

Como hacer un merge de varios proyectos

Hay veces en que necesitamos sacar las metricas de sujetos que pertenecen a varios proyectos. Ahi va un ejemplo de como sacarlos todos. Voy a sacar el *estimated Total Intracranial Volume (eTIV)* pero esta claro que el procedimiento fundamental es igualmente valido para cualquier cosa.

Sacando los datos de XNAT

Primero tengo que bajarme la lista de sujetos del proyecto y encontrar los IDs de las MRI correspondientes. Vamos a hacerlo en *FACEHBI baseline*,

Nota: tengo que cambiar todos los `for x in cat file; do whatever; done` a `while read var; do whatever, done < file` para poder manejar los errores de encoding.

<https://stackoverflow.com/questions/4642191/read-line-by-line-in-bash-script#4642213>

```
$ xnatapic list_subjects --project_id facehbi --label > xnat_subjects.list
$ for x in `cat /nas/data/facehbi/xnat_subjects.list`; do sbj=$(echo $x |
awk -F"," '{print $1}'); lbl=$(echo $x | awk -F"," '{print $2}');
xex=$(xnatapic list_experiments --project_id facehbi --subject_id ${sbj} --
modality MRI); echo "${lbl}, ${sbj}, ${xex}"; done >
/nas/data/facehbi/xnat_sub_exp.list
```

Ahora, para sacar un valor de la segmentacion (o varios!) debo bajarme las *stats* adecuadas,

```
for l in `cat xnat_sub_exp.list`; do s=$(echo ${l} | awk -F"," '{print
$2}'); e=$(echo ${l} | awk -F"," '{print $3}'); mkdir -p fsresults/${s};
xnatapic get_fsresults --experiment_id ${e} --stats aseg fsresults/${s}/;
done
```

En este caso estoy buscando el *eTIV* asi que hago un *grep*.

```
for line in $(cat /nas/data/facehbi/xnat_sub_exp.list); do lbl=$(echo
${line} | awk -F"," '{print $1}'); xex=$(echo ${line} | awk -F"," '{print
$2}'); vol=$(grep EstimatedTotalIntraCranialVol
/nas/data/facehbi/fsresults/${xex}/aseg.stats | awk -F"," '{print $4}');
echo "${lbl},${vol}"; done > facehbi_icv.csv
sed -i 'liPSubject,eTIV' facehbi_icv.csv
(head -n 1; sort -t, -k 1) < facehbi_icv.csv > facehbi_icv_sorted.csv
```

y ahi queda todo ordenadito.

Nota: En caso de querer otras variables puedo usar las herramientas del pipeline que integran XNAT.

- [Recuperar resultados de XNAT](#)
- [sacar metricas de los resultados](#)

Ahora, para integrar con otros proyectos ajenos a neuroimagen necesito una lista de codigos internos. Una vez la consiga (del PI o asi), ya puedo enlazar los resultados con cualquier cosa,

```
join -t, /nas/data/facehbi/internos.csv facehbi_icv_sorted.csv | awk -F",", "
'{print $2","$3}' > interno_etiv_facehbi.csv
```

Acoplado los proyectos

Una vez que hice lo anterior en **todos** los proyectos y tengo todos los archivos de datos en el mismo formato puedo hacer un `cat` simple y acoplarlos.

Problemas posibles:

1. La visitas varias de un mismo sujeto y proyecto tienen el mismo código interno así que hay que tener claro cuál queremos o en caso de que se quieran los dos, hay que incluir la fecha de adquisición del MRI. esto último lo puedo hacer con **dkey -k "AcquisitionDate"** en el dicom o con **xnatopic list_experiments -project_id PROYECTO -subject_id SUJETO -modality MRI -date** en XNAT. No es difícil pero ha de tenerse en cuenta de antemano.
2. A veces los sujetos están duplicados entre algunos proyectos. Ejemplo: un sujeto de EPAD puede haber sido reclutado también para BIOFACE o MOPEAD. Aquí se escoge dependiendo del criterio de diseño.

Ahora, para hacer un `cat` voy a quitar las cabeceras y a pegar, ejemplo,

```
$ tail -n +2 interno_etiv_bioface.csv | sort -t, -k 1 >
interno_etiv_bioface.csv_data
$ tail -n +2 interno_etiv_epad.csv | sort -t, -k 1 >
interno_etiv_epad.csv_data
$ tail -n +2 interno_etiv_mopad.csv | sort -t, -k 1 >
interno_etiv_mopead.csv_data
$ tail -n +2 interno_etiv_mopead.csv | sort -t, -k 1 >
interno_etiv_mopead.csv_data
$ cat interno_all_facehbi.csv interno_etiv_bioface.csv_data
interno_etiv_mopead.csv_data interno_etiv_epad.csv_data >
project_all_interno_etiv.csv
```

y esto es lo que he de enviar.

Para buscar duplicados tengo que revisar el archivo final y los originales, o sease,

```
$ grep -f internos.list project_all_interno_etiv_sorted.csv | awk -F",", "
'{if($2!="") print $1}' | uniq -c | awk '{if($1=="2") print $2}' >
duplicados.list
$ while read duh; do grep ${duh} interno_etiv_*.csv; echo; done <
duplicados.list
interno_etiv_bioface.csv:20130864,1475358.690045
interno_etiv_epad.csv:20130864,1477459.121125
```

```
interno_etiv_bioface.csv:20140926,1614550.686010
interno_etiv_epad.csv:20140926,1606870.232021

interno_etiv_bioface.csv:20141453,1437298.542239
interno_etiv_epad.csv:20141453,1430215.060153

interno_etiv_epad.csv:20180766,1633492.746290
interno_etiv_mopead.csv:20180766,1586864.537846

interno_etiv_epad.csv:20181011,1438380.312748
interno_etiv_mopead.csv:20181011,1464329.314373

interno_etiv_bioface.csv:20181300,1442662.338398
interno_etiv_epad.csv:20181300,1457458.617213

interno_etiv_bioface.csv:20181413,1487928.336476
interno_etiv_epad.csv:20181413,1489329.254143

interno_etiv_bioface.csv:20181607,1578874.062744
interno_etiv_epad.csv:20181607,1353869.461084

interno_etiv_bioface.csv:20190009,1506688.086362
interno_etiv_epad.csv:20190009,1569090.695244

interno_etiv_bioface.csv:20190245,1381622.878206
interno_etiv_epad.csv:20190245,1400824.691327
```

Nota: La forma correcta de hacer el *loop* es esta ultima!!!!

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

Permanent link:
http://detritus.fundacioace.com/wiki/doku.php?id=neuroimagen:merging_all

Last update: **2022/03/18 15:06**

