

brainageR

He de ejecutar [brainageR](#) en todos las MRI que tengo y sacar los resultados. Este proyecto pretende determinar la edad del sujeto a través del envejecimiento del cerebro. O al menos la edad cerebral.

Básicamente esto es bajarlo, ejecutarlo y ya está. No obstante hay algunos detalles a tener en cuenta. El proyecto depende entre otras cosas de poder ejecutar MATLAB. Yo tengo una sola licencia individual de MATLAB, así que no voy a poder hacer una ejecución como desearía, usando el cluster HPC. Pero lo que si se puede es ejecutarlo en serie sobre todos los sujetos que tengo.

Ejecución

Hay algunas cosas que si tenemos a favor.

1. Todas las MRI estan incluidas en proyectos por lo que podemos utilizar la estructura del pipeline para ejecutar las tareas
2. Todos los proyecto utilizan el formato BIDS, por lo que la localización de los T1w individuales es extremadamente sencilla

Así que voy a planificar la ejecución por proyecto y ya después tendré que unirlos en un solo archivo de resultados.

Para empezar sigo las instrucciones de instalación y configuración. Me bajo todo y lo coloco tal y como se indica en las instrucciones y edito el archivo `software/brainageR` con la localización correcta de mi software,

```
## set software directories for your local system
brainageR_dir=/home/osotolongobrainageR
spm_dir=/nas/software/spm/spm12/
matlab_path=/usr/local/bin/matlab
FSLDIR=/usr/local/fsl/
```

Nota:

La única llamada a MATLAB esta en la línea,

```
### SPM12 segmentation and normalisation
cmd="$matlab_path -nosplash -nodesktop -r
\"addpath(['$brainageR_dir', '/software/']);spm_preprocess_brainageR('$full_p
ath_file', ['$brainageR_dir', '/software/templates/'], '$spm_dir');exit\"
eval ${cmd}
```

así que probablemente pueda cambiarse y utilizar SPM runtime. 

Ahora, se ha de ejectuar,

```
$ software/brainagerR -f T1w.nii -o output.csv
```

Importante aqui: El script no admite archivos compactados, así que antes que nada ha de tomarse el T1w de la estructura BIDS y descomprimirse a otro sitio.

Lo que hare es un script Perl que haga un directorio *brainage* para el proyecto, recorra todos los BIDS del proyecto, seleccione el T1w, lo descomprima al directorio creado y ejecute la orden para cada sujeto.

Algo así,

```
#!/usr/bin/perl

use strict;
use warnings;

use NEUR04 qw(load_project check_or_make get_subjects check_subj);
my $brainage = '/nas/osotolongo/brainagerR/software/brainagerR';
my $proj = shift;
die "Should supply project name" unless $proj;
my %std = load_project $proj;
my $wdir = $std{'DATA'}. '/brainage';
check_or_make $wdir;
my $proj_file = $std{'DATA'}. '/' . $proj . '_mri.csv';
my @guys = get_subjects($proj_file);
#dump sort @guys;
foreach my $subject (@guys){
    my $sbj = $proj . "_" . $subject;
    my %nifti = check_subj($std{'DATA'}, $subject);
    if($nifti{'T1w'}){
        my $sdir = $wdir. '/' . $sbj;
        check_or_make($sdir);
        my $nout = $sdir. '/' . $sbj . '.nii';
        my $xrd = 'gunzip -c ' . $nifti{'T1w'}[0]. ' > ' . $nout;
        print "$xrd\n";
        system($xrd);
        my $order = $brainage. ' -f ' . $nout. ' -o ' . $sdir. '/' . $sbj . '_age.csv';
        print "$order\n";
        system($order);
    }
}
```

Esto ejecutara para cada sujeto algo como.

```
gunzip -c /nas/data/epad/bids/sub-0077/anat/sub-0077_T1w.nii.gz >
/nas/data/epad/brainage/epad_0077/epad_0077.nii
/nas/osotolongo/brainagerR/software/brainagerR -f
/nas/data/epad/brainage/epad_0077/epad_0077.nii -o
```

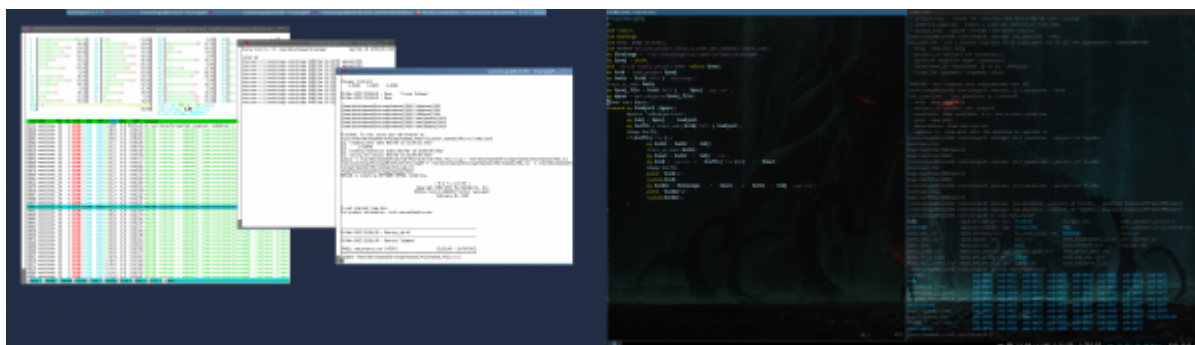
```
/nas/data/epad/brainage/epad_0077/epad_0077_age.csv
```

De momento he puesto los *prints* para saber que está haciendo. Una vez que se vea que todo va OK se pueden quitar. O no.

entonces bastara hacer algo como,

```
$ ./getage.pl epad
```

y tendremos un escenario como este,



Nota: No es necesario hacerlo con X, pero he querido por esta vez para controlar el funcionamiento del MATLAB y tener la sesión activa durante días. Podría hacer uso *screen* para lo mismo pero suelo olvidar las terminales en ejecuciones tan largas.

Al final tendremos un output similar a,

```
[osotolongo@brick03 brainageR]$ ls /nas/data/epad/brainage/
epad_0001  epad_0009  epad_0017  epad_0025  epad_0033  epad_0041  epad_0050
epad_0058  epad_0066  epad_0074
epad_0002  epad_0010  epad_0018  epad_0026  epad_0034  epad_0042  epad_0051
epad_0059  epad_0067  epad_0075
epad_0003  epad_0011  epad_0019  epad_0027  epad_0035  epad_0043  epad_0052
epad_0060  epad_0068  epad_0076
epad_0004  epad_0012  epad_0020  epad_0028  epad_0036  epad_0044  epad_0053
epad_0061  epad_0069  epad_0077
epad_0005  epad_0013  epad_0021  epad_0029  epad_0037  epad_0046  epad_0054
epad_0062  epad_0070  epad_0078
epad_0006  epad_0014  epad_0022  epad_0030  epad_0038  epad_0047  epad_0055
epad_0063  epad_0071
epad_0007  epad_0015  epad_0023  epad_0031  epad_0039  epad_0048  epad_0056
epad_0064  epad_0072
epad_0008  epad_0016  epad_0024  epad_0032  epad_0040  epad_0049  epad_0057
epad_0065  epad_0073
[osotolongo@brick03 brainageR]$ tree /nas/data/epad/brainage/epad_0001/
/nas/data/epad/brainage/epad_0001/
├── epad_0001_age.csv
├── epad_0001.nii
├── epad_0001_tissue_volumes.csv
└── slicesdir_epad_0001.nii
    ├── _home_data_epad_brainage_epad_0001__c1epad_0001.png
    └── _home_data_epad_brainage_epad_0001__c2epad_0001.png
```

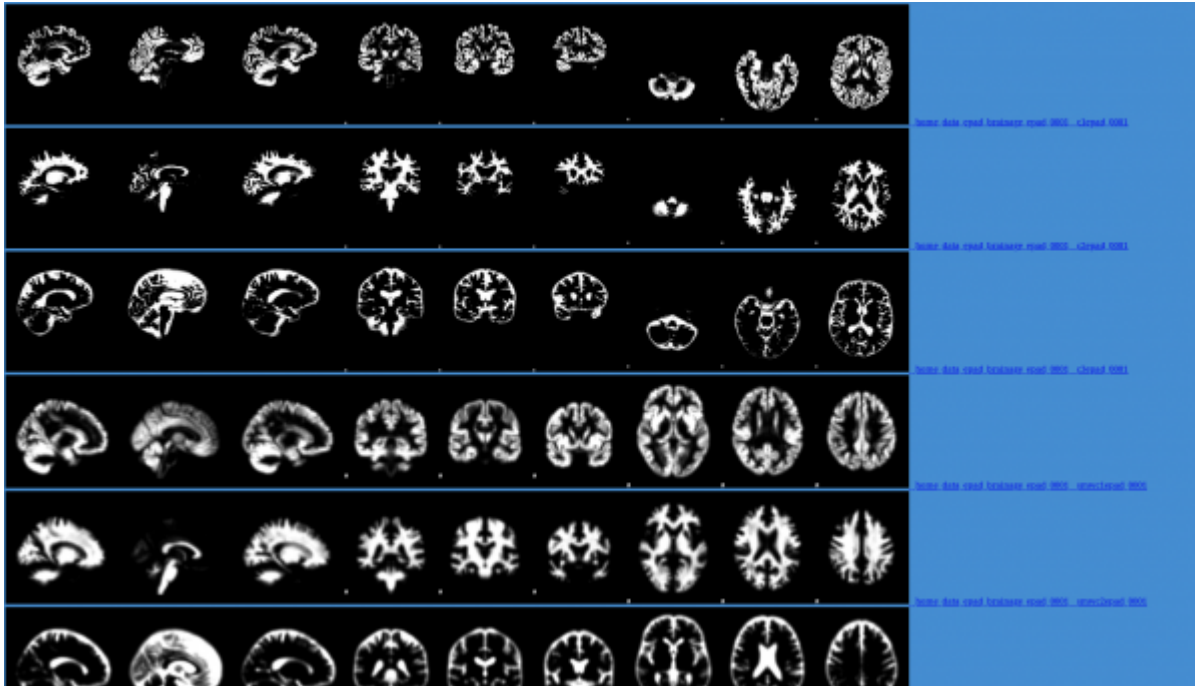
```

├── _home_data_epad_brainage_epad_0001__c3epad_0001.png
├── _home_data_epad_brainage_epad_0001__smwc1epad_0001.png
├── _home_data_epad_brainage_epad_0001__smwc2epad_0001.png
├── _home_data_epad_brainage_epad_0001__smwc3epad_0001.png
└── index.html

```

1 directory, 10 files

Incluso deja unos slices de la segmentación de WM para hacer el QC,



DB de los resultados

Sacar los resultados debería ser sencillo. Primero los headers de la DB,

```

[osotolongo@brick03 brainager]$ echo
"File,brain.predicted_age,lower.CI,upper.CI" > header_ages.csv

```

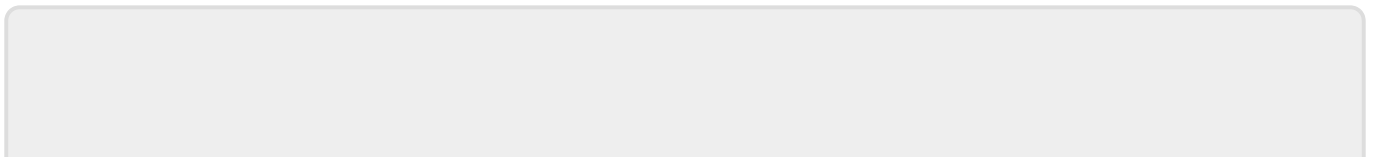
Ahora para cada proyecto,

```

[osotolongo@brick03 brainager]$ for x in /nas/data/epad/brainage/*; do
n=$(echo ${x} | awk -F"/" '{print $(NF)}'); l=$(grep -v "File"
${x}/${n}_age.csv); echo "${l}"; done > epad_ages.csv

```

Al final bastaría pegar todos los archivos para tener los datos agrupados,



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

Permanent link:
<https://imagen.fundacioace.com/wiki/doku.php?id=neuroimagen:brainager>

Last update: **2022/03/29 09:23**

