while (<CORD>){
    (my $subj) = /subjects\/$study\_(\.*)\/scripts\/dmrirc/;
    #my $cpath = $fsdir.'/'.'$study.'_'. $subj.'/dmri.bedpostX/xfms';
    #system('mkdir -p '.$cpath);
    my $orderfile = $outdir.'/'.'$subj.'_trac_path.sh';
    open ORD, ">$orderfile";
    print ORD '#!/bin/bash'."\n";
    print ORD '#SBATCH -J trac_path_'. $study."\n";
    print ORD '#SBATCH --time=72:0:0'."\n";
    print ORD '#SBATCH --mail-type=FAIL,TIME_LIMIT,STAGE_OUT'."\n";
    print ORD '#SBATCH -o '.$outdir.'/trac_path-%j'."\n";
    print ORD '#SBATCH -c 8'."\n";
    print ORD '#SBATCH --mem-per-cpu=4G'."\n";
    print ORD '#SBATCH -p fast'."\n";
}

```

```

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

```

-stat

Y la ultima parte,

```

[tester@detritus fstest]$ trac-all -stat -c dmri.rc -jobs trac_step4.txt
INFO: SUBJECTS_DIR is /nas/data/subjects
INFO: Diffusion root is /nas/data/subjects
Actual FREESURFER_HOME /nas/usr/local/opt/freesurfer

```

Son unas lineas mas pero es iguaun solo archivo, sin dependencias,

```

[tester@detritus fstest]$ cat trac_step4.txt
/nas/usr/local/opt/freesurfer/bin/dmri_group --list
/nas/data/subjects/stats/lh.cst_AS.avg33_mni_bbr.inputs.txt --ref
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz --out
/nas/data/subjects/stats/lh.cst_AS.avg33_mni_bbr |& tee -a
/nas/data/subjects/stats/lh.cst_AS.avg33_mni_bbr.log
/nas/usr/local/opt/freesurfer/bin/dmri_group --list
/nas/data/subjects/stats/rh.cst_AS.avg33_mni_bbr.inputs.txt --ref
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz --out
/nas/data/subjects/stats/rh.cst_AS.avg33_mni_bbr |& tee -a
/nas/data/subjects/stats/rh.cst_AS.avg33_mni_bbr.log
/nas/usr/local/opt/freesurfer/bin/dmri_group --list
/nas/data/subjects/stats/lh.ilf_AS.avg33_mni_bbr.inputs.txt --ref
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz --out
/nas/data/subjects/stats/lh.ilf_AS.avg33_mni_bbr |& tee -a
/nas/data/subjects/stats/lh.ilf_AS.avg33_mni_bbr.log
/nas/usr/local/opt/freesurfer/bin/dmri_group --list
/nas/data/subjects/stats/rh.ilf_AS.avg33_mni_bbr.inputs.txt --ref
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz --out

```

```
/nas/data/subjects/stats/rh.ilf_AS.avg33_mni_bbr |& tee -a
/nas/data/subjects/stats/rh.ilf_AS.avg33_mni_bbr.log
/nas/usr/local/opt/freesurfer/bin/dmri_group --list
/nas/data/subjects/stats/lh.unc_AS.avg33_mni_bbr.inputs.txt --ref
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz --out
/nas/data/subjects/stats/lh.unc_AS.avg33_mni_bbr |& tee -a
/nas/data/subjects/stats/lh.unc_AS.avg33_mni_bbr.log
/nas/usr/local/opt/freesurfer/bin/dmri_group --list
/nas/data/subjects/stats/rh.unc_AS.avg33_mni_bbr.inputs.txt --ref
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz --out
/nas/data/subjects/stats/rh.unc_AS.avg33_mni_bbr |& tee -a
/nas/data/subjects/stats/rh.unc_AS.avg33_mni_bbr.log
/nas/usr/local/opt/freesurfer/bin/dmri_group --list
/nas/data/subjects/stats/fmajor_PP.avg33_mni_bbr.inputs.txt --ref
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz --out
/nas/data/subjects/stats/fmajor_PP.avg33_mni_bbr |& tee -a
/nas/data/subjects/stats/fmajor_PP.avg33_mni_bbr.log
/nas/usr/local/opt/freesurfer/bin/dmri_group --list
/nas/data/subjects/stats/fminor_PP.avg33_mni_bbr.inputs.txt --ref
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz --out
/nas/data/subjects/stats/fminor_PP.avg33_mni_bbr |& tee -a
/nas/data/subjects/stats/fminor_PP.avg33_mni_bbr.log
/nas/usr/local/opt/freesurfer/bin/dmri_group --list
/nas/data/subjects/stats/lh.atr_PP.avg33_mni_bbr.inputs.txt --ref
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz --out
/nas/data/subjects/stats/lh.atr_PP.avg33_mni_bbr |& tee -a
/nas/data/subjects/stats/lh.atr_PP.avg33_mni_bbr.log
/nas/usr/local/opt/freesurfer/bin/dmri_group --list
/nas/data/subjects/stats/rh.atr_PP.avg33_mni_bbr.inputs.txt --ref
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz --out
/nas/data/subjects/stats/rh.atr_PP.avg33_mni_bbr |& tee -a
/nas/data/subjects/stats/rh.atr_PP.avg33_mni_bbr.log
/nas/usr/local/opt/freesurfer/bin/dmri_group --list
/nas/data/subjects/stats/lh.ccg_PP.avg33_mni_bbr.inputs.txt --ref
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz --out
/nas/data/subjects/stats/lh.ccg_PP.avg33_mni_bbr |& tee -a
/nas/data/subjects/stats/lh.ccg_PP.avg33_mni_bbr.log
/nas/usr/local/opt/freesurfer/bin/dmri_group --list
/nas/data/subjects/stats/rh.ccg_PP.avg33_mni_bbr.inputs.txt --ref
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz --out
/nas/data/subjects/stats/rh.ccg_PP.avg33_mni_bbr |& tee -a
/nas/data/subjects/stats/rh.ccg_PP.avg33_mni_bbr.log
/nas/usr/local/opt/freesurfer/bin/dmri_group --list
/nas/data/subjects/stats/lh.cab_PP.avg33_mni_bbr.inputs.txt --ref
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz --out
/nas/data/subjects/stats/lh.cab_PP.avg33_mni_bbr |& tee -a
/nas/data/subjects/stats/lh.cab_PP.avg33_mni_bbr.log
/nas/usr/local/opt/freesurfer/bin/dmri_group --list
/nas/data/subjects/stats/rh.cab_PP.avg33_mni_bbr.inputs.txt --ref
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz --out
/nas/data/subjects/stats/rh.cab_PP.avg33_mni_bbr |& tee -a
```

```
/nas/data/subjects/stats/rh.cab_PP.avg33_mni_bbr.log
/nas/usr/local/opt/freesurfer/bin/dmri_group --list
/nas/data/subjects/stats/lh.slfp_PP.avg33_mni_bbr.inputs.txt --ref
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz --out
/nas/data/subjects/stats/lh.slfp_PP.avg33_mni_bbr |& tee -a
/nas/data/subjects/stats/lh.slfp_PP.avg33_mni_bbr.log
/nas/usr/local/opt/freesurfer/bin/dmri_group --list
/nas/data/subjects/stats/rh.slfp_PP.avg33_mni_bbr.inputs.txt --ref
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz --out
/nas/data/subjects/stats/rh.slfp_PP.avg33_mni_bbr |& tee -a
/nas/data/subjects/stats/rh.slfp_PP.avg33_mni_bbr.log
/nas/usr/local/opt/freesurfer/bin/dmri_group --list
/nas/data/subjects/stats/lh.slft_PP.avg33_mni_bbr.inputs.txt --ref
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz --out
/nas/data/subjects/stats/lh.slft_PP.avg33_mni_bbr |& tee -a
/nas/data/subjects/stats/lh.slft_PP.avg33_mni_bbr.log
/nas/usr/local/opt/freesurfer/bin/dmri_group --list
/nas/data/subjects/stats/rh.slft_PP.avg33_mni_bbr.inputs.txt --ref
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz --out
/nas/data/subjects/stats/rh.slft_PP.avg33_mni_bbr |& tee -a
/nas/data/subjects/stats/rh.slft_PP.avg33_mni_bbr.log
```

Hay que retocar un poco y ya esta,

[ctrac_stat.pl](#)

```
#!/usr/bin/perl

# Copyright 2020 O. Sotolongo <asqwerty@gmail.com>

# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
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 $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/ctrac_prep.hlp';
exit;}
}
my $study = shift;
unless ($study) { print_help $ENV{'PIPEDIR'}.'./doc/ctrac_prep.hlp';
exit;}
my %std = load_project($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 $outdir = "$std{'DATA'}/slurm";
check_or_make($outdir);

my @subjects = cut_shit($db, $data_dir."/".$cfile);
my $dmrirc = $data_dir.'/dmri.rc';
my $tmp_orders = 'trac_step4.txt';
unless (-e $dmrirc && -r $dmrirc){
    my $subjlist = 'set subjlist = ( ';
    my $dmclist = 'set dcmlist = ( ';
    my $bveclist = 'set bveclist = ( ';
    my $bavllist = 'set bvallist = ( ';

    foreach my $subject (@subjects) {
        my %nifti = check_subj($std{'DATA'},$subject);
        if($nifti{'dwi'}){
            $subjlist.=$study.'_'. $subject.' ';
            $dmclist.=$nifti{'dwi'}.' ';
            (my $bvec = $nifti{'dwi'}) =~ s/nii\.gz$/bvec/;
            (my $bval = $nifti{'dwi'}) =~ s/nii\.gz$/bval/;
            $bveclist.=$bvec.' ';
            $bavllist.=$bval.' ';
        }
    }
    $subjlist.=')';
    $dmclist.=')';
    $bveclist.=')';
    $bavllist.=')';
    open CIF, ">$dmrirc" or die "Couldnt open dmrirc file for
writing\n";
    print CIF "$subjlist\n$dmclist\n$bveclist\n$bavllist\n";
    close CIF;
}

```

```

}
my $pre_order = 'trac-all -stat -c '.$dmrirc.' -jobs '.$tmp_orders;
system($pre_order);
my $count = 0;
open CORD, "<$tmp_orders" or die "Could find orders file";
while (<CORD>){
    #(my $subj) = /subjects\/$study\_(.*)\/scripts\/dmrirc/;
    #my $cpath = $fsdir.'\/'.$study.'_'.$subj.'/dmri.bedpostX/xfms';
    #system('mkdir -p '.$cpath);
    my $orderfile = $outdir.'/group_trac_stat_'.$count.'.sh';
    open ORD, ">$orderfile";
    print ORD '#!/bin/bash'\n";
    print ORD '#SBATCH -J trac_stat_'.$study.\n";
    print ORD '#SBATCH --time=72:0:0'\n";
    print ORD '#SBATCH --mail-type=FAIL,TIME_LIMIT,STAGE_OUT'\n";
    print ORD '#SBATCH -o '.$outdir.'/trac_stat-%j'\n";
    print ORD '#SBATCH -c 8'\n";
    print ORD '#SBATCH --mem-per-cpu=4G'\n";
    print ORD '#SBATCH -p fast'\n";
    print ORD '#SBATCH --mail-user='.$ENV{'USER'}\n";
    print ORD;
    close ORD;
    system("sbatch $orderfile");
    $count++;
}
my $orderfile = $outdir.'/trac_stat_end.sh';
open ORD, ">$orderfile";
print ORD '#!/bin/bash'\n";
print ORD '#SBATCH -J trac_stat_'.$study.\n";
print ORD '#SBATCH --mail-type=END'\n"; #email cuando termine
print ORD '#SBATCH --mail-user='.$ENV{'USER'}\n";
print ORD '#SBATCH -p fast'\n";
print ORD '#SBATCH -o '.$outdir.'/trac_stat_end-%j'\n";
print ORD "mv $fsdir/stats $data_dir/stats\n";
close ORD;
my $order = 'sbatch --dependency=singleton '.$orderfile;
exec($order);

```

tl;dr

```

$ ctrac_prep.pl proj
$ ctrac_bedp.pl proj
$ ctrac_path.pl proj
$ ctrac_stat.pl proj

```


<input type="checkbox"/> ☆ ▷ Trolls United Co.	SLURM Job_id=133658 Name=trac_stat_fstest Ended, Run time 00:00:00, COMPLETED, ExitCode 0	8:43 PM
<input type="checkbox"/> ☆ ▷ Trolls United Co.	SLURM Job_id=133639 Name=trac_path_fstest Ended, Run time 00:00:00, COMPLETED, ExitCode 0	8:42 PM
<input type="checkbox"/> ☆ ▷ Trolls United Co.	SLURM Job_id=133633 Name=trac_post_bedp_fstest Ended, Run time 00:00:00, COMPLETED, ExitCode 0	7:47 PM
<input type="checkbox"/> ☆ ▷ Trolls United Co.	SLURM Job_id=133627 Name=trac_bedp_fstest Ended, Run time 00:00:01, COMPLETED, ExitCode 0	7:46 PM
<input type="checkbox"/> ☆ ▷ Trolls United Co.	SLURM Job_id=133526 Name=trac_pre_bedp_fstest Ended, Run time 00:00:00, COMPLETED, ExitCode 0	6:02 PM
<input type="checkbox"/> ☆ ▷ Trolls United Co.	SLURM Job_id=133519 Name=trac_prep_fstest Ended, Run time 00:00:00, COMPLETED, ExitCode 0	5:57 PM

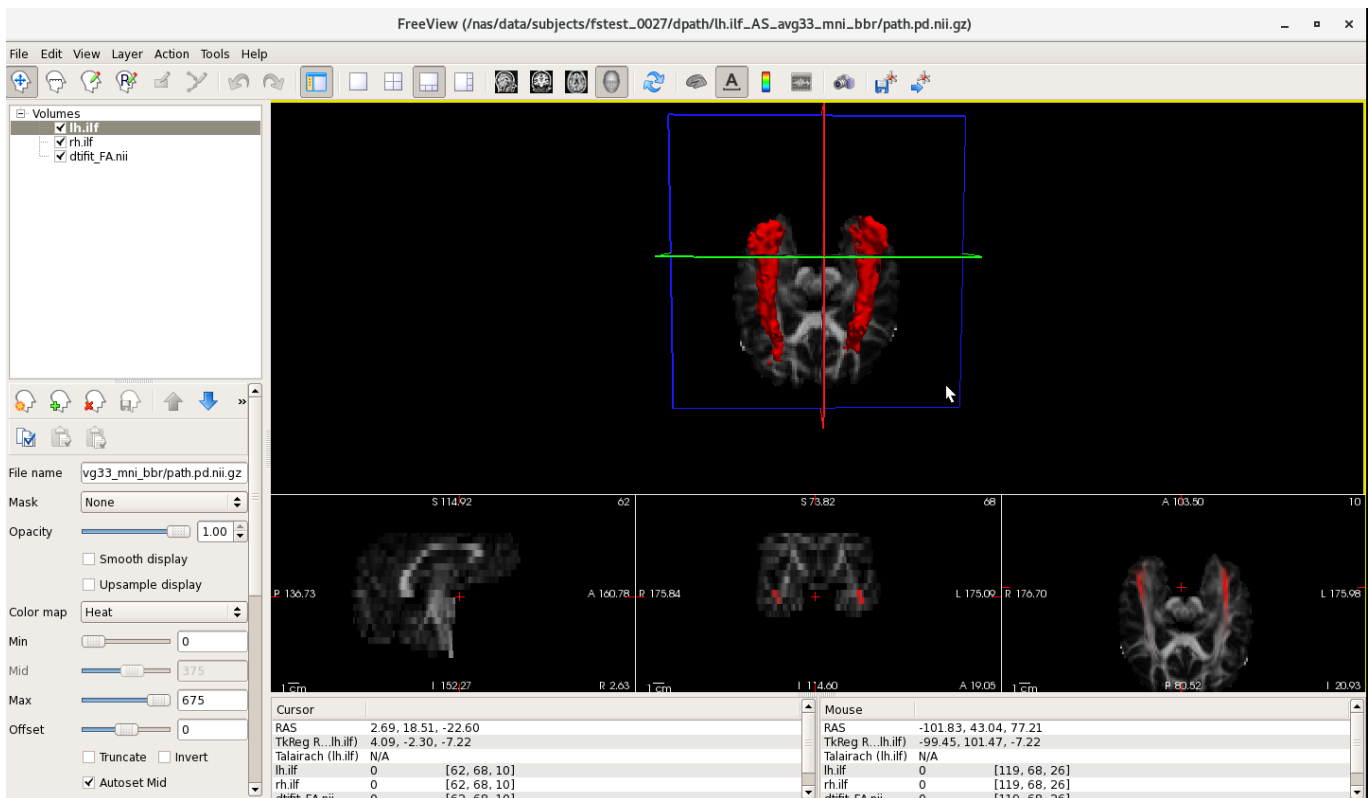
Lista de los PATHs

```
lh.cst_AS: Left corticospinal tract
rh.cst_AS: Right corticospinal tract
lh.ilf_AS: Left inferior longitudinal fasciculus
rh.ilf_AS: Right inferior longitudinal fasciculus
lh.unc_AS: Left uncinate fasciculus
rh.unc_AS: Right uncinate fasciculus
fmajor_PP: Corpus callosum - forceps major
fminor_PP: Corpus callosum - forceps minor
lh.atr_PP: Left anterior thalamic radiations
rh.atr_PP: Right anterior thalamic radiations
lh.ccg_PP: Left cingulum - cingulate gyrus endings
rh.ccg_PP: Right cingulum - cingulate gyrus endings
lh.cab_PP: Left cingulum - angular bundle
rh.cab_PP: Right cingulum - angular bundle
lh.slfp_PP: Left superior longitudinal fasciculus - parietal endings
rh.slfp_PP: Right superior longitudinal fasciculus - parietal endings
lh.slft_PP: Left superior longitudinal fasciculus - temporal endings
rh.slft_PP: Right superior longitudinal fasciculus - temporal endings
```

Resultados

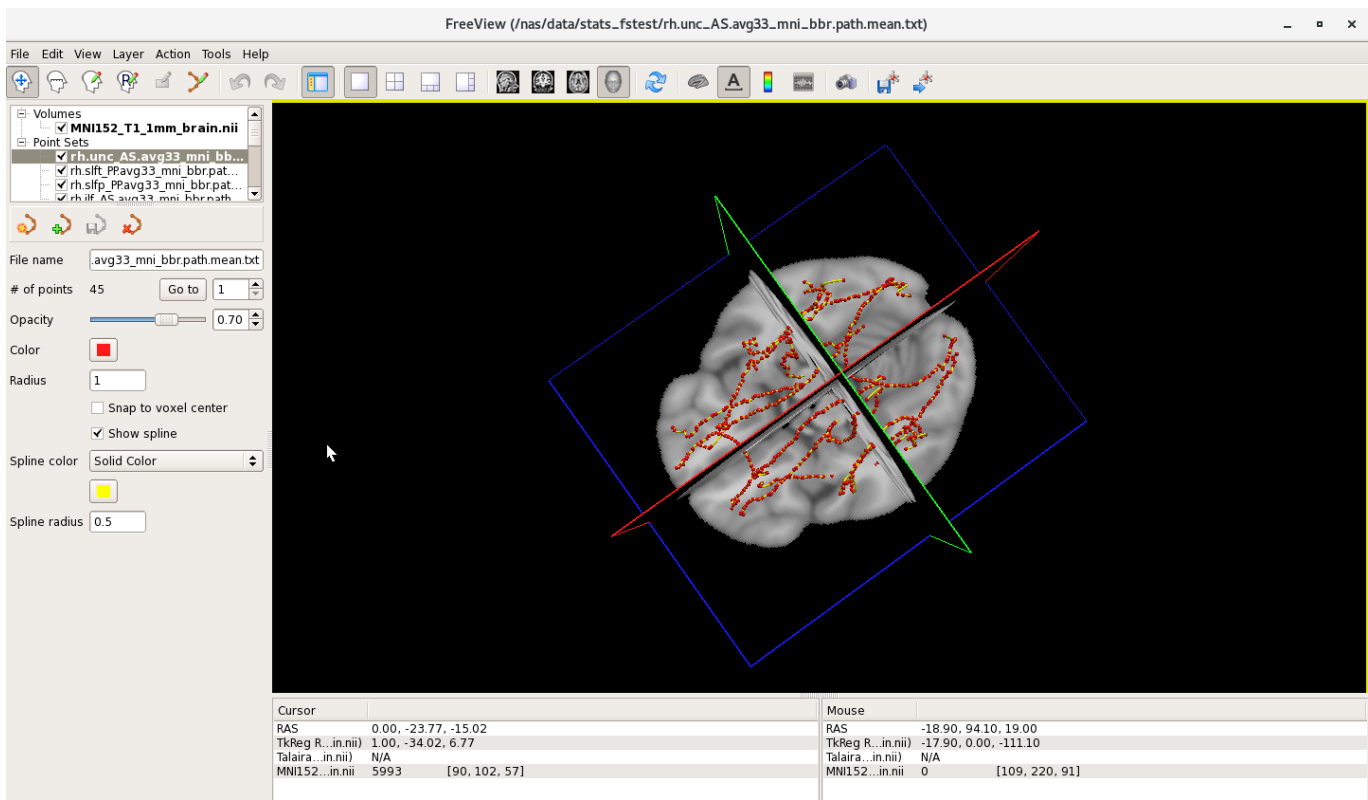
Tractos,

```
[fsluser@FSLVm7_64 ~]$ freeview -v
/nas/data/subjects/fstest_0027/dmri/dtifit_FA.nii.gz
/nas/data/subjects/fstest_0027/dpath/rh.ilf_AS_avg33_mni_bbr/path.pd.nii.gz:
colormap=heat:isosurface=0,0:color='Red':name=rh.ilf
/nas/data/subjects/fstest_0027/dpath/lh.ilf_AS_avg33_mni_bbr/path.pd.nii.gz:
colormap=heat:isosurface=0,0:color='Red':name=lh.ilf
```



Pathways,

```
[fsluser@FSLVm7_64 ~]$ freeview -v  
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz -w  
/nas/data/stats_fstest/*.path.mean.txt
```



Ojo, esto para FACEHBI no sirve

```
[fsluser@FSLVm7_64 ~]$ freeview -v
/usr/local/fsl/data/standard/MNI152_T1_1mm_brain.nii.gz
/nas/data/subjects/fstest_0027/dmri/mni/dtifit_FA.bbr.nii.gz &
```



Parseando

Freesurfer deja todos los resultados dentro del directorio del sujeto y saca, además, estadísticas grupales relativas a los tractos que se han calculado. Pero lo que queremos es algo distinto, determinar el valor medio de FA y MD en los tractos calculados, para todos los sujetos y ponerlo en un mismo archivo que se pueda integrar con los resultados del resto de proyectos.

Cada tracto está almacenado en un directorio independiente dentro del directorio *dpath*,

```
[tester@detritus fstest]$ ls /nas/data/subjects/fstest_0001/dpath/
fmajor_PP_avg33_mni_bbr lh.cab_PP_avg33_mni_bbr lh.ilf_AS_avg33_mni_bbr
lh.unc_AS_avg33_mni_bbr rh.cab_PP_avg33_mni_bbr rh.ilf_AS_avg33_mni_bbr
rh.unc_AS_avg33_mni_bbr
fminor_PP_avg33_mni_bbr lh.ccg_PP_avg33_mni_bbr lh.slfp_PP_avg33_mni_bbr
merged_avg33_mni_bbr.mgz rh.ccg_PP_avg33_mni_bbr rh.slfp_PP_avg33_mni_bbr
lh.atr_PP_avg33_mni_bbr lh.cst_AS_avg33_mni_bbr lh.slft_PP_avg33_mni_bbr
rh.atr_PP_avg33_mni_bbr rh.cst_AS_avg33_mni_bbr rh.slft_PP_avg33_mni_bbr
```

Y dentro de este subdirectorio nos interesa el archivo *pathstats.overall.txt*,

```
[tester@detritus fstest]$ ls
/nas/data/subjects/fstest_0001/dpath/fmajor_PP_avg33_mni_bbr/
endpt1.pd.nii.gz  endpt2.pd.nii.gz  length.samples.txt  log.txt
path.map.nii.gz  path.map.txt  path.pd.nii.gz  path.pd.trk
pathstats.byvoxel.txt  pathstats.overall.txt  pd.samples.txt
```

Y dentro de este archivo el valor que necesitamos,

```
[tester@detritus fstest]$ grep "FA_Avg "
/nas/data/subjects/fstest_0001/dpath/fmajor_PP_avg33_mni_bbr/pathstats.overall.txt
FA_Avg 0.528634
```

Entonces, solo hay que ir sujeto por sujeto compilando estos valores y escribiendolos en un archivo de salida.

[Esto se resume facilmente](#)

[ctract_metrics.pl](#)

```
#!/usr/bin/perl

use strict; use warnings;

use File::Find::Rule;
use NEURO4 qw(get_subjects print_help get_pair get_list load_project
shit_done cut_shit);
use Data::Dump qw(dump);
use File::Basename qw(basename);

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/ctract_metrics.hlp';
exit;}
}

my $study = shift;
unless ($study) { print_help $ENV{'PIPEDIR'}.'/doc/ctract_metrics.hlp';
exit;}
my %std = load_project($study);

my $w_dir=$std{'WORKING'};
my $data_dir=$std{'DATA'};
my $subjsdir = $ENV{'SUBJECTS_DIR'};
my $db = $std{'DATA'}.'/'.$study.'_mri.csv';
my %trdirs = ( 'fmajor' => 'fmajor_PP_avg33_mni_bbr',
```

```

    'lh.cab' => 'lh.cab_PP_avg33_mni_bbr',
    'lh.ilf' => 'lh.ilf_AS_avg33_mni_bbr',
    'lh.unc' => 'lh.unc_AS_avg33_mni_bbr',
    'rh.cab' => 'rh.cab_PP_avg33_mni_bbr',
    'rh.ilf' => 'rh.ilf_AS_avg33_mni_bbr',
    'rh.unc' => 'rh.unc_AS_avg33_mni_bbr',
    'fminor' => 'fminor_PP_avg33_mni_bbr',
    'lh.ccg' => 'lh.ccg_PP_avg33_mni_bbr',
    'lh.slpf' => 'lh.slpf_PP_avg33_mni_bbr',
    'rh.ccg' => 'rh.ccg_PP_avg33_mni_bbr',
    'rh.slpf' => 'rh.slpf_PP_avg33_mni_bbr',
    'lh.atr' => 'lh.atr_PP_avg33_mni_bbr',
    'lh.cst' => 'lh.cst_AS_avg33_mni_bbr',
    'lh.slft' => 'lh.slft_PP_avg33_mni_bbr',
    'rh.atr' => 'rh.atr_PP_avg33_mni_bbr',
    'rh.cst' => 'rh.cst_AS_avg33_mni_bbr',
    'rh.slft' => 'rh.slft_PP_avg33_mni_bbr',
);
my $pathr_file = 'pathstats.overall.txt';
print "Collecting needed files\n";

my @dtis = cut_shit($db, $data_dir."/".$.scfile);

my %csv;

my $ofile = $data_dir."/".$.study."_dti_tracula.csv";
open OF, ">$ofile";
print OF "Subject";
foreach my $track (sort keys %trdirs){
    print OF ";$track","_FA",";$track","_MD";
}
print OF "\n";

my %results;

foreach my $subject (@dtis){
    if($subject){
        print OF "$subject";
        my $fa; my $md;
        my $spath =
$subjsdir.'/'.$.study.'_'. $subject.'/dpath/';
        foreach my $track (sort keys %trdirs){
            my $ifile =
$spath.$trdirs{$track}.'/'. $pathr_file;
            open IRF, "<$ifile";
            while (<IRF>) {
                if(/^FA_Avg\s/) {($fa) = /^FA_Avg
(0\.\d+)\$/;};
                if(/^MD_Avg\s/) {($md) = /^MD_Avg
(0\.\d+)\$/;};
            }
        }
    }
}

```

```
        close IRF;
        print OF ";$fa;$md";
    }
    print OF "\n";
}

close OF;
my $zfile=$std{DATA}."/". $study."_dti_tracula_results.tgz";
system("tar czf $zfile $ofile");
shit_done basename($ENV{$_}), $study, $zfile;
```

Basta hacer,

```
$ ctrac_metrics.pl proj
```

y el archivo de salida sera algo como,

```
[tester@detrirus fstest]$ head fstest_dti_tracula.csv
Subject;fmajor_FA;fmajor_MD;fminor_FA;fminor_MD;lh.atr_FA;lh.atr_MD;lh.cab_F
A;lh.cab_MD;lh.ccg_FA;lh.ccg_MD;lh.cst_FA;lh.cst_MD;lh.ilf_FA;lh.ilf_MD;lh.s
lfp_FA;lh.slfp_MD;lh.slft_FA;lh.slft_MD;lh.unc_FA;lh.unc_MD;rh.atr_FA;rh.atr
_MD;rh.cab_FA;rh.cab_MD;rh.ccg_FA;rh.ccg_MD;rh.cst_FA;rh.cst_MD;rh.ilf_FA;rh
.ilf_MD;rh.slfp_FA;rh.slfp_MD;rh.slft_FA;rh.slft_MD;rh.unc_FA;rh.unc_MD
0001;0.528634;0.000827984;0.418862;0.000837851;0.38762;0.000808873;0.285617;
0.000864355;0.480356;0.000927924;0.486141;0.00071761;0.404125;0.000826366;0.
311155;0.000771376;0.359977;0.000805707;0.364496;0.000803023;0.36817;0.00080
3661;0.347436;0.000818787;0.398203;0.000860262;0.467309;0.00073149;0.42421;0.
000816664;0.349221;0.000768625;0.384073;0.000786565;0.368076;0.000801726
0027;0.537571;0.000800813;0.420022;0.000846507;0.420897;0.00079418;0.290632;
0.000903317;0.353056;0.000831895;0.508674;0.000771471;0.418434;0.000842544;0.
329285;0.00079904;0.368359;0.000812858;0.35905;0.000849702;0.393285;0.00081
1594;0.26983;0.000928979;0.361511;0.000850733;0.464204;0.000768423;0.422892;
0.000849195;0.333106;0.000782956;0.371509;0.000784775;0.361369;0.000828747
0106;0.570249;0.000748165;0.452091;0.000846462;0.423598;0.000723258;0.296025
;0.000793815;0.467001;0.000824213;0.509462;0.000698;0.458105;0.000795525;0.3
82791;0.000692079;0.433996;0.000715618;0.370571;0.000816281;0.424781;0.00071
601;0.226781;0.000810964;0.462459;0.000751646;0.477618;0.000710536;0.409424;
0.000773103;0.382323;0.000694046;0.407935;0.000691525;0.378478;0.00080298
0119;0.614449;0.00074317;0.446337;0.000752267;0.386983;0.000713591;0.324273;
0.000784348;0.441098;0.00081737;0.54778;0.000678324;0.528145;0.000754968;0.3
8144;0.000697071;0.437348;0.000700224;0.447795;0.000759076;0.386529;0.000724
425;0.273868;0.000809277;0.440816;0.000772664;0.502312;0.000701165;0.4749;0.
000745737;0.370977;0.000699952;0.442473;0.000708482;0.396617;0.000739047
0187;0.466919;0.00087339;0.400457;0.000966647;0.362489;0.000821896;0.211317;
0.00098259;0.348662;0.000967363;0.498185;0.00080447;0.363993;0.00089466;0.33
1463;0.0008317;0.382911;0.000838648;0.309629;0.00091525;0.343212;0.000843028
;0.275517;0.000922106;0.349813;0.000884783;0.459096;0.000764194;0.40892;0.00
0906821;0.342572;0.000813444;0.386516;0.000817556;0.34704;0.000918453
```

Updating to TRACULA FS 7.2

Con la nueva version de FS, se introduce la capacidad de corregir el DTI con otro hecho en sentido inverso. Vamos a hacer esto obligatorio en el pipeline.

```
[osotolongo@brick03 ~]$ make_proj.pl ttrac /nas/bioface_raw
[osotolongo@brick03 ~]$ cd /nas/data/ttrac/
[osotolongo@brick03 ttrac]$ update_mri_db.pl ttrac
[osotolongo@brick03 bids]$ bulk2bids.pl ttrac
[osotolongo@brick03 ttrac]$ cp ttrac_mri.csv ttrac_mri.csv.orig
[osotolongo@brick03 ttrac]$ head ttrac_mri.csv.orig > ttrac_mri.csv
```

cambios dmri.rc

Voy a generar un archivo *dmri* para que se tenga en cuenta la correccion ([ctrac_dmri.pl](#)).

[Aqui necesito los valores de Echo spacing](#) y *EPI factor*. Esto lo tengo que sacar del protocolo de DTI de cada proyecto.

Secuencia - Parte 1

Nombre de la secuencia	epse
Tipo pulsos RF	Normal
Modo Gradiente	Rápido
Ancho de banda	1314 Hz/Px
Esp. entre ecos libre	Desconectar
Esp. entre ecos	0,96 ms
Optimizacion	Min. TE
Factor EPI	112

Y ahora basicamente lo que hago es algo asi,

```
ctrac_dmri.pl -epif 112 -echosp 0.96 ttrac
```

que genera un archivo como,

```
[osotolongo@brick03 ttrac]$ cat dmri.rc
set dob0 = 2
set sublist = ( ttrac_0002 ttrac_0002 ttrac_0004 ttrac_0004 ttrac_0006
ttrac_0006 ttrac_0007 ttrac_0007 ttrac_0009 ttrac_0009 ttrac_0010 ttrac_0010
)
set dcmlist = ( /old_nas/data/ttrac/bids/sub-0002/dwi/sub-0002_dwi.nii.gz
/old_nas/data/ttrac/bids/sub-0002/dwi/sub-0002_dir-PA_sbref_dwi.nii.gz
```



```

/old_nas/data/ttrac/bids/sub-0004/dwi/sub-0004_dwi.nii.gz
/old_nas/data/ttrac/bids/sub-0004/dwi/sub-0004_dir-PA_sbref_dwi.nii.gz
/old_nas/data/ttrac/bids/sub-0006/dwi/sub-0006_dwi.nii.gz
/old_nas/data/ttrac/bids/sub-0006/dwi/sub-0006_dir-PA_sbref_dwi.nii.gz
/old_nas/data/ttrac/bids/sub-0007/dwi/sub-0007_run-01_dwi.nii.gz
/old_nas/data/ttrac/bids/sub-0007/dwi/sub-0007_dir-PA_sbref_dwi.nii.gz
/old_nas/data/ttrac/bids/sub-0009/dwi/sub-0009_dwi.nii.gz
/old_nas/data/ttrac/bids/sub-0009/dwi/sub-0009_dir-PA_sbref_dwi.nii.gz
/old_nas/data/ttrac/bids/sub-0010/dwi/sub-0010_dwi.nii.gz
/old_nas/data/ttrac/bids/sub-0010/dwi/sub-0010_dir-PA_sbref_dwi.nii.gz )
set bveclist = ( /old_nas/data/ttrac/bids/sub-0002/dwi/sub-0002_dwi.bvec
/nas/software/neuro.dev/lib/b0.bvec
/old_nas/data/ttrac/bids/sub-0004/dwi/sub-0004_dwi.bvec
/nas/software/neuro.dev/lib/b0.bvec
/old_nas/data/ttrac/bids/sub-0006/dwi/sub-0006_dwi.bvec
/nas/software/neuro.dev/lib/b0.bvec
/old_nas/data/ttrac/bids/sub-0007/dwi/sub-0007_run-01_dwi.bvec
/nas/software/neuro.dev/lib/b0.bvec
/old_nas/data/ttrac/bids/sub-0009/dwi/sub-0009_dwi.bvec
/nas/software/neuro.dev/lib/b0.bvec
/old_nas/data/ttrac/bids/sub-0010/dwi/sub-0010_dwi.bvec
/nas/software/neuro.dev/lib/b0.bvec )
set bvallist = ( /old_nas/data/ttrac/bids/sub-0002/dwi/sub-0002_dwi.bval
/nas/software/neuro.dev/lib/b0.bval
/old_nas/data/ttrac/bids/sub-0004/dwi/sub-0004_dwi.bval
/nas/software/neuro.dev/lib/b0.bval
/old_nas/data/ttrac/bids/sub-0006/dwi/sub-0006_dwi.bval
/nas/software/neuro.dev/lib/b0.bval
/old_nas/data/ttrac/bids/sub-0007/dwi/sub-0007_run-01_dwi.bval
/nas/software/neuro.dev/lib/b0.bval
/old_nas/data/ttrac/bids/sub-0009/dwi/sub-0009_dwi.bval
/nas/software/neuro.dev/lib/b0.bval
/old_nas/data/ttrac/bids/sub-0010/dwi/sub-0010_dwi.bval
/nas/software/neuro.dev/lib/b0.bval )
set pedir = ( AP PA AP PA AP PA AP PA AP PA AP PA )
set echospacing = 0.96
set epifactor = 112

```

Aqui guardan los valores del *B0* en direccion *PA* y todo lo que es necesario para hacer la correccion.

cambios -prep

El proximo commando, que seria algo como,

```
[osotolongo@brick03 ttrac]$ trac-all -prep -c dmri.rc -jobs trac_step1.txt
```

genera las ordenes que deben lanzarse en paralelo,

```
[osotolongo@brick03 ttrac]$ cat trac_step1.txt
```

```
trac-preproc -c /old_nas/subjects/ttrac_0002/scripts/dmrirc.local -log
/old_nas/subjects/ttrac_0002/scripts/trac-all.log -cmd
/old_nas/subjects/ttrac_0002/scripts/trac-all.cmd
trac-preproc -c /old_nas/subjects/ttrac_0004/scripts/dmrirc.local -log
/old_nas/subjects/ttrac_0004/scripts/trac-all.log -cmd
/old_nas/subjects/ttrac_0004/scripts/trac-all.cmd
trac-preproc -c /old_nas/subjects/ttrac_0006/scripts/dmrirc.local -log
/old_nas/subjects/ttrac_0006/scripts/trac-all.log -cmd
/old_nas/subjects/ttrac_0006/scripts/trac-all.cmd
trac-preproc -c /old_nas/subjects/ttrac_0007/scripts/dmrirc.local -log
/old_nas/subjects/ttrac_0007/scripts/trac-all.log -cmd
/old_nas/subjects/ttrac_0007/scripts/trac-all.cmd
trac-preproc -c /old_nas/subjects/ttrac_0009/scripts/dmrirc.local -log
/old_nas/subjects/ttrac_0009/scripts/trac-all.log -cmd
/old_nas/subjects/ttrac_0009/scripts/trac-all.cmd
trac-preproc -c /old_nas/subjects/ttrac_0010/scripts/dmrirc.local -log
/old_nas/subjects/ttrac_0010/scripts/trac-all.log -cmd
/old_nas/subjects/ttrac_0010/scripts/trac-all.cmd
```

Asi como el config para que cada orden se ejecute individualmente,

```
[osotolongo@brick03 ttrac]$ cat
/old_nas/subjects/ttrac_0002/scripts/dmrirc.local
# Run command file generated automatically by trac-all
#
setenv FREESURFER_HOME /usr/local/freesurfer/7.2.0-1
setenv SUBJECTS_DIR /old_nas/subjects
set dtroot = (/old_nas/subjects)
set subj = (ttrac_0002)
#
# Processing steps to be performed
#
set docorr = (1)
set doqa = (1)
set dointra = (1)
set dotensor = (1)
set dointer = (1)
set dopriors = (1)
#
# Options for image corrections and tensor fit
#
set dcmroot = ()
set dcmfile = (/old_nas/data/ttrac/bids/sub-0002/dwi/sub-0002_dwi.nii.gz
/old_nas/data/ttrac/bids/sub-0002/dwi/sub-0002_dir-PA_sbref_dwi.nii.gz)
set bvecfile = (/old_nas/data/ttrac/bids/sub-0002/dwi/sub-0002_dwi.bvec
/nas/software/neuro.dev/lib/b0.bvec)
set bvalfile = (/old_nas/data/ttrac/bids/sub-0002/dwi/sub-0002_dwi.bval
/nas/software/neuro.dev/lib/b0.bval)
set bmax = ()
set bshell = ()
```

```
set nb0 = ()
set dob0 = (2)
set b0mfile = ()
set b0pfile = ()
set echospacing = (0.96)
set epifactor = (112)
set pedir = (AP PA)
set dTE = ()
set doeddy = (2)
set dorotbvecs = (1)
set thrbet = (0.3)
#
# Options for registrations
#
set intrareg = (3)
set intracost = ()
set intradof = (6)
set intrarot = ()
set interreg = (5)
set intercost = ()
set intertrg =
(/usr/local/freesurfer/7.2.0-1/trctrain/hcp/MGH35_HCP_FA_template.nii.gz)
#
# Options for anatomical segmentation
#
set segname = (aparc+aseg)
set gmgrow = (2)
set usehahnuc = (1)
set usemaskanat = (1)
#
# Options for ball-and-stick model fit
#
set nstick = (2)
#
# Options for path reconstructions
#
set pathlist = (acomm cc.bodyc cc.bodyp cc.bodypf cc.bodypm cc.bodyt cc.genu
cc.rostrum cc.splenium mcp lh.af rh.af lh.ar rh.ar lh.atr rh.atr lh.cbd
rh.cbd lh.cbv rh.cbv lh.cst rh.cst lh.emc rh.emc lh.fat rh.fat lh.fx rh.fx
lh.ilf rh.ilf lh.mlf rh.mlf lh.or rh.or lh.slf1 rh.slf1 lh.slf2 rh.slf2
lh.slf3 rh.slf3 lh.uf rh.uf)
set gmids = (0 0 0 0 0 0 0 0 0 0 0 0 0 10 49 10 49 0 0 0 0 16 16 0 0 0 0 28 60
0 0 0 0 10 49 0 0 0 0 0 0 0 0)
set ncpts = (7 8 9 7 7 12 5 5 7 7 9 9 4 4 4 4 7 7 5 5 7 7 7 7 5 5 9 9 8 8 7
7 5 5 7 7 8 8 6 6 7 7)
set trainsubjlist = ($FREESURFER_HOME/trctrain/hcp/mgh_1001
$FREESURFER_HOME/trctrain/hcp/mgh_1006
$FREESURFER_HOME/trctrain/hcp/mgh_1007
$FREESURFER_HOME/trctrain/hcp/mgh_1008
$FREESURFER_HOME/trctrain/hcp/mgh_1009
$FREESURFER_HOME/trctrain/hcp/mgh_1010)
```

```

$FREESURFER_HOME/trctrain/hcp/mgh_1011
$FREESURFER_HOME/trctrain/hcp/mgh_1012
$FREESURFER_HOME/trctrain/hcp/mgh_1013
$FREESURFER_HOME/trctrain/hcp/mgh_1015
$FREESURFER_HOME/trctrain/hcp/mgh_1016
$FREESURFER_HOME/trctrain/hcp/mgh_1017
$FREESURFER_HOME/trctrain/hcp/mgh_1019
$FREESURFER_HOME/trctrain/hcp/mgh_1021
$FREESURFER_HOME/trctrain/hcp/mgh_1023
$FREESURFER_HOME/trctrain/hcp/mgh_1024)
set ntrainlist = (16)
set avgname = (avg)
set reinit = (0)
set usestrunc = (1)
set doxyzprior = (0)
set dosegprior = (1)
set dotangprior = (0)
set dopathsubdirs = (0)
set fmin = (0.05)
set nburnin = (200)
set nsample = (7500)
set nupdate = (0)
set nkeep = (5)
set doinitprop = (1)
set pmin = (.2)
set projmin = (-3)
set projmax = (3)
set dproj = (.1)
set overwrite = (1)
#
# Location of TRACULA executables
#
set trcdir = (/usr/local/freesurfer/7.2.0-1/bin)

```

Este paso y los siguientes son similares a la version antigua de FS, y estan en [ctrac_prep.pl](#), [ctrac_bedp.pl](#), [ctrac_path.pl](#) y [ctrac_stat.pl](#).

Aqui hay que añadir un *FIX* cuando se hace el *-prep*. Segun

<https://www.mail-archive.com/freesurfer@nmr.mgh.harvard.edu/msg70655.html>, se han de lanzar las tareas independientes con una orden previa de *setenv*

MY_MORPHS_DO_NOT_CONFORM_DEAL_WITH_IT. Asi que aqui el envio de ordenes toma la forma,

```

my %ptask;
$ptask{'job_name'} = 'trac_prep_'. $study;
$ptask{'cpus'} = $cpus;
$ptask{'time'} = '72:0:0';
$ptask{'mem_per_cpu'} = '4G';
$ptask{'partition'} = 'fast';
while (<CORD>){
    #ahora hay que enviar cada orden de este archivo al cluster

```

```

(my $subj) = /subjects\/$study\/(.*)\/scripts\/dmrirc/;
#print "$subj\n";
$ptask{'filename'} = $outdir.'/' . $subj . '_trac_prep.sh';
$ptask{'output'} = $outdir.'/' . 'trac_prep';
$ptask{'command'} = 'export
MY_MORPHS_DO_NOT_CONFORM_DEAL_WITH_IT=1' . "\n" . $_;
send2slurm(\%ptask);
}
close CORD;

```

Por simplicidad he quitado la capacidad de los scripts intermedios de construir el *dmri.rc* así que ahora es obligatorio que exista siempre.

metrics

Las métricas también han cambiado en esta versión (<https://dmri.mgh.harvard.edu/tract-atlas/>). En la versión 7.2 puedo obtener la lista de paths haciendo algo así,

```

[osotolongo@brick03 ttrac]$ ls
/old_nas/subjects/ttrac_0002/dpath/*/pathstats.overall.txt | awk -F"/"
'{print $6}' | awk -F "_" {'printf("\047%s\047 => \047%s\047,\n", $1, $0)'}
'acomm' => 'acomm_avg16_syn_bbr',
'cc.bodyc' => 'cc.bodyc_avg16_syn_bbr',
'cc.bodyp' => 'cc.bodyp_avg16_syn_bbr',
'cc.bodypf' => 'cc.bodypf_avg16_syn_bbr',
'cc.bodypm' => 'cc.bodypm_avg16_syn_bbr',
'cc.bodyt' => 'cc.bodyt_avg16_syn_bbr',
'cc.genu' => 'cc.genu_avg16_syn_bbr',
'cc.rostrum' => 'cc.rostrum_avg16_syn_bbr',
'cc.splenium' => 'cc.splenium_avg16_syn_bbr',
'lh.af' => 'lh.af_avg16_syn_bbr',
'lh.ar' => 'lh.ar_avg16_syn_bbr',
'lh.atr' => 'lh.atr_avg16_syn_bbr',
'lh.cbd' => 'lh.cbd_avg16_syn_bbr',
'lh.cbv' => 'lh.cbv_avg16_syn_bbr',
'lh.cst' => 'lh.cst_avg16_syn_bbr',
'lh.emc' => 'lh.emc_avg16_syn_bbr',
'lh.fat' => 'lh.fat_avg16_syn_bbr',
'lh.fx' => 'lh.fx_avg16_syn_bbr',
'lh.ilf' => 'lh.ilf_avg16_syn_bbr',
'lh.mlf' => 'lh.mlf_avg16_syn_bbr',
'lh.or' => 'lh.or_avg16_syn_bbr',
'lh.slf1' => 'lh.slf1_avg16_syn_bbr',
'lh.slf2' => 'lh.slf2_avg16_syn_bbr',
'lh.slf3' => 'lh.slf3_avg16_syn_bbr',
'lh.uf' => 'lh.uf_avg16_syn_bbr',
'mcp' => 'mcp_avg16_syn_bbr',
'rh.af' => 'rh.af_avg16_syn_bbr',
'rh.ar' => 'rh.ar_avg16_syn_bbr',
'rh.atr' => 'rh.atr_avg16_syn_bbr',

```

```
'rh.cbd' => 'rh.cbd_avg16_syn_bbr',  
'rh.cbv' => 'rh.cbv_avg16_syn_bbr',  
'rh.cst' => 'rh.cst_avg16_syn_bbr',  
'rh.emc' => 'rh.emc_avg16_syn_bbr',  
'rh.fat' => 'rh.fat_avg16_syn_bbr',  
'rh.fx' => 'rh.fx_avg16_syn_bbr',  
'rh.ilf' => 'rh.ilf_avg16_syn_bbr',  
'rh.mlf' => 'rh.mlf_avg16_syn_bbr',  
'rh.or' => 'rh.or_avg16_syn_bbr',  
'rh.slf1' => 'rh.slf1_avg16_syn_bbr',  
'rh.slf2' => 'rh.slf2_avg16_syn_bbr',  
'rh.slf3' => 'rh.slf3_avg16_syn_bbr',  
'rh.uf' => 'rh.uf_avg16_syn_bbr',
```

tl;dr

Entonces, ahora el procedimiento general es,

```
$ ctrac_dmri.pl -epif XXX -echosp X.XX proj  
$ ctrac_prep.pl proj  
$ ctrac_bedp.pl proj  
$ ctrac_path.pl proj  
$ ctrac_stat.pl proj  
$ ctrac_metrics.pl proj
```

From:

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

Permanent link:

<http://detritus.fundacioace.com/wiki/doku.php?id=neuroimagen:tracula>

Last update: **2021/08/20 15:12**

