

ACE Pipeline (v0.3.0)

De momento, el procesamiento MRI, rs/fMRI y DTI esta integrado en el cluster. El procesamiento de las imagenes PET no lo esta.

Creando el proyecto

Para comenzar a trabajar es necesario crear la estructura del proyecto, que debe ser fija. Para ello existe el script *make_std.pl* que toma como argumento el nombre del proyecto y crea un sencillo archivo de configuracion, así como un par de directorios necesarios.

Ejemplo,

```
$ make_std.pl mopead
```

crea el archivo *\$HOME/.config/neuro/mopead.cfg*, con la estructura del proyecto,

```
$ cat ~/.config/neuro/mopead.cfg
DATA = /nas/osotolongo/data/mopead
DATABASE = /nas/osotolongo/data/mopead/mopead.csv
WORKING = /nas/osotolongo/data/mopead/working
NIFTI = /nas/osotolongo/data/mopead/nifti
MRI = /nas/osotolongo/data/mopead/mri
fMRI = /nas/osotolongo/data/mopead/fmri
DTI = /nas/osotolongo/data/mopead/dti
PET-FDG = /nas/osotolongo/data/mopead/pet-fdg
PET-PIB = /nas/osotolongo/data/mopead/pet-pib
PET-FBB = /nas/osotolongo/data/mopead/fbb
FBB-NC = /nas/osotolongo/data/mopead/fbb_first
SPECT = /nas/osotolongo/data/mopead/spect
```

Este archivo lo leen todos los demas scripts, así que si es necesario mover algun directorio de sitio, basta con cambiar este archivo para que los scripts busquen la imagenes en el nuevo directorio.

Fuera de crear el directorio del proyecto y el directorio *working*, el script no hace mas nada. Hay que crear la base de datos del proyecto, que debe tener esta forma,

```
$ cat mopead.csv
0001;mop
0002;mop
0003;mop
0004;mop
0005;mop
0006;mop
0007;mop
0008;mop
0009;mop
```

```
0010;mop  
0011;mop  
0012;mop  
0013;mop  
0014;mop  
0015;mop  
0016;mop  
0017;mop  
0018;mop  
0019;mop  
0020;mop  
0021;mop  
0022;mop  
0023;mop
```

Esta no es más que una cadena numerica de entre 4 y 6 numeros (no tiene por que ser consecutiva ni estar ordenada) y una cadena de caracteres de entre 1 y 4 caracteres (no tienen que ser iguales, puede usarse para diferenciar grupos), separados por un punto y coma (;). *Cuatro numeros y tres caracteres es un buen compromiso.*

MRI

La organizacion de las tareas de segmentacion se realizo con el proyecto MOPEAD. Ver [la pagina del proyecto](#) para mas info.

Los archivos de imagen *T1* deben estar almacenados en el directorio *mri* dentro del directorio del proyecto y en formato NIfTI-1. Deben ademas ser nombrados de acuerdo a la base de datos del proyecto. El archivo *mop0001sXXXX.nii.gz* (las X son cualquier numero) es el correspondiente a la linea *0001;mop* de la base de datos.

El script *fsl2fs.pl* crea la estructura de *freesurfer* del sujeto y convierte los archivos *T1* en el formato y estructura necesarios para ejecutar la segmentacion de *freesurfer*. Envía un email cuando termina. Las opciones son,

- *-a*, adjunta el archivo log al email
- *-cut*, para hacer solo un subconjunto de los archivos. Ver mas abajo en [Opciones comunes](#).

Así,

```
$ fsl2fs.pl mopead
```

crea la estructura y convierte todos los T1 en *mopead/mri* hacia *\$SUBJECTS_DIR/mopead_mopXXXX/mri/orig/XXX.mgz*

```
$ ls /nas/data/mopead/mri  
mop0001s0005.nii.gz  mop0007s0005.nii.gz  mop0013s0005.nii.gz  
mop0019s0005.nii.gz  
mop0002s0005.nii.gz  mop0008s0005.nii.gz  mop0014s0005.nii.gz  
mop0020s0005.nii.gz
```

```
mop0003s0005.nii.gz mop0009s0005.nii.gz mop0015s0005.nii.gz
mop0021s0005.nii.gz
mop0004s0005.nii.gz mop0010s0005.nii.gz mop0016s0005.nii.gz
mop0022s0005.nii.gz
mop0005s0005.nii.gz mop0011s0005.nii.gz mop0017s0005.nii.gz
mop0023s0005.nii.gz
mop0006s0005.nii.gz mop0012s0005.nii.gz mop0018s0005.nii.gz
```

```
$ ls -d /nas/data/subjects/mopead_*
/nas/data/subjects/mopead_mop0001 /nas/data/subjects/mopead_mop0013
/nas/data/subjects/mopead_mop0002 /nas/data/subjects/mopead_mop0014
/nas/data/subjects/mopead_mop0003 /nas/data/subjects/mopead_mop0015
/nas/data/subjects/mopead_mop0004 /nas/data/subjects/mopead_mop0016
/nas/data/subjects/mopead_mop0005 /nas/data/subjects/mopead_mop0017
/nas/data/subjects/mopead_mop0006 /nas/data/subjects/mopead_mop0018
/nas/data/subjects/mopead_mop0007 /nas/data/subjects/mopead_mop0019
/nas/data/subjects/mopead_mop0008 /nas/data/subjects/mopead_mop0020
/nas/data/subjects/mopead_mop0009 /nas/data/subjects/mopead_mop0021
/nas/data/subjects/mopead_mop0010 /nas/data/subjects/mopead_mop0022
/nas/data/subjects/mopead_mop0011
```

```
$ ls /nas/data/subjects/mopead_mop0001/mri/orig/
001.mgz
```

Ya la estructura esta preparada para lanzar la segmentación. Para esto es el script *precon.pl*, con las opciones,

- *-a*, adjunta el archivo log al email
- *-cut*, para hacer solo un subconjunto de los sujetos. Ver [Opciones comunes](#).

La orden,

```
$ precon.pl mopead
```

lanza la segmentacion de freesurfer al *cluster*, repartiendo las tareas en caso necesario entre las distintas maquinas. El script lanza todos los procesos en paralelo bajo la misma tarea por lo que si hay algún problema con la segmentación la tarea seguirá indefinidamente. Si se prolonga demasiado habra que matar la tarea. Para todo esto es necesario utilizar las herramientas de *slurm*. Ver mas abajo en [Opciones comunes](#) y revisar [Como usar el cluster sin morir en el intento](#).

Para comprobar si hay algun error se lanza otro script,

```
$ check_fs_run.pl mopead
```

Este lee los logs de *freesurfer*, busca los errores e informa los sujetos que estan mal. Ojo, que si hubo un error previo, hay que borrar los logs antes de relanzar la segmentacion.

Tras esto solo queda sacar las metricas,

```
$ mri_metrics.pl mopead
```

que devuelve el archivo *mopead_mri.csv* con los resultados.

DTI

Para la redefinición del cálculo DTI se ha utilizado el proyecto MOPEAD. Ver la [pagina de este proyecto](#) para más info sobre como funcionan las herramientas.

Preprocesamiento

Los archivos de *DTI* deben estar almacenados en el directorio *dti* dentro del directorio del proyecto y en formato NIFTI-1. Deben además ser nombrados de acuerdo a la base de datos del proyecto. El archivo *mop0001sXXXX.nii.gz* (las X son cualquier número) es el correspondiente a la línea *0001;mop* de la base de datos. Deben existir además los archivos *mop0001sXXXX.bval* y *mop0001sXXXX.bvec*. En caso de tenerlo, el archivo *mop0001_p2a_b0.nii.gz*, permite hacer corrección de susceptibilidad.

De cara al cluster, hay dos maneras de ejecutar el procesamiento DTI. Los scripts *dti_reg.pl* y *dti_reg_split.pl* hacen exactamente lo mismo pero mientras el primero lanza todas las tareas en el mismo trabajo, el segundo lanza una tarea por trabajo permitiendo un control individual del proceso de cada sujeto a cambio de velocidad de procesamiento. Dependiendo de las opciones, la cantidad de sujetos y los fallos que se obtenga en el proceso es preferible uno o el otro.

Las opciones son,

- *-cut*, (Ver [Opciones comunes](#))
- *-nocorr*, no realiza correcciones (para cuando no existe el $P \gg A$)
- *-legacy*, en lugar de la versión moderna de *eddy* utiliza *eddy_correct* y es útil para los casos en que falla el procesamiento.

Por ejemplo,

```
$ dti_reg_split -nocorr -cut soloestos.csv facehbi
```

procesara los sujetos contenidos en el archivo *soloestos.csv* que pertenezcan al proyecto *facehbi* sin realizar correcciones por susceptibilidad.

Los valores de FA y MD se obtienen ejecutando *dti_metrics.pl*. Este script calcula, y envía la tabla por email, estos valores en las regiones de referencia definidos por los [atlas de la JHU](#).

```
$ dti_metrics.pl facehbi
```

Por defecto utiliza el *JHU white-matter tractography atlas*. Las opciones son,

- *-l*, no envía la tabla por email
- *-a1*, usa el *ICBM-DTI-81 white-matter labels atlas*
- *-a2*, usa el *JHU white-matter tractography atlas*
- *-sd*, incluye también las desviaciones standard

Tractografía

Para la tractografía se utiliza el script `dti_track.pl`. Este script ejecuta `bedpostx` y `probtrackx` sobre los sujetos del proyecto. Para ejecutar este script es necesario crear el archivo `dti_tracks.seed` en el directorio del proyecto. Este archivo contiene las ROI que se utilizarán como máscara para `probtrackx`. Por defecto se utiliza la segmentación de `freesurfer` y las ROI se indican por el índice que se puede ver en `FreeSurferColorLUT.txt`.

Las opciones son,

- `-cut` (Ver [Opciones comunes](#))
- `-slow`: ejecuta `bedpost` sin utilizar el cluster (**desaconsejada**)
- `-uofm`: utiliza el [atlas de la UofM](#), en lugar de la segmentación de `freesurfer`
- `-s2t`: ejecuta `probtrackx` con clasificación de targets. Necesita los archivos `dti_track.seed` y `dti_track.targets`

Por ejemplo para ejecutar el script con los hipocampos como máscara,

```
$ cat dti_track.seed
17
53

$ dti_track.pl mopead
```

Para ejecutar esto mismo en el *Default Mode Network dorsal*, definido por el [atlas de la UofM](#),

```
$ dti_track.pl -uofm DMN_dorsal mopead
```

Para hacer un *seed to targets* del nodo 1 de la *Laguage Network*, de este atlas, a los demás nodos,

```
$ cat dti_track.seed
mri_LN1.nii

$ cat dti_track.targets
mri_LN2.nii
mri_LN3.nii
mri_LN4.nii
mri_LN5.nii
mri_LN6.nii
mri_LN7.nii

$ dti_track.pl -s2t -uofm LN mopead
```

Métricas en tractos escogidos

Una vez hecha la tractografía tenemos un mapa de determinado grupo de tractos sobre los que podemos sacar, para cada sujeto, los valores medios de FA y MD. Esto lo podemos hacer con el script `dti_metrics_alt.pl`. Este script busca los archivos `output` de la tractografía, y hace una máscara por encima de un valor determinado (por defecto 25% del valor máximo). Luego, calcula los valores de FA

y MD usando esta mascara. Esto se hace para cada sujeto individualmente.

El procedimiento depende de que tracto hayamos procesado. Por ejemplo,

```
$ dti_metrics_alt.pl -path DMN facehbi
```

buscara los archivos *output* de *probtrackx* en *working/sujeto_probtrack_DMN* y dejara las metricas en *facehbi_dti_DMN.csv*.

La forma de proceder es, primero realizar la tractografia con el atlas y la red deseada,

```
dti_track.pl -uofm DMN facehbi
```

luego mover los directorios al *path* correcto (esto se puede hacer automaticamente pero es un paso sencillo y para mantener la generalidad es mejor dejarlo asi por ahora),

```
for x in `ls -d working/*_probtrack_out`; do mv $x `echo $x | sed 's/out/DMN/'`;done
```

y finalmente sacar las metricas,

```
dti_metrics_alt.pl -path DMN facehbi
```

Nota: El paso 2 puede obviarse y el script por defecto buscara la tractografia en *working/sujeto_probtrack_out* y dejara las metricas en *facehbi_dti_out.csv*. Pero por cuestiones de organizacion no es recomendable.

Las opciones son,

- *-cut, -l* (Ver [Opciones comunes](#))
- *-path*, define la red sobre la que se trabajara. Cambia el directorio de busqueda de las redes y el nombre del archivo de salida
- *-thr*, seguido de un valor entre 0 y 1, cambia el valor de umbral bajo el cual se cortan las mascararas
- *-sd*, añade la desviacion standard

PET-FDG

El primer paso para analizar las imágenes PET-FDG es registrarlas al espacio MNI. Para ello es necesario tener las imágenes MRI previamente procesadas con Freesurfer. El script es capaz de buscar y convertir las imágenes necesarias y copiarlas al directorio *working*.

El comando a utilizar para el coregistro es,

```
$ pet_reg.pl estudio
```

Las opciones son,

- *-cut, -a, -l*, (Ver [Opciones comunes](#))

- *-norm*, normaliza las imagenes entre 0 y 1

El script de registro en el espacio MNI produce además un informe de dicho registro en el archivo *working/pets/index.html* que puede consultarse para comprobar que el registro de todas las imágenes se haya producido satisfactoriamente. Tras esto solo hay que decidir que normalización ha de hacerse y que ROI debe analizarse. Por ejemplo,

```
cd working; for x in *_pet_inMNI*; do a=`echo $x | awk -F"_" '{print $1}' | sed 's/mci//;s/hc//'`; b=`fslstats $x -k /opt/neuro.dev/lib/bin_rpons_vermis.nii.gz -M`; c=`fslstats $x -k /opt/neuro.dev/lib/ADNI_Composite.nii.gz -M`; echo $a, `echo $c/$b | bc -l` ; done > ../slandau.csv
```

normaliza por el *pons* todos los PET-FDG de los sujetos catalogados como *mci* o *hc* y halla el valor medio de la ROI de trabajo de ADNI. Esta línea está aquí a modo de ejemplo pero también puede hacerse con el comando,

```
$ parallel_fdg_adniroi_metrics.pl estudio
```

que deja los resultados en el archivo, *estudio_fdg_adni_suvr_predef.csv*. Ver más sobre las [ROI de ADNI](#).

También pueden obtenerse estos valores en toda la segmentación de freesurfer, normalizados por *pons*,

```
$ pet_fdg_rois_metrics.pl estudio
```

o normalizados por el cerebelo,

```
$ pet_fs_roi_metrics.pl estudio
```

PET-PIB

Las imágenes PET-PIB se registran al espacio anatómico del sujeto,

```
$ pib_reg.pl estudio
```

Las opciones son,

- *-cut, -a, -l*, (Ver [Opciones comunes](#))
- *-useb*, utiliza el cerebro extraído para hacer el registro
- *-usem*, registra una máscara del PET-PIB al espacio de sujeto y utiliza esta información para registrar la imagen

Nota: Las imágenes PIB son bastante problemáticas de registrar, por ello aquí las opciones *-useb* y *-usem*, así como el informe del registro. Se ha de inspeccionar visualmente el registro y relanzar con estas opciones en los sujetos que no haya funcionado.

Luego se extrae el valor medio de la captación en el cortex (**en las 5 regiones que todo el mundo**

usa), normalizado por el cerebelo,

```
$ parallel_pib_rois_metrics.pl estudio
```

los resultados se escriben en el archivo *estudio_pib_fs_suvr_predef.csv*.

PET-FBB

Esta herramienta se creo utilizando el estudio FACEHBI como piloto. El procedimiento sigue las lineas de este estudio. Se reciben 4 imagenes, cada una integrando 5 minutos. Las imagenes se coregistran independientemente al espacio del sujeto y se luego promedian.

El analisis de FBB está integrado en el cluster y ahora utiliza SLURM por lo que puede controlarse con las herramientas correspondientes.

Para ello,

```
$ fbb_correct.pl estudio
```

Las opciones son,

- *-cut*, (Ver [Opciones comunes](#))

Tras esto, se extrae el valor medio en el cortex (**si, en las mismas 5 regiones**), normalizado por el valor en la materia gris del cerebelo,

```
$ fbb_rois_metrics.pl estudio
```

los resultados se escriben en el archivo *estudio_fbb_fs_suvr_rois.csv*.

También pueden extraerse los valores de Centiloide, medidos tomando como referencia el valor en el cerebelo (completo), según [Rowe et. al.](#).

```
$ fbb_cl_metrics.pl estudio
```

y los resultados quedan en el archivo *estudio_fbb_cl.csv*.

fMRI

El tratamiento de las fMRI se ha implementado, en principio, solo para *resting state*. Se ha implementado el *Independent Component Analysis (ICA)* utilizando *feat* y *melodic* de **FSL** e integrando todo en **slurm**.

ICA individual

Para realizar el ICA de un solo sujeto se utiliza el script *rs_ica_one.pl*. La unica opcion disponible es -

cut (Ver [Opciones comunes](#)) para realizar el analisis individual en una seleccion de individuos. En caso contrario lo realiza en todos los sujetos del proyecto. Se lanza simplemente con,

```
$ rs_ica_one.pl mopead
```

Nota: Para realizar esta tarea se utiliza una plantilla que se encuentra en `${PIPEDIR}/lib/lone_ica_template.fsf` y se rellena con los directorios particulares del proyecto en si.No obstante, en caso de tener que cambiar alguna opción mas, como el *TR*, habria que editar directamente la plantilla. De momento, se ha hecho asi para simplificar el uso ya que los fMRI por ahora vienen con un unico protocolo pero esto puede estar sujeto a cambios.

Los resultados de cada sujeto quedan en *working/sujeto_rs.ica/* y el informe en *working/sujeto_rs.ica/filtered_func_data.ica/report/00index.html*

ICA grupal

Para realizar el ICA del proyecto o un grupo de sujetos se utiliza el script *rs_ica_group.pl*. La unica opcion disponible es *-cut* (Ver [Opciones comunes](#)) para realizar el analisis grupal en una seleccion de individuos. En caso contrario lo realiza en todos los sujetos del proyecto. Se lanza con,

```
rs_ica_group.pl mopead
```

Nota: Para realizar el analisis se utilizan varias plantillas que se encuentran en `${PIPEDIR}/lib/`

```
notalone_ica_template.fsf
group_ica_template_p1.fsf
group_ica_template_targets.fsf
group_ica_template_p2.fsf
```

Estas plantillas se rellenan con los directorios particulares del proyecto en si. No obstante, en caso de tener que cambiar alguna opción mas, como el *TR*, habria que editar directamente las plantillas correspondientes. Notese que en cualquier caso habria que cambiar dos plantillas, *notalone_ica_template.fsf* y la plantilla *group_ica_template_???.fsf* donde se encuentre el parametro en cuestion. Esto puede estar sujeto a cambios.

Los resultados de cada sujeto quedan en *working/rs.gica/groupmelodic.ica/* y el informe en *working/rs.gica/groupmelodic.ica/report/00index.html*

Opciones comunes

Para realizar la misma tarea se usan en cada script las mismas opciones de entrada.

- *-cut*: toma un archivo *.csv* como argumento y ejecuta el script solo sobre los sujetos nombrados en ese archivo. Este archivo debe tener el mismo formato que la base de datos del proyecto y debe ser un subconjunto de ella. O sea, solo se ejecutara el script sobre los sujetos que **tambien** esten en la base de datos del proyecto.

```
script.pl -cut soloestos.csv proyecto
```

ejecutara *script.pl* sobre los sujetos que esten en los archivos *soloestos.csv* y *proyecto.csv*. Esto es util para cuando se desea ejecutar el procesamiento sobre sujetos que llegan nuevos o repetir algunos sujetos.

- *-a*: Envía como attachment los resultados finales o los *logs* que producen los scripts
- *-/*: Cuando por defecto el script envía attachments, esta opción evita que los haga
- Para manejar el cluster se usa [slurm](#). Usando *squeue* y *scancel* se puede encontrar y eliminar cualquier tarea que se haya lanzado. Una explicación simple de estas herramientas se puede ver en [Como usar el cluster sin morir en el intento](#).

From:

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

Permanent link:

<http://detritus.fundacioace.com/wiki/doku.php?id=neuroimagen:pipe03&rev=1556617981>

Last update: 2020/08/04 10:47

