

Operaciones con XNAT

Uno de los objetivos de la nueva version del pipeline es permitir una interaccion con la [plataforma de XNAT de Fundacio ACE](#).

Aunque XNAT permite operar a traves de su interfaz web, tiene una potente REST API que facilita mucho la interacción para las tareas comunes. Para ello se ha implementado un cliente de CLI, **xnatapic**

Subir DICOMs

El primer paso debe ser definir los sujetos del proyecto,

```
[osotolongo@brick03 mopead]$ for x in /nas/corachan/MOPEAD_OSCAR_ZIP/*.zip;
do s=$(basename ${x} | sed 's/\.zip//'); xnatapic create_subject --
project_id facemopead --subject_id ${s}; done
```

Ahora, segun esa lista, podemos subir el DICOM correspondiente a cada uno,

```
[osotolongo@brick03 mopead]$ xnatapic list_subjects --project_id facemopead
--label > xnat_subjects.list
[osotolongo@brick03 mopead]$ for x in `awk -F"," '{print $2}'
xnat_subjects.list`; do xnatapic upload_dicom --project_id facemopead --
subject_id ${x} --pipelines --zip /nas/corachan/MOPEAD_OSCAR_ZIP/${x}.zip;
done
```

Para los PET, el procedimiento de subida es similar,

```
[osotolongo@brick03 facehbi]$ for x in `cat pets2xnat.list`; do s=$(echo
${x} | sed 's/FACEHBI-\(F[0-9]*\)B/1/'); xnatapic upload_dicom --project_id
facehbi --subject_id ${s} /nas/clinic/facehbi/${x}; done
```

XNAT automaticamente encuentra el tipo de imagen y la asigna al sujeto que señalamos,

PROJECT: facehbi > F001

Subject Details: F001

| Details | Projects |
|--|----------|
| Accession # XNAT4_S00159 | |
| Date Added 2020-12-04 11:25:23.355 (osotolongo) | |
| Birth year -- | |
| Gender | |
| Handedness | |

| Actions |
|-----------------------|
| Edit |
| View XML |
| Add Experiment |
| Download XML |
| Email |
| Manage Files |
| Delete |

Experiments

| Date | Experiment | Project | Label |
|------------|-----------------------------|-----------|---------|
| 2014-12-05 | MR Session | FACEHBIV0 | D415479 |
| 2014-12-11 | PET Session | FACEHBIV0 | F001B |

Ejecutar pipeline

Se puede obtener una lista de los pipelines activos en un proyecto,

```
[osotolongo@brick03 facehbi]$ xnatapic list_pipelines --project_id facehbi
RunFreesurfer
RegisterPETwithMRImatch
```

Freesurfer

Lo primero que se ha hecho con el XNAT es implementar el procesamiento de FS automatizado. Para ejecutar el procesamiento de FS, se ha de hacer algo como,

```
[osotolongo@detritus ~]$ xnatapic run_pipeline --project_id epad --pipeline
RunFreesurfer
```

PET

Primero encontramos los experimentos correctos,

```
[osotolongo@brick03 facehbi]$ xnatapic list_experiments --project_id facehbi
--type | grep petSessionData | awk -F"," {'print $1'} >
xnat_pet_experiments.list
[osotolongo@brick03 facehbi]$ head xnat_pet_experiments.list
XNAT5_E00276
XNAT5_E00278
```

```
XNAT5_E00283
XNAT5_E00296
XNAT5_E00299
XNAT5_E00308
XNAT5_E00316
XNAT5_E00318
XNAT5_E00319
XNAT5_E00323
```

Y ahora mandamos todo,

```
[osotolongo@brick03 facehbi]$ for x in `cat xnat_pet_experiments.list`; do
xnatapic run_pipeline --project_id facehbi --pipeline
RegisterPETwithMRImatch --experiment_id ${x}; done
```

Recuperar datos de Freesurfer

Puede recuperarse el directorio completo de reconstrucción de FS. Primero necesito saber sujeto de xnat y experimento correspondiente al proyecto.

```
[osotolongo@brick03 f5cehbi]$ xnatapic list_subjects --project_id f5cehbi --
label > xnat_subjects.list
[osotolongo@brick03 f5cehbi]$ for x in `awk -F"," {'print $1'}
xnat_subjects.list`; do e=$(xnatapic list_experiments --project_id f5cehbi -
-subject_id ${x} --modality MRI); if [[ ${e} ]]; then echo "${x},${e}"; fi;
done > xnat_subject_mri.list
```

Para bajar uno solo es bastante sencillo.

```
[osotolongo@brick03 f5cehbi]$ xnatapic get_fsresults --experiment_id
XNAT_E00112 --all-tgz tmp/
[osotolongo@brick03 f5cehbi]$ ls tmp/
D17193133.tar.gz
```

descomprimir uno de los sujetos es mas o menos asi,

```
[osotolongo@brick03 tmp]$ tar xzvf f5cehbi_0020/*.tar.gz -C
/nas/data/subjects/f5cehbi_0020/ --transform='s/XNAT5_S00278/' --
exclude='fsaverage'
```

Ahora, voy a sacar la correspondencia entre el proyecto, el pipeline y XNAT. Esto no es tan complicado como parece, hay, sobre todo que ordenar las cosas,

```
[osotolongo@brick03 f5cehbi]$ sed 's/;/,/g' f5cehbi_mri.csv > all_mri.list
[osotolongo@brick03 f5cehbi]$ sort -t, -k 2 all_mri.list >
all_mri_sorted.list
[osotolongo@brick03 f5cehbi]$ join -t, xnat_subjects.list
xnat_subject_mri.list > tmp_mri.list
```

```
[osotolongo@brick03 f5cehbi]$ sort -t, -k 2 tmp_mri.list >
xnat_tmp_mri_sorted.list
[osotolongo@brick03 f5cehbi]$ join -t, -j 2 all_mri_sorted.list
xnat_tmp_mri_sorted.list > xnat_guys.list
[osotolongo@brick03 f5cehbi]$ head xnat_guys.list
F001,0001,XNAT_S00085,XNAT_E00104
F005,0002,XNAT_S00086,XNAT_E00105
F006,0003,XNAT_S00087,XNAT_E00106
F007,0004,XNAT_S00088,XNAT_E00107
F009,0005,XNAT_S00089,XNAT_E00108
F010,0006,XNAT_S00090,XNAT_E00109
F013,0017,XNAT5_S00252,XNAT5_E01131
F014,0007,XNAT_S00091,XNAT_E00110
F015,0008,XNAT_S00092,XNAT_E00111
F023,0009,XNAT_S00093,XNAT_E00112
```

Ahora hacemos un script para que deje todo en su sitio,

[pullfs.pl](#)

```
#!/usr/bin/perl
#Copyright 2021 O. Sotolongo <asqwerty@gmail.com>

use strict;
use warnings;

my $prj = shift;
my $guide = shift;
my $tmpdir = $ENV{'PWD'}. "/tmp";
mkdir $tmpdir unless (-d $tmpdir);
open IDF, "<$guide" or die "No such file or directory\n";
while(<IDF>){
    my ($pid, $imgid, $xsubj, $xexp) = /(.*),(.*),(.*),(.*)/;
    my $fsdir = $ENV{'SUBJECTS_DIR'}. "/" . $prj . "_" . $imgid;
    unless ( -d $fsdir){
        my $tfsdir = $tmpdir. "/" . $prj . "_" . $imgid;
        mkdir $tfsdir;
        my $xorder = "xnatopic get_fsresults --experiment_id ".$xexp."
--all-tgz ".$tfsdir;
        print "$xorder\n";
        system($xorder);
        mkdir $fsdir;
        my $order = "tar xzvf ".$tfsdir."/*.tar.gz -C ".$fsdir."/ --
transform='s/' . $xsubj . '//\' --exclude='fsaverage\'";
        print "$order\n";
        system($order);
    }
}
my $order = "rm -rf ".$tmpdir;
```

```
system($order);
```

y al ejecutarlo,

```
[osotolongo@brick03 f5cehbi]$ ./pullfs.pl f5cehbi xnat_guys.list
```

Deberia quedar todo en su sitio,

```
[osotolongo@brick03 f5cehbi]$ ls -d /nas/data/subjects/f5cehbi*
/nas/data/subjects/f5cehbi_0001 /nas/data/subjects/f5cehbi_0007
/nas/data/subjects/f5cehbi_0013
/nas/data/subjects/f5cehbi_0002 /nas/data/subjects/f5cehbi_0008
/nas/data/subjects/f5cehbi_0014
/nas/data/subjects/f5cehbi_0003 /nas/data/subjects/f5cehbi_0009
/nas/data/subjects/f5cehbi_0015
/nas/data/subjects/f5cehbi_0004 /nas/data/subjects/f5cehbi_0010
/nas/data/subjects/f5cehbi_0017
/nas/data/subjects/f5cehbi_0005 /nas/data/subjects/f5cehbi_0011
/nas/data/subjects/f5cehbi_0018
/nas/data/subjects/f5cehbi_0006 /nas/data/subjects/f5cehbi_0012
```

Podemos hacerlo mejor que esto. Puede integrarse en el pipeline completamente, de manera que suministrando los dos nombres de proyecto (pipeline y xnat) se integren los datos

xnat_pullfs.pl

```
#!/usr/bin/perl
#Copyright 2021 O. Sotolongo <asqwerty@gmail.com>

use strict;
use warnings;
use NEURO4 qw(load_project print_help populate check_or_make);

my $prj;
my $xprj;
my $tmpdir;
chomp($tmpdir = `mktemp -d /tmp/fstar.XXXXXX`);
my $guide = $tmpdir."/xnat_guys.list";
@ARGV = ("-h") unless @ARGV;
while (@ARGV and $ARGV[0] =~ /^-/) {
    $_ = shift;
    last if /^--$/;
    if (/^-p/) { $prj = shift; chomp($prj);}
    if (/^-x/) { $xprj = shift; chomp($xprj);}
    if (/^-h/) { print_help $ENV{'PIPEDIR'}.'/doc/xnat_pull.hlp';
}
exit;}
}
$xprj = $prj unless $xprj;
my %std = load_project($prj);
```

```
my $order = "xnatopic list_subjects --project_id ".$xprj." --label >
".$std{'DATA'}."/xnat_subjects.list";
print "Getting XNAT subject list\n";
system($order);
$order = "for x in `awk -F\"\", \"\" {'print \$1'} xnat_subjects.list`; do
e=\$(xnatopic list_experiments --project_id ".$xprj." --subject_id
\${x} --modality MRI); if [[ \${e} ]]; then echo "\"\${x},\${e}\""; fi;
done > ".$std{'DATA'}."/xnat_subject_mri.list";
print "Getting experiments\n";
system($order);
my $proj_file = $std{'DATA'}.'/'."$prj."'_mri.csv';
$order = "sed 's/;/,/g' ".$proj_file." > ".$tmpdir."/all_mri.list;";
$order.= "sort -t, -k 2 ".$tmpdir."/all_mri.list >
".$tmpdir."/all_mri_sorted.list;";
$order.= "join -t, xnat_subjects.list xnat_subject_mri.list >
".$tmpdir."/tmp_mri.list;";
$order.= "sort -t, -k 2 ".$tmpdir."/tmp_mri.list >
".$tmpdir."/xnat_tmp_mri_sorted.list;";
$order.= "join -t, -j 2 ".$tmpdir."/all_mri_sorted.list
".$tmpdir."/xnat_tmp_mri_sorted.list > ".$tmpdir."/xnat_guys.list";
print "Sorting, joining, keeping shit together\n";
system($order);
print "OK, now I'm going on\nDownloading and extracting\n";
open IDF, "<$guide" or die "No such file or directory\n";
while(<IDF>){
    my ($pid, $imgid, $xsubj, $xexp) = /(.*),(.*),(.*),(.*)/;
    my $fsdir = $ENV{'SUBJECTS_DIR'}."/".$prj."_".$imgid;
    unless ( -d $fsdir){
        my $tfmdir = $tmpdir."/".$prj."_".$imgid;
        mkdir $tfmdir;
        my $xorder = "xnatopic get_fsresults --experiment_id
".$xexp." --all-tgz ".$tfmdir;
        print "$xorder\n";
        system($xorder);
        mkdir $fsdir;
        my $order = "tar xzvf ".$tfmdir."/*.*.tar.gz -C
".$fsdir."/ --transform='s/'."$xsubj."/'\\" --exclude='fsaverage'";
        print "$order\n";
        system($order);
    }
}
$order = "rm -rf ".$tmpdir;
system($order);
```

que se ejecuta como,

```
[osotolongo@brick03 f5cehbi]$ xnat_pullfs.pl -p f5cehbi
```

y quedan los directorios tal y como si se hubiera ejecutado el pipeline.

Obtener metricas

MRI

Obtener una metrica

Para obtener los resultados de la manera usual, debo organizarlos po sujeto y no por experimento, que es la manera *default* de XNAT. Empiezo por obtener una lista de los sujetos del proyecto.

```
[osotolongo@brick03 mopead]$ xnatapic list_subjects --project_id facemopead --label > xnat_subjects.list
```

Ahora puedo tener el experimento que quiero vincular con cada sujeto,

```
[osotolongo@brick03 mopead]$ for x in `awk -F"," '{print $2}' xnat_subjects.list`; do e=$(xnatapic list_experiments --project_id facemopead --subject_id ${x} --modality MRI --label); echo "${x},${e}"; done > xnat_sub_exp.list
```

Intento bajar ahora uno de los archivos de estadística,

```
[osotolongo@brick03 mopead]$ for l in `cat xnat_sub_exp.list`; do s=$(echo ${l} | awk -F"," '{print $1}'); e=$(echo ${l} | awk -F"," '{print $2}'); mkdir -p fsresults/${s}; xnatapic get_fsresults --experiment_id ${e} --stats aseg fsresults/${s}/; done
```

y ya puedo obtener los datos de los que se han procesados con éxito,

```
[osotolongo@brick03 mopead]$ ls fsresults/*/aseg.stats | head
fsresults/4NKAYEW6/aseg.stats
fsresults/4THMLQBG/aseg.stats
fsresults/52JAYALW/aseg.stats
fsresults/58H3I76A/aseg.stats
fsresults/5DBJ2MX6/aseg.stats
fsresults/6599YP5C/aseg.stats
fsresults/6HBBACWP/aseg.stats
fsresults/6WYXPIKZ/aseg.stats
fsresults/8BMXAD87/aseg.stats
```

Para sacar de aquí una métrica específica,

```
[osotolongo@brick03 mopead]$ for x in `ls fsresults/*/aseg.stats`; do s=$(echo ${x} | awk -F"/" '{print $2}'); v=$(grep "Left-Hippocampus" ${x} | awk '{print $4}'); echo "${s},${v}"; done | sed 'li"Subject,Left-Hippocampus"' > Left-Hippocampus.csv
[osotolongo@brick03 mopead]$ head Left-Hippocampus.csv
"Subject,Left-Hippocampus"
4NKAYEW6,3589.8
```

```
4THMLQBG,3312.0
52JAYALW,3326.9
58H3I76A,4106.6
5DBJ2MX6,3605.8
6599YP5C,2830.5
6HBBACWP,3470.4
6WYXPIKZ,4654.0
8BMXAD87,3592.4
```

Obtener una tabla de metricas

De esta estructura es bastante simple obtener una tabla con todas las metricas. Ejemplo, si queremos sacar el contenido de `aseg.stats` hacemos un script para reorganizar los datos.

[pulltable.pl](#)

```
#!/usr/bin/perl
#Copyright 2021 O. Sotolongo <asqwerty@gmail.com>

use strict;
use warnings;

my $rdir = shift;
my $stat = shift;
my @users = `ls $rdir/*/ $stat.stats | awk -F"/" '{print \$2}'`;
chomp @users;

my $okheader = 0;
print "Subject_ID";
foreach my $user (@users){
    my @tdata = `grep -v "^#" $rdir/$user/$stat.stats | awk '{print \$5",\"\$4}'`;
    chomp @tdata;
    my %udata = map { my ($key, $value) = split ","; $key => $value } @tdata;
    unless ($okheader) {
        foreach my $dhead (sort keys %udata){
            print ", $dhead";
        }
        $okheader = 1;
        print "\n";
    }
    print "$user";
    foreach my $roi (sort keys %udata){
        print ", $udata{$roi}";
    }
    print "\n";
}
```


y se lanza como,

```
[osotolongo@brick03 mopead]$ ./pulltable.pl fsresults aseg > aseg.table
```

Ejemplo aparcs.stats

```
[osotolongo@brick03 bioface]$ for l in `cat xnat_sub_exp.list`; do s=$(echo
${l} | awk -F"," '{print $1}'); e=$(echo ${l} | awk -F"," '{print $2}');
mkdir -p fsresults/${s}; xnataptic get_fsresults --experiment_id ${e} --stats
lh.aparc fsresults/${s}/; done
[osotolongo@brick03 bioface]$ for l in `cat xnat_sub_exp.list`; do s=$(echo
${l} | awk -F"," '{print $1}'); e=$(echo ${l} | awk -F"," '{print $2}');
mkdir -p fsresults/${s}; xnataptic get_fsresults --experiment_id ${e} --stats
rh.aparc fsresults/${s}/; done
[osotolongo@brick03 bioface]$ for x in `ls fsresults/*/lh.aparc.stats`; do
s=$(echo ${x} | awk -F"/" '{print $2}'); v=$(grep "fusiform" ${x} | awk
'{print $4}'); echo "${s},${v}"; done | sed
'li"Subject,lh.fusiform.GrayVol"' > lh.fusiform.grayvol.csv
[osotolongo@brick03 bioface]$ head lh.fusiform.grayvol.csv
"Subject,lh.fusiform.GrayVol"
B001,6569
B002,7204
B003,7244
B004,5333
B005,7753
B006,5848
B007,6483
B008,6447
B009,5765
```

PET

Los resultados del calculo PET son simples de extraer,

```
[osotolongo@brick03 f5cehbi]$ xnataptic get_registration_report --project_id
f5cehbi > centiloid_results.csv
```

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|----|--------------|---------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 1 | 010 1001P001 | 0100001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 2 | 010 1001P002 | 0100002 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 3 | 010 1001P003 | 0100003 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 4 | 010 1001P004 | 0100004 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 5 | 010 1001P005 | 0100005 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 6 | 010 1001P006 | 0100006 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 7 | 010 1001P007 | 0100007 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 8 | 010 1001P008 | 0100008 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 9 | 010 1001P009 | 0100009 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 10 | 010 1001P010 | 0100010 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 11 | 010 1001P011 | 0100011 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 12 | 010 1001P012 | 0100012 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 13 | 010 1001P013 | 0100013 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 14 | 010 1001P014 | 0100014 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 15 | 010 1001P015 | 0100015 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 16 | 010 1001P016 | 0100016 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 17 | 010 1001P017 | 0100017 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 18 | 010 1001P018 | 0100018 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 19 | 010 1001P019 | 0100019 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |
| 20 | 010 1001P020 | 0100020 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 | 01001 |

El archivo de resultados incluye un link por sujeto para realizar el QA. Tras revisar todos los sujetos se debería obtener nuevamente el archivo de resultados. Tras comprobar que todo es correcto, se pueden obtener las metricas en el formato de salida del pipeline,

Esto esta integrado en el pipeline,

xnat_pullcl.pl

```
#!/usr/bin/perl
#Copyright 2021 O. Sotolongo <asqwerty@gmail.com>

use strict;
use warnings;
use NEURO4 qw(load_project print_help populate check_or_make);

my $prj;
my $xprj;
my $tmpdir;
chomp($tmpdir = `mktemp -d /tmp/petcl.XXXXXX`);
my $guide = $tmpdir."/xnat_guys.list";
@ARGV = ("-h") unless @ARGV;
while (@ARGV and $ARGV[0] =~ /^-/) {
    $_ = shift;
    last if /^--$/;
    if (/^-p/) { $prj = shift; chomp($prj);}
    if (/^-x/) { $xprj = shift; chomp($xprj);}
    if (/^-h/) { print_help $ENV{'PIPEDIR'}.'/doc/xnat_pull.hlp';
}
exit;}
}
$xprj = $prj unless $xprj;
my %std = load_project($prj);
my $proj_file = $std{'DATA'}.'/'.'$prj.'_mri.csv';
print "Wait a minute, getting XNAT subject list\n";
my $order = "xnatopic list_subjects --project_id ".$xprj." --label >
".$std{'DATA'}."/xnat_subjects.list";
system($order);
print "Getting PET data now\n";
$order = "xnatopic get_registration_report --project_id ".$xprj." --
output ".$std{'DATA'}."/xnat_pet_results.csv";
$order.= "sort -t, -k 1 ".$std{'DATA'}."/xnat_subjects.list >
".$tmpdir."/xnat_subjects_sorted.list";
system($order);
print "Sorting, joining, keeping shit together\n";
$order = "awk -F\", \" '{print \$2\", \"\$6\", \"\$7}'
xnat_pet_results.csv | sed 's/\\/\\/g' | tail -n +2 | sort -t, -k 1 >
".$tmpdir."/pet.results";
system($order);
$order = "join -t, ".$std{'DATA'}."/xnat_subjects_sorted.list
".$tmpdir."/pet.results | sort -t, -k 2 >
".$tmpdir."/xnat_tmp_pet_sorted.list";
system($order);
$order = "sed 's/;/,/g' ".$proj_file." | sort -t, -k 2 >
".$tmpdir."/all_pet.list";
$order.= "join -t, -j 2 ".$tmpdir."/all_pet.list
```

```

"}.${tmpdir.}/xnat_tmp_pet_sorted.list | awk -F",\" \" '{print
\\$2\\";\\\"\\$4\\";\\\"\\$5}' | sed 'liSubject;SUVR;Centilod' >
"}.${std{'DATA'}.}/xnat_fbb_cl.csv";
system($order);
$order = "rm -rf "}.${tmpdir};
system($order);

```

```

[osotolongo@brick03 f5cehbi]$ xnat_pullcl.pl -p f5cehbi
[osotolongo@brick03 f5cehbi]$ head xnat_fbb_cl.csv
Subject;SUVR;Centilod
0001;1.0556667901822554;7.939285613957977
0002;0.9450962730154702;-9.022231719426872
0004;1.729791900514304;111.35007753889424
0005;1.1478956483545038;22.087192457580862
0006;0.9186037995919812;-13.086177142590072
0017;1.2627808309659585;39.71057947017803
0007;0.9619379523886946;-6.438718103574249
0008;0.9449992095030613;-9.037121262230404
0009;0.9422126618895114;-9.464577666148953

```

Troubleshooting

Sacando los PET a ejecutar

¿Que sujetos tengo en el proyecto?

```

[osotolongo@brick03 f5cehbi]$ xnatapic list_subjects --project_id f5cehbi --
label > xnat_subjects.list

```

¿De aqui, que sujetos tienen MRI?

```

[osotolongo@brick03 f5cehbi]$ for x in `awk -F"," {'print $1'}
xnat_subjects.list`; do e=$(xnatapic list_experiments --project_id f5cehbi -
-subject_id ${x} --modality MRI); if [[ ${e} ]]; then echo "${x},${e}"; fi;
done > xnat_subject_mri.list

```

¿De estos, cuales tiene PET?

```

[osotolongo@brick03 f5cehbi]$ for x in `awk -F"," {'print $1'}
xnat_subject_mri.list`; do e=$(xnatapic list_experiments --project_id
f5cehbi --subject_id ${x} --modality PET); if [[ ${e} ]]; then echo
"${x},${e}"; fi; done > xnat_subjects_mri_pet.list

```

¿De estos PET, cuantos ha sido calculados?

```

[osotolongo@brick03 f5cehbi]$ awk -F',' '{if($6!="") print $3}'

```

```
xnat_pet_results.csv | sed 's/"//g' | tail -n +2 > already_done.txt
```

Y por ultimo, ¿Cuales me falta por calcular?

```
[osotolongo@brick03 f5cehbi]$ grep -v "`cat already_done.txt`"  
xnat_subjects_mri_pet.list | awk -F',' '{print $2}' > pet2do.list
```

Vamos a hacerlos entonces!

```
[osotolongo@brick03 f5cehbi]$ for x in `cat pet2do.list`; do xnatapic  
run_pipeline --project_id f5cehbi --pipeline RegisterPETwithMRImatch --  
experiment_id ${x}; done  
[osotolongo@brick03 f5cehbi]$ queue
```

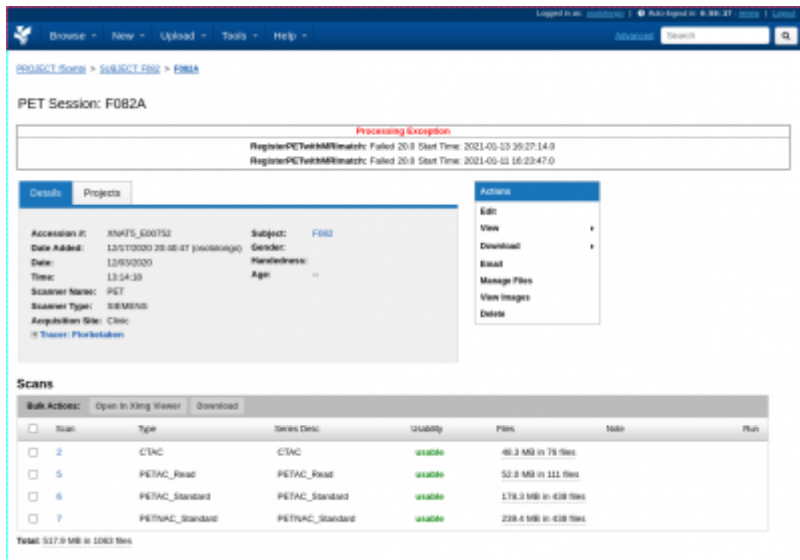
| JOBID | PARTITION | NAME | USER | ST |
|---------|-----------|--------------------------------------|------|----|
| 13027 | fast | RunFreesurfer.XNAT5_E00001 | xnat | R |
| 4:35:39 | 1 brick01 | | | |
| 13028 | fast | RegisterPETwithMRImatch.XNAT5_E00710 | xnat | R |
| 0:15 | 1 brick01 | | | |
| 13029 | fast | RegisterPETwithMRImatch.XNAT5_E00729 | xnat | R |
| 0:15 | 1 brick01 | | | |
| 13030 | fast | RegisterPETwithMRImatch.XNAT5_E00733 | xnat | R |
| 0:15 | 1 brick01 | | | |
| 13031 | fast | RegisterPETwithMRImatch.XNAT5_E00735 | xnat | R |
| 0:12 | 1 brick01 | | | |
| 13032 | fast | RegisterPETwithMRImatch.XNAT5_E00737 | xnat | R |
| 0:12 | 1 brick01 | | | |
| 13033 | fast | RegisterPETwithMRImatch.XNAT5_E00742 | xnat | R |
| 0:12 | 1 brick01 | | | |
| 13034 | fast | RegisterPETwithMRImatch.XNAT5_E00743 | xnat | R |
| 0:12 | 1 brick02 | | | |
| 13035 | fast | RegisterPETwithMRImatch.XNAT5_E00746 | xnat | R |
| 0:12 | 1 brick02 | | | |
| 13036 | fast | RegisterPETwithMRImatch.XNAT5_E00751 | xnat | R |
| 0:12 | 1 brick02 | | | |
| 13037 | fast | RegisterPETwithMRImatch.XNAT5_E00752 | xnat | R |
| 0:12 | 1 brick02 | | | |
| 13038 | fast | RegisterPETwithMRImatch.XNAT5_E00753 | xnat | R |
| 0:12 | 1 brick02 | | | |

PET con tag erroneo

Si el tag con que estan etiquetadas las imagenes es incorrecto, el registro de PET dara error. Típicamente recibes un email con un subject como:

```
XNAT update: Processing failed for F082A
```

Habria que revisar el sujeto, mirar como se ha etiquetado la imagen,



y relanzar algo como,

```
[osotolongo@brick03 f5cehbi]$ xnatapic run_pipeline --project_id f5cehbi --
pipeline RegisterPETwithMRImatch --experiment_id XNAT5_E00752 --dcmFBBtag
PETNAC_Standard
[osotolongo@brick03 f5cehbi]$ queue
JOBID PARTITION NAME USER ST
TIME NODES NODELIST(REASON)
13039 fast RegisterPETwithMRImatch.XNAT5_E00752 xnat R
0:03 1 brick01
```

failsafe PET

Un escenario que puede ocurrir es cuando el DICOM PET no se convierte correctamente con las herramientas automaticas. Es posible entonces, subir el PET en formato Nifti-1 y ejecutar un pipeline para que registre este ultimo archivo.

```
[osotolongo@brick03 f2cehbi]$ xnatapic upload_nifti --project_id f2cehbi --
subject_id F043 --experiment_id F043F --scan_id 5
/nas/clinic/nii/v2/F043/PET.nii PET.json
```

El archivo nifti debe ser el correspondiente al total de tiempo integrado,

```
[osotolongo@brick03 f2cehbi]$ fslinfo /nas/clinic/nii/v2/F043/PET.nii
data_type      FLOAT32
dim1           400
dim2           400
dim3           109
dim4           1
datatype       16
pixdim1        1.018210
pixdim2        1.018210
pixdim3        2.027008
pixdim4        0.000000
```

```
cal_max      0.0000
cal_min      0.0000
file_type    NIFTI-1+
```

El archivo json debe contener la info minima para identificar el protocolo correcto,

```
[osotolongo@brick03 f2cehbi]$ cat PET.json
{
  "SeriesDescription": "FACEHBI_Florbetaben_20min",
  "ProtocolName": "FACEHBI_Florbetaben_20min"
}
```

ambos archivos deben tener el mismo nombre.

Tras esto ya es posible lanzar el pipeline *RegisterPETwithMRI*,

```
[osotolongo@brick03 f2cehbi]$ xnatapic run_pipeline --project_id f2cehbi --
pipeline RegisterPETwithMRI --experiment_id F043F
```

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

Permanent link:
https://cortafuegos.fundacioace.com/wiki/doku.php?id=neuroimagen:xnat_operations

Last update: **2021/11/09 11:45**

