

Problemas y soluciones comunes relativas al proyecto FACEHBI

procesamiento paulatino

MRI

to NiFTI

Las imagenes van llegando poco a poco y se han de ir procesando paulatinamente. Lo primero es saber cual de las imagenes que han subido no han sido procesadas,

```
$ ls /home/data/subjects/facehbi_smc0*/stats/aseg.stats | awk -F"_smc"
{'print $2'} | awk -F"/" {'print $1'} > mri_hechos.txt
$ sed 's/0/F/' mri_hechos.txt > mri_hechos.dir
$ ls /nas/corachan/facehbi/ > mri_up.dir
$ grep -v "`cat mri_hechos.dir`" mri_up.dir > yet.dir
$ for a in `cat yet.dir`; do for x in -d /nas/corachan/facehbi/$a/*; do if
[[ `dckey -k "SeriesDescription" $x/Img00001.dcm 2>&1 | grep t1_mprage` ]];
then dcm2nii -o tmp/ $x/Img00001.dcm; fi; done; mkdir processed/$a; mv tmp/*
processed/$a; done
$ sed 's/F/0/;s/\(.*\)\/\1;smc/' mri_hechos.dir > /nas/facehbi/wave1_mri.csv
$ sed 's/F/0/;s/\(.*\)\/\1;smc/' yet.dir > /nas/facehbi/wave2_mri.csv
$ cat /nas/facehbi/wave1_mri.csv /nas/facehbi/wave2_mri.csv >
/nas/facehbi/facehbi.csv
$ for a in `ls processed/F*/*.nii.gz`; do b=$(echo $a | sed
s'/processed\F\/nas\/facehbi\/mri\/smc0/; s\/s0/s00/; s/a1001//'); cp $a
$b; done
$ pfs12fs.pl -cut wave2_mri.csv facehbi
```

DELETEME

Ya las tengo en el archivo *yet.txt* Ahora las copio en otro sitio (*al que no tenga acceso el que las sube*) y las proceso.

```
$ for a in `cat yet.txt`; do cp -r /nas/corachan/facehbi/$a
/nas/raw_images/facehbi/mri/; done
$ for a in `cat yet.txt`; do for x in -d /nas/raw_images/facehbi/mri/$a/*;
do if [[ `dckey -k "SeriesDescription" $x/Img00001.dcm 2>&1 | grep
t1_mprage` ]]; then dcm2nii -o tmp/ $x/Img00001.dcm; fi; done; mkdir
processed/$a; mv tmp/* processed/$a; done
```

Averiguar cual me falta por pasarle el FS

```
$ ls -d data/subjects/facehbi* | awk -F"smc" {'print $2'} >
```

```
~/data/facehbi/fs_done.txt
$ ls -d /nas/corachan/facehbi/* | awk -F"/" {'print $5'} | sed 's/F/0/' >
~/data/facehbi/haymri.txt
$ grep -v "`cat ~/data/facehbi/fs_done.txt`" ~/data/facehbi/haymri.txt
```

Ahora viene la parte jodida. He de copiar las imagenes correctamente segun la nomenclatura del procesamiento. Voy a ir una por una y copiarlas segun el esquema,

Una a una seria asi,

```
$ cp processed/F042/s005a1001.nii.gz /nas/facehbi/mri/smc0042s0005.nii.gz
```

Esto se puede mejorar!!!

```
$ for a in `ls processed/F*/*.nii.gz`; do b=$(echo $a | sed
s'/processed\F\/\nas\/facehbi\/mri\/smc0/; s\/s0/s00/; s/a1001//'); echo
"$a $b"; done
```

to Freesurfer

Ahora tengo que convertir el archivo *yet.txt* en una lista que entienda el pipeline.

```
$ sed 's/F/0;/s\/(.*)\/\1;smc/' yet.txt > /nas/facehbi/wave2_mri.csv
```

He modificado el script *pfs/2fs.pl* para que me deje asignarle solo un cierto numero de sujetos en un csv. Esto deberia funcionar. 😊

```
$ pfs/2fs.pl -cut wave2_mri.csv facehbi
```

No se ha de olvidar el incluir las iamgenes nuevas en el listado del estudio

```
$ mv facehbi.csv wave1_mri.csv
$ cat wave1_mri.csv wave2_mri.csv > facehbi.csv
```

Ahora ya se podria procesar con Freesurfer

```
$ precon.pl -cut wave2_mri.csv facehbi
```

DTI

to NiFTI

```
$ for a in `ls /nas/corachan/facehbi/`; do for x in -d
/nas/corachan/facehbi/$a/*; do if [[ `dckey -k "SeriesDescription"
$x/Img00001.dcm 2>&1 | grep "ep2d_diff_mddw_64_p2$" ` ]]; then dcm2nii -o
/nas/facehbi/tmp_dti/ $x/; fi; done; mkdir processed/$a; mv tmp_dti/*
processed/$a; done
```

```
$ cd /nas/facehbi
$ for x in `find processed/ -name "*.bval"`; do nm=$(echo $x | sed
's/processed\/F\(.*\)\s\(.*\)a001.bval/smc0\1s0\2/'); cp $x dti/${nm}.bval;
cp ${x%.bval}.bvec dti/${nm}.bvec; cp ${x%.bval}.nii.gz dti/${nm}.nii.gz;
done
```

Procesamiento

```
$ cat acqparams.txt
0 1 0 0.12192
0 -1 0 0.12192

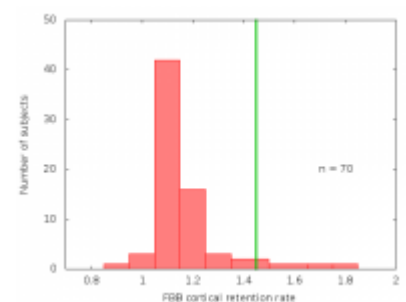
$ indx=""
$ for ((i=1; i<=143; i+=1)); do indx="$indx 1"; done
$ echo $indx > dti_index.txt

$ dti_reg.pl -nocorr facehbi

$ dti_metrics.pl facehbi
$ dti_metrics.pl -a1 facehbi
$ dti_metrics_custom.pl -uofm SN facehbi
$ dti_metrics_custom.pl -uofm SN_anterior facehbi

$ dti_track.pl -uofm DMN facehbi
$ for x in `ls -d working/smc0*_probtrack_out`; do mv $x `echo $x | sed
's/out/DMN/'`;done
$ dti_metrics_alt.pl -path DMN facehbi
```

FBB



to NiFTI

Para ir procesando según vayan llegando las imágenes,

```
$ cp -ru clinic/* raw_images/facehbi/fbb/
$ ls raw_images/facehbi/fbb/ | sed 's/FACEHBI-F\(.*\)B\/\1/' >
facehbi/fbb_copied.txt
$ grep -v "`cat facehbi/fbb_done.txt`" facehbi/fbb_copied.txt >
```

```
facehbi/fbb_wave2.txt
$ grep -v "`cat facehbi/fbb_done.txt`" facehbi/fbb_copied.txt | sed
's/\(.*\)\/0\1;smc/' > facehbi/wave2_fbb.csv
$ sed 's/\(.*\)\/0\1;smc/' facehbi/fbb_copied.txt > facehbi/wave1_fbb.csv
$ sed 's/\(.*\)\/0\1;smc/' facehbi/fbb_copied.txt > facehbi/all_fbb.csv
```

Otro ejemplo puede ser, si ya tengo hecho un pedazo, ¿Como saco de las que no he procesado? **Pues así**

```
$ ls /nas/clinic/ | sed 's/FACEHBI.*F\(.*\)B.*\/0\1/' | sort | uniq >
hayfbb.txt
$ awk -F";" {'print $1'} facehbi_fbb_fs_suvr.csv > hechos.txt
$ grep -v "`cat hechos.txt`" hayfbb.txt | sed 's/\(.*\)\/\1;smc/' > yet.txt
```

Problemas de almacenamiento y conversión de las imágenes



Los DICOM llegan en carpetas de hasta 500 imágenes DICOM. Esto quiere decir que hay que copiar previamente todas las imágenes que queremos convertir a una misma carpeta y a partir de ahí es que se convierten.

Si tenemos un archivo txt (yet.txt en el ejemplo) y sabemos cual imagen hay que buscar hacemos,

```
$ for x in `awk -F";" {'print $1'} yet.txt`; do y=/nas/clinic/FACEHBI-
F$(echo $x|sed 's/0\(.*\)\/\1/')B/DICOM/; for od in `find $y -maxdepth 2 -
name "*0000" -o -name "*0001"`; do for okf in `ls $od`; do if [[ `dckey -k
"SeriesDescription" $od/$okf 2>&1 | grep "5min"` ]]; then cp $od/$okf
/nas/facehbi/tmp_2nifti/; fi; done; ff=$(ls /nas/facehbi/tmp_2nifti/ | head -
n 1); dcm2nii -o /nas/facehbi/tmp/ /nas/facehbi/tmp_2nifti/$ff; done; rm -rf
/nas/facehbi/tmp_2nifti/*; nf=$(ls /nas/facehbi/tmp/*.nii.gz | head -n 1);
fslsplit $nf /nas/facehbi/fbb_ok/smc${x}s; rm -rf /nas/facehbi/tmp/*; done
```



[Esto está mejor aquí](#)

Datos demograficos

lo que sale del omi es una mierda y hay que convertirlo en algo potable:

```
$ awk -F"," '{printf "%04d, %s, %d, %s, %s\n", $7,$9,$4,$3,$8}'
mierda_demo.csv | sed 's/Varón/1;/s/Mujer/0;/s/
\([0-9]*\)\/\([0-9]*\)\/\([0-9]*\)\/, \3-\2-\1/g' | awk -F"," '{printf "%04d,
%d, %d, %d\n", $1,$2,$3,$5-$4}' > mier_demo.csv
```

y sacamos la edad de la fecha de nacimiento y la del screening

```
#!/usr/bin/env perl

use strict;
use warnings;
use utf8;
use Date::Manip;
use Math::Round;

my $ifile = shift;
my $heads;
open IDF, "<$ifile" or die "Couldn't find input file";
while(<IDF>){
    if(/^Subject.*/){
        print "Subject, Gender, Education, Age\n";
    }
    if(my ($nodate, $date1, $date2) = /(\d{4}, \d+, \d{1,2}), (\d+-\d+-\d+),
(\d+-\d+-\d+)/){
        my $age = nearest(0.1,
Delta_Format(DateCalc($date1,$date2),"%hv")/(24*365.2425));
        print "$nodate, $age\n";
    }
}
close IDF;
```

Extract image date

MRI


First approach,

```
[osotolongo@detritus facehbi]$ for y in /nas/corachan/facehbi/*; do for x in
${y}/*; do if [[ `dckey -k "SeriesDescription" ${x}/Img00001.dcm 2>&1 | grep
t1_mprage` ]]; then a=$(dckey -k "StudyDate" ${x}/Img00001.dcm 2>&1); fi;
done; echo ${y} ${a};done
/nas/corachan/facehbi/F001 20141205
/nas/corachan/facehbi/F002 20141205
/nas/corachan/facehbi/F003 20141211
/nas/corachan/facehbi/F004 20141212
.....
```

Vamos a ordenarlo un poco,

```
[osotolongo@detritus facehbi]$ for y in /nas/corachan/facehbi/*; do for x in
${y}/*; do if [[ `dckey -k "SeriesDescription" ${x}/Img00001.dcm 2>&1 | grep
t1_mprage` ]]; then a=$(dckey -k "AcquisitionDate" ${x}/Img00001.dcm 2>&1);
fi; done; echo ${y} ${a};done | sed 's/.*F/0/; s/ /;/ ' >
```

```
/home/osotolongo/facehbi/dicom_mri_v0_date.csv
```

O mas rapido,  Delete!

```
[osotolongo@detritus facehbi]$ for y in /nas/corachan/facehbi_2/*; do for x  
in `find ${y} -type f | head -n 1`; do a=$(dckey -k "AcquisitionDate" ${x}  
2>&1); done; echo ${y} ${a}; done | sed 's/.*F/0/; s/_.* / /; s/ /;/ ' | uniq  
| grep -v 0ACEHBIS > /home/osotolongo/facehbi/dicom_mri_v2_date.csv
```

Cuidado que en la visita 2 hay algunas adquisiciones de la visita 0.

```
[osotolongo@detritus corachan]$ for y in /nas/corachan/facehbi_2/*; do for x  
in `find ${y} -type f | head -n 1`; do a=$(dckey -k "AcquisitionDate" ${x}  
2>&1); done; echo ${y} ${a}; done  
.....  
/nas/corachan/facehbi_2/F171 20171130  
/nas/corachan/facehbi_2/F171_._._(1D15024531) 20151128  
.....
```

Vamos a darle una vuelta,

```
[osotolongo@detritus corachan]$ for y in /nas/corachan/facehbi_2/*; do for x  
in `find ${y} -type f | head -n 1`; do a=$(dckey -k "AcquisitionDate" ${x}  
2>&1); done; echo ${y} ${a}; done | sed 's/.*F/0/; s/_.* / /; s/ /;/ ' | grep  
-v 0ACEHBIS > /home/osotolongo/facehbi/dicom_mri_v2_pre_date.csv
```

y voy a hacer un script en perl para que escoja el que tenga la fecha mas alta,

[clean_pre_date.pl](#)

```
#!/usr/bin/perl  
  
use strict;  
use warnings;  
  
my $ifile = '/home/osotolongo/facehbi/dicom_mri_v2_pre_date.csv';  
my $ofile = '/home/osotolongo/facehbi/dicom_mri_v2_date.csv';  
my %imgdates;  
  
open IDF, "<$ifile" or die "No such file!";  
while (<IDF>){  
    (my $subject, my $imdate) = /(.*);(.*)/;  
    if (exists($imgdates{$subject}))  
        $imgdates{$subject} = $imdate unless  
($imgdates{$subject} > $imdate);  
    }else{  
        $imgdates{$subject} = $imdate;  
    }  
}  
close IDF;
```

```
open ODF, ">$ofile" or die "Couldn't create file";
foreach my $subject (sort keys %imgdates){
    print ODF "$subject;$imgdates{$subject}\n";
}
close ODF;
```

A ver ahora,

```
[osotolongo@detritus facehbi]$ ./clean_pre_date.pl
[osotolongo@detritus facehbi]$ grep "0171;" dicom_mri_v2_date.csv
0171;20171130
```

Parece que funciona.

Encontrar los directorios erroneos en la V2

Primero lista de los directorios y fecha de los DCM,

```
[osotolongo@detritus facehbi_2]$ for y in /nas/corachan/facehbi_2/*; do for
x in `find ${y} -type f | head -n 1`; do a=$(dckey -k "AcquisitionDate" ${x}
2>&1); done; s=$(echo ${y} | sed 's/.*F/0/; s/_.*//'); echo ${s} ${a} ${y};
done | grep -v repe | sed 's/ /;/g'>
/home/osotolongo/facehbi/listado_mri_v2.csv
[osotolongo@detritus facehbi_2]$ head
/home/osotolongo/facehbi/listado_mri_v2.csv
0001;20170124;/nas/corachan/facehbi_2/F001
0002;20170323;/nas/corachan/facehbi_2/F002
0003;20170123;/nas/corachan/facehbi_2/F003
0005;20170123;/nas/corachan/facehbi_2/F005
0006;20170124;/nas/corachan/facehbi_2/F006
0007;20170120;/nas/corachan/facehbi_2/F007
0008;20170125;/nas/corachan/facehbi_2/F008
0009;20170207;/nas/corachan/facehbi_2/F009
0010;20170208;/nas/corachan/facehbi_2/F010
0011;20170127;/nas/corachan/facehbi_2/F011
```

hago un script para escoger los de menor fecha,

[find_bad_guys.pl](#)

```
#!/usr/bin/perl

use strict;
use warnings;

my $ifile = '/home/osotolongo/facehbi/listado_mri_v2.csv';
my %imgdates;
```

```
open IDF, "<$ifile" or die "No such file!";
while (<IDF>){
    (my $subject, my $imdate, my $imdir) = /(.*);(.*);(.*)/;
    if (exists($imgdates{$subject}) &&
exists($imgdates{$subject}{'date'})){
        if ($imgdates{$subject}{'date'} > $imdate) {
            print "$imdir; $imdate\n";
        }else{
            print "$imgdates{$subject}{'dir'};
$imgdates{$subject}{'date'}\n";
            $imgdates{$subject}{'date'} = $imdate;
            $imgdates{$subject}{'dir'} = $imdir;
        }
    }else{
        $imgdates{$subject}{'date'} = $imdate;
        $imgdates{$subject}{'dir'} = $imdir;
    }
}
close IDF;
```

Y vamos a comprobar cuantos directorios repetidos o de otra visita hay,

```
[osotolongo@detritus facehbi]$ ./find_bad_guys.pl
/nas/corachan/facehbi_2/F122_._._(1D17105595); 20171003
/nas/corachan/facehbi_2/F134_._._(1D15001121); 20151003
/nas/corachan/facehbi_2/F135_._._(1D15004700); 20151013
/nas/corachan/facehbi_2/F164_._._(1D15028382); 20151209
/nas/corachan/facehbi_2/F171_._._(1D15024531); 20151128
/nas/corachan/facehbi_2/F173_._._(1D15026521); 20151203
/nas/corachan/facehbi_2/F174_._._(1D15029459); 20151211
/nas/corachan/facehbi_2/F175_._._(1D15025918); 20151202
/nas/corachan/facehbi_2/F177_._._(1D16000959); 20160107
/nas/corachan/facehbi_2/F178_._._(1D15029317); 20151211
/nas/corachan/facehbi_2/F179_._._(1D16002329); 20160111
/nas/corachan/facehbi_2/F181_._._(1D15027377); 20151205
/nas/corachan/facehbi_2/F182_._._(1D15032421); 20151220
/nas/corachan/facehbi_2/F183_._._(1D15033327); 20151222
/nas/corachan/facehbi_2/F184_._._(1D15029736); 20151212
/nas/corachan/facehbi_2/F185_._._(1D15029773); 20151212
/nas/corachan/facehbi_2/F186_._._(1D15033618); 20151223
/nas/corachan/facehbi_2/F188_._._(1D15032797); 20151221
/nas/corachan/facehbi_2/F189_._._(1D16007077); 20160121
/nas/corachan/facehbi_2/F190_._._(1D15033280); 20151222
/nas/corachan/facehbi_2/F192_._._(1D15034692); 20151228
/nas/corachan/facehbi_2/F194_._._(1D15035643); 20151230
/nas/corachan/facehbi_2/F195_._._(1D16003897); 20160114
/nas/corachan/facehbi_2/F196_._._(1D16003334); 20160113
/nas/corachan/facehbi_2/F197_._._(2D16002969); 20160112
/nas/corachan/facehbi_2/F198_._._(1D16001968); 20160109
```




Lo he escrito con la fecha para que me sirva de guía pero en fin,

```
[osotolongo@detritus facehbi]$ ./find_bad_guys.pl | awk -F";" '{print $1}' >
mover_estos.txt
[osotolongo@detritus facehbi]$ su -
Password:
[root@detritus ~]# cd /nas/corachan/facehbi_2/
[root@detritus facehbi_2]# mkdir badguys
[root@detritus facehbi_2]# for x in `cat
/home/osotolongo/facehbi/mover_estos.txt`; do mv ${x} badguys/; done
[root@detritus facehbi_2]# ls badguys/
F122_._._(1D17105595)  F171_._._(1D15024531)  F177_._(1D16000959)
F182_._._(1D15032421)  F186_._._(1D15033618)  F192_._._(1D15034692)
F197_._._(2D16002969)
F134_._._(1D15001121)  F173_._(1D15026521)   F178_._._(1D15029317)
F183_._(1D15033327)   F188_._._(1D15032797)  F194_._._(1D15035643)
F198_._._(1D16001968)
F135_._._(1D15004700)  F174_._._(1D15029459)  F179_._._(1D16002329)
F184_._._(1D15029736)  F189_._._(1D16007077)  F195_._._(1D16003897)
F164_._(1D15028382)   F175_._._(1D15025918)  F181_._._(1D15027377)
F185_._._(1D15029773)  F190_._._(1D15033280)  F196_._._(1D16003334)
```

Y ahora viene lo bueno pues hay que arreglar y/o rehacer toda la visita 2 por los posibles errores producto de esto.

FBB

one random file by subject,

```
[osotolongo@detritus facehbi]$ find /nas/clinic/facehbi/FACEHBI-F001B/DICOM/
-type f | head -n 1
/nas/clinic/facehbi/FACEHBI-F001B/DICOM/15012118/23390000/27112349
```

ahora,

```
[osotolongo@detritus facehbi]$ for y in /nas/clinic/facehbi/*; do for x in
`find ${y}/DICOM/ -type f | head -n 1`; do a=$(dckey -k "AcquisitionDate"
${x} 2>&1); done; echo ${y} ${a}; done | sed 's/.*-F/0/; s/B//; s/ /;/' >
/home/osotolongo/facehbi/dicom_fbb_v0_date.csv
[osotolongo@detritus facehbi]$ for y in /nas/clinic/facehbi_2/*; do for x in
`find ${y}/ -type f | head -n 1`; do a=$(dckey -k "AcquisitionDate" ${x}
2>&1); done; echo ${y} ${a}; done | grep -v Error | sed 's/_/_/g; s/.*-F/0/;
s/F//; s/ /;/' > /home/osotolongo/facehbi/dicom_fbb_v2_date.csv
```

Parece raro pero dado el poco consistente formato de los archivos hay que cambiar las ordenes para cada directorio.

Pegando

Las fechas quedan ahora en cuatro archivos,

```
[osotolongo@detritus facehbi]$ ls -l dicom_*
-rw-rw---- 1 osotolongo osotolongo 2800 May 20 10:06 dicom_fbb_v0_date.csv
-rw-rw---- 1 osotolongo osotolongo 2786 May 20 10:14 dicom_fbb_v2_date.csv
-rw-rw---- 1 osotolongo osotolongo 2884 May 18 14:59 dicom_mri_v0_date.csv
-rw-rw---- 1 osotolongo osotolongo 3262 May 20 10:02 dicom_mri_v2_date.csv
```

Voy a hacer un parser para juntar todo,

[date_parser.pl](#)

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

my %fdates = ( FBBv0 => "dicom_fbb_v0_date.csv",
              FBBv2 => "dicom_fbb_v2_date.csv",
              MRIv0 => "dicom_mri_v0_date.csv",
              MRIv2 => "dicom_mri_v2_date.csv",
              );

my $fdpath = '/home/osotolongo/facehbi/';
my %imgdates;

foreach my $fdate (sort keys %fdates){
    $real_file = $fdpath.$fdates{$fdate};
    open IDF, "<$real_file" or die "No such file";
    while(<IDF>){
        (my $subject, my $imdate) = /(.*);(.*)/;
        $imgdates{$subject}{$fdate} = $imdate;
    }
    close IDF;
}
print "Subject";
foreach my $fdate (sort keys %fdates){
    print ", $fdate";
}
print "\n";
foreach my $subject (sort keys %imgdates){
    print "$subject";
    foreach my $fdate (sort keys %fdates){
        if (exists($imgdates{$subject}{$fdate}))){
            print ", $imgdates{$subject}{$fdate}";
        }else{
            print ", -";
        }
    }
}
```

```

    }
}
print "\n";
}

```

y ahí va,

```

[osotolongo@detritus facehbi]$ ./date_parser.pl > dicom_dates.csv
[osotolongo@detritus facehbi]$ head dicom_dates.csv
Subject, FBBv0, FBBv2, MRIV0, MRIV2
0001, 20141211, 20170126, 20141205, 20170124
0002, 20141211, 20170420, 20141205, 20170323
0003, 20141218, 20170126, 20141211, 20170123
0004, 20141218, -, 20141212, -
0005, 20150122, 20170202, 20150107, 20170123
0006, 20150115, 20170126, 20141223, 20170124
0007, 20150115, 20170126, 20141219, 20170120
0008, 20150115, 20170202, 20141220, 20170125
0009, 20150129, 20170216, 20150110, 20170207

```

Reprocesamiento

Primero reconvertir los que están mal, borrar los directorios de FS, crearlos de nuevo y recalcular FS.

```

[osotolongo@detritus facehbi]$ awk -F"/" '{print $5}' mover_estos.txt | sed
's/F/0/;s/_.*/;smc/' > /nas/data/v2MriPet/repetir.csv
...
[osotolongo@detritus v2MriPet]$ awk -F";" '{print $1}' repetir.csv | sed
's/0/F/' > mri_repetir.dir
[osotolongo@detritus v2MriPet]$ for x in `cat mri_repetir.dir`; do for s in
/nas/corachan/facehbi_2/${x}/*; do if [[ `dckey -k "SeriesDescription"
${s}/Img00001.dcm 2>&1 | grep "t1_mprage_sag_p2_iso_1.0$" ` ]]; then dcm2niix
-z y -o /nas/data/v2MriPet/tmp ${s}; fi; done; mkdir
/nas/data/v2MriPet/processed/${x}; mv tmp/*
/nas/data/v2MriPet/processed/${x}/ ;done
[osotolongo@detritus v2MriPet]$ for a in `ls processed/F*/*.nii.gz`; do
b=$(echo $a | sed s'/processed\F/\nas\data\v2MriPet\mri\smc0/;
s/\Serie(/s000/; s)/.*/.nii.gz/'); mv ${a} ${b}; done
[osotolongo@detritus v2MriPet]$ for x in `awk -F";" '{print $1}'
repetir.csv`; do rm -rf /nas/data/subjects/v2MriPet_smc${x}; done
[osotolongo@detritus v2MriPet]$ fsl2fs.pl -cut repetir.csv v2MriPet
[osotolongo@detritus v2MriPet]$ precon.pl -cut repetir.csv v2MriPet
Submitted batch job 17319
[osotolongo@detritus v2MriPet]$ squeue
          JOBID PARTITION      NAME      USER ST       TIME  NODES
NODELIST(REASON)
          17319      devel fs_recon osotolon PD         0:00      1
(Dependency)

```

17293	devel	fs_recon	osotolon	R	0:04	1	brick03
17294	devel	fs_recon	osotolon	R	0:04	1	brick03
17295	devel	fs_recon	osotolon	R	0:04	1	brick03
17296	devel	fs_recon	osotolon	R	0:04	1	brick03
17297	devel	fs_recon	osotolon	R	0:04	1	brick03
17298	devel	fs_recon	osotolon	R	0:04	1	brick03
17299	devel	fs_recon	osotolon	R	0:04	1	brick03
17300	devel	fs_recon	osotolon	R	0:04	1	brick03
17301	devel	fs_recon	osotolon	R	0:04	1	brick03
17302	devel	fs_recon	osotolon	R	0:04	1	brick03
17303	devel	fs_recon	osotolon	R	0:04	1	brick03
17304	devel	fs_recon	osotolon	R	0:04	1	brick03
17305	devel	fs_recon	osotolon	R	0:04	1	brick03
17306	devel	fs_recon	osotolon	R	0:04	1	brick03
17307	devel	fs_recon	osotolon	R	0:04	1	brick03
17308	devel	fs_recon	osotolon	R	0:04	1	brick03
17309	devel	fs_recon	osotolon	R	0:04	1	brick03
17310	devel	fs_recon	osotolon	R	0:04	1	brick03
17311	devel	fs_recon	osotolon	R	0:04	1	brick03
17312	devel	fs_recon	osotolon	R	0:04	1	brick03
17313	devel	fs_recon	osotolon	R	0:04	1	brick03
17314	devel	fs_recon	osotolon	R	0:04	1	brick03
17315	devel	fs_recon	osotolon	R	0:04	1	brick03
17316	devel	fs_recon	osotolon	R	0:04	1	brick03
17317	devel	fs_recon	osotolon	R	0:04	1	brick03
17318	devel	fs_recon	osotolon	R	0:04	1	brick03

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

Permanent link:
<http://detritus.fundacioace.com/wiki/doku.php?id=neuroimagen:facehbi&rev=1559723494>

Last update: **2020/08/04 10:46**

