

CPAC

Una opción para preprocesar las imágenes fMRI es utilizar [C-PAC](#).

Se instala bajando una imagen [docker](#) que he transformado a [Singularity](#).

Sujetos individuales

Siguiendo [la documentacion](#) puede lanzarse como,

```
[osotolongo@brick01 mopead]$ singularity run --cleanenv -B
/nas/data/mopead/bids:/bids_dataset -B /nas/data/mopead/cpac_out:/outputs -B
/nas/data/mopead/tmp:/scratch /nas/software/cpac-latest.simg /bids_dataset
/outputs participant --participant_label sub-0001
```

Nota: el tag `-cleanenv` es necesario para que *singularity* no tome las variables del *bash_profile* de la máquina sino las de la imagen.

y por supuesto, la primera vez no funciona,

```
*****
190516-08:51:36,763 nipype.workflow ERROR:
    could not run node:
    resting_preproc_sub-0001_ses-1.anat_preproc_afni_0.anat_skullstrip
190516-08:51:36,765 nipype.workflow INFO:
    crashfile: /outputs/crash/crash-20190516-084320-osotolongo-
    anat_skullstrip-28dd4540-f205-44b7-9ad2-c4af06706a15.pklz
190516-08:51:36,769 nipype.workflow INFO:
    *****
```

Puedo cambiar el *skullstrip* para hacerlo con FSL, pero he de especificarlo en la configuración del pipeline. Así que me copio un archivo previo y lo cambio.

```
# Choice of using AFNI or FSL-BET to perform SkullStripping
skullstrip_option: [BET]
```

La orden ahora es ligeramente diferente pues he de especificar el archivo de configuración del pipeline,

```
[osotolongo@brick01 mopead]$ singularity run --cleanenv -B
/nas/data/mopead:/project -B /nas/data/mopead/bids:/bids_dataset -B
/nas/data/mopead/cpac_out:/outputs -B /nas/data/mopead/tmp:/scratch
/nas/software/cpac-latest.simg --pipeline_file
/project/cpac_pipeline_config.yml /bids_dataset /outputs participant --
participant_label sub-0002
```

183 minutos después,

End of subject workflow resting_preproc_sub-0002_ses-1

CPAC run complete:

```
    Pipeline configuration: analysis
    Subject workflow: resting_preproc_sub-0002_ses-1
    Elapsed run time (minutes): 184.085074282
    Timing information saved in
    /outputs/log/cpac_individual_timing_analysis.csv
    System time of start:      2019-05-16 09:17:52
    System time of completion: 2019-05-16 12:21:49
```

El output es enorme,

```
[osotolongo@brick01 mopead]$ ls
cpac_out/output/pipeline_analysis_nuisance/sub-0002_ses-1/
alff_to_standard_smooth_zstd
frame_wise_displacement_power
anatomical_brain                                functional_brain_mask
anatomical_csf_mask
functional_brain_mask_to_standard
anatomical_gm_mask                              functional_freq_filtered
anatomical_reorient
functional_nuisance_regressors
anatomical_to_mni_nonlinear_xfm
functional_to_anat_linear_xfm
anatomical_to_standard                          functional_to_standard
anatomical_to_symmetric_mni_nonlinear_xfm      mean_functional_to_standard
anatomical_wm_mask
mni_to_anatomical_nonlinear_xfm
ants_affine_xfm                                 motion_correct
ants_initial_xfm                               motion_params
ants_rigid_xfm                                 qc
ants_symmetric_affine_xfm                      qc_html
ants_symmetric_initial_xfm                    roi_timeseries
ants_symmetric_rigid_xfm                      spatial_map_timeseries
centrality_smooth_zstd
spatial_map_timeseries_for_DR
dr_tempreg_maps_files_to_standard_smooth
symmetric_anatomical_to_standard
dr_tempreg_maps_zstat_files_to_standard_smooth
symmetric_mni_to_anatomical_nonlinear_xfm
falff_to_standard_smooth_zstd                  vmhc_fisher_zstd_zstat_map
frame_wise_displacement_jenkinson
```

Afortunadamente, [la explicacion de cada directorio esta documentada](#).

Integrando en el cluster

El esquema para correr en el cluster es el de lanzar las imagenes de singularity en paralelo. Hay que hacer varias pruebas por si se solapan unas a otras pero en principio intentaremos lanzar unos 8 procesos por nodo (o menos).

[Aqui el codigo](#)

cpac.pl

```
#!/usr/bin/perl
# Copyright 2019 O. Sotolongo <asqwerty@gmail.com>
use strict; use warnings;

use File::Find::Rule;
use NEURO qw(print_help get_pair load_study achtung shit_done get_lut
check_or_make centiloid_fbb);
use Data::Dump qw(dump);
use File::Remove 'remove';
use File::Basename qw(basename);

my $cpac_img = '/nas/software/cpac-latest.simg';
my $pipe_conf = 'cpac_pipeline_config.yml';
my $lib_conf = $ENV{'PIPEDIR'}.'/lib/'. $pipe_conf;
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/cpac.hlp'; exit;}
}
my $study = shift;
unless ($study) { print_help $ENV{'PIPEDIR'}.'/doc/cpac.hlp'; exit;}
my %std = load_study($study);
my $w_dir = $std{'WORKING'};
my $data_dir = $std{'DATA'};
my $bids_dir = $data_dir.'/bids';
my $fmriout_dir = $data_dir.'/cpac_out';
check_or_make($fmriout_dir);
my $outdir = "$std{'DATA'}/slurm";
check_or_make($outdir);
my $tmpdir = "$std{'DATA'}/ctmp";
check_or_make($tmpdir);
my $proj_conf = $data_dir.'/'. $pipe_conf;
system("cp $lib_conf $proj_conf") unless (-e $proj_conf);
my @subjects;
if($cfile){
    open DBF, $cfile or die "No such file\n";
    while(<DBF>) {
```

```

        chomp;
        push @subjects, $_;
    }
    close DBF;
}
else{
    opendir DBD, $bids_dir or die "Cold not open dir\n";
    while (my $thing = readdir DBD){
        if ($thing eq '.' or $thing eq '..') {
            next;
        }
        if ($thing =~ /sub-*/) {
            push @subjects, $thing;
        }
    }
    closedir DBD;
}
foreach my $subject (@subjects) {
    my $orderfile = $outdir.'/'.$subject.'_cpac.sh';
    open ORD, ">$orderfile";
    print ORD '#!/bin/bash'\n";
    print ORD '#SBATCH -J cpac_'. $study.\n";
    print ORD '#SBATCH --time=72:0:0'\n"; #si no ha terminado en X
horas matalo
    print ORD '#SBATCH --mail-type=FAIL,TIME_LIMIT,STAGE_OUT'\n"; #no
quieres que te mande email de todo
    print ORD '#SBATCH -o '.$outdir.'/cpac-%j'\n";
    print ORD '#SBATCH -c 20'\n";
    print ORD '#SBATCH -p fast'\n";
    print ORD '#SBATCH --mail-user='.$ENV{'USER'}\n";
    print ORD 'srun singularity run --cleanenv -B
'.$data_dir.':/project -B '.$bids_dir.':/bids_dataset -B
'.$fmriout_dir.':/outputs -B '.$tmpdir.':/scratch '.$cpac_img.' --
pipeline_file /project/'.$pipe_conf.' /bids_dataset /outputs
participant --participant_label '.$subject.\n";
    close ORD;
    system("sbatch $orderfile");
    sleep(20);
}
my $orderfile = $outdir.'/cpac_end.sh';
open ORD, ">$orderfile";
print ORD '#!/bin/bash'\n";
print ORD '#SBATCH -J cpac_'. $study.\n";
print ORD '#SBATCH --mail-type=END'\n"; #email cuando termine
print ORD '#SBATCH --mail-user='.$ENV{'USER'}\n";
print ORD '#SBATCH -p fast'\n";
print ORD '#SBATCH -o '.$outdir.'/cpac_end-%j'\n";
print ORD ":\n";
close ORD;
my $order = 'sbatch --dependency=singleton '.$orderfile;
exec($order);

```

Nota: Al correr *c-pac* en el cluster dan errores de I/O. Esto puede ser debido a un intenso uso de la red o de disco. He de reducir el nuenmro de lanzamientos concurrentes y probar de nuevo.

Output

C-PAC genera un directorio de output por cada sujeto analizado

```
[osotolongo@detritus mopead]$ ls cpac_out/output/pipeline_analysis_nuisance/  
sub-0001_ses-1  sub-0003_ses-1  sub-0005_ses-1  sub-0007_ses-1  
sub-0009_ses-1  sub-0012_ses-1  sub-0015_ses-1  sub-0018_ses-1  
sub-0020_ses-1  sub-0022_ses-1  
sub-0002_ses-1  sub-0004_ses-1  sub-0006_ses-1  sub-0008_ses-1  
sub-0010_ses-1  sub-0014_ses-1  sub-0017_ses-1  sub-0019_ses-1  
sub-0021_ses-1  sub-0023_ses-1
```

El output completo para cada sujeto se puede ver [aquí](#).

Errores

```
190529-07:56:45,394 nipype.workflow INFO:  
    [Node] Setting-up "resting_preproc_sub-0004_ses-1.sinker_0_13" in  
    "/scratch/working/resting_preproc_sub-0004_ses-1/sinker_0_13".  
190529-07:56:46,61 nipype.workflow INFO:  
    [Node] Running "sinker_0_13" ("nipype.interfaces.io.DataSink")  
190529-07:56:46,617 nipype.workflow INFO:  
    [Node] Finished "resting_preproc_sub-0004_ses-1.sinker_0_13".  
srun: error: brick01: task 0: Exited with exit code 1
```

Grupos

Hay diferentes niveles de analisis grupal, explicados en http://fcp-indi.github.io/docs/user/group_analysis.html. Primero ha de elegirse lo que se desee hacer.

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

Permanent link:
<http://detritus.fundacioace.com/wiki/doku.php?id=neuroimagen:cpac&rev=1559206907>

Last update: **2020/08/04 10:45**

