

# Realizar el QC a la segmentacion de Freesurfer de los sujetos analizados en XNAT

[El manual de desarrollo aqui](#)

La idea es que la segmentacion de Freesurfer la estoy haciendo en XNAT pero quiero hacer un *quality control* decente. Hemos escogido [VisualQC](#) pues proporciona informacion visual detallada de la segmentacion y tiene una interfaz amigable

## Preliminares

Se necesita tener instalado [xnatapic](#). [Ver detalles de la instalación aqui](#).

Para hacerlo en el propio cluster, es tan simple como crear el directorio `~/.xnatapic` y dentro el archivo `xnat.conf` con el contenido,

`xnat.conf`

```
HOST=http://detritus.fundacioace.com:8088
USER=usuario
PASSWORD=contrasenna
```

Ahora, los procedimientos para el QC no son originales de `xnatapic`. Para que funcionen los procedimientos que siguen es necesario bajar y descomprimir en el directorio `~/.xnatapic` [los scripts que añaden estas funciones](#).

Tras esto debe quedar algo parecido a,

```
[osotolongo@brick03 ~]$ ls -l ~/.xnatapic/
total 193
-rwxrwxr-x 1 osotolongo osotolongo 2351 Jun 11 19:17 get_fsqc.sh
-rwxrwxr-x 1 osotolongo osotolongo 2619 Jun 12 18:11 prepare_fsqc.sh
-rwxrwxr-x 1 osotolongo osotolongo 2008 Jun 11 19:06 upload_fsqc.sh
-rw-rw---- 1 osotolongo osotolongo 131 Jun 5 18:57 xnat.conf
```

## Preparando el directorio base

El primer paso para poder hacer el QC es bajar los archivos necesarios de XNAT para cada sujeto. La forma mas simple es ejecutando,

```
[osotolongo@brick03 test_xnat_fsqc]$ xnatapic prepare_fsqc --project_id
f5cehbi
```

Que encuentra las MRI del proyecto, crea el subdirectorio apropiado bajo el archivo comprimido que contiene la segmentacion de Freesurfer y extrae los archivos necesarios. Además genera una lista de las MRI de las cuales ha podido encontrar la segmentacion.

Las opciones pueden verse en la ayuda,

```
[osotolongo@brick03 ~]$ xnatapic prepare_fsqc --help
prepare_fsqc - create the structure for Freesurfer QC with visualQC
--help: show this help
--project <project id> [mandatory]
--outdir <output directory> [defaults to <project id>_fsresults]
--list <list of downloaded experiments> [defaults to <project
id>_experiments.list]
```

## Haciendo el QC con visualQC

El QC de la segmentacion lo hacemos con VisualQC. Para esto hay que entrar en el cluster con interfaz grafica y llamar el programa con algo parecido a,

```
[osotolongo@brick03 test_xnat_fsqc]$ vqcfs -i f5cehbi_experiments.list -f
f5cehbi_fsresults -o visualqc_output
```

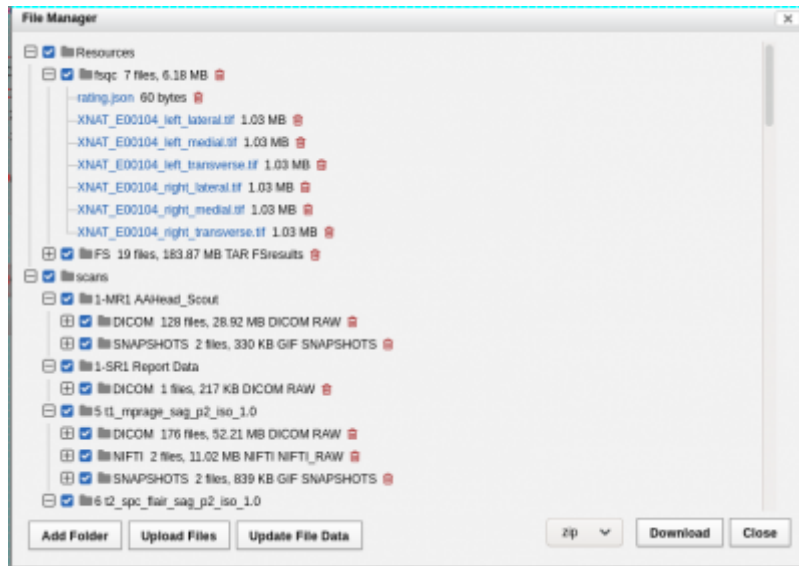
donde *f5cehbi\_experiments.list* es un archivo que contiene una lista de las segmentaciones a revisar, *f5cehbi\_fsresults* es el directorio que contiene los datos de Freesurfer y *visualqc\_output* es donde generaremos los resultados. La base de este software [esta ya descrita](#) pero la [ayuda online](#) es muy buena.

## Guardando los resultados

Después que hayamos revisado las imagenes segmentadas tendremos que guardar en XNAT los resultados del QC. Esto se hace con la opcion *upload\_fsqc*, indicando la ubicacion del directorio output de *visualqc\_freesurfer*,

```
[osotolongo@brick03 test_xnat_fsqc]$ xnatapic upload_fsqc --qcdir
visualqc_output
```

Los resultados del QC quedan almacenados dentro de la informacion de cada MRI en XNAT.



En particular, el archivo *rating.json* almacena los resultados del rating.

```

{
  "ResultSet":
  {
    "Result":
    [
      {
        "rating": "gOOD",
        "notes": "None"
      }
    ]
  }
}

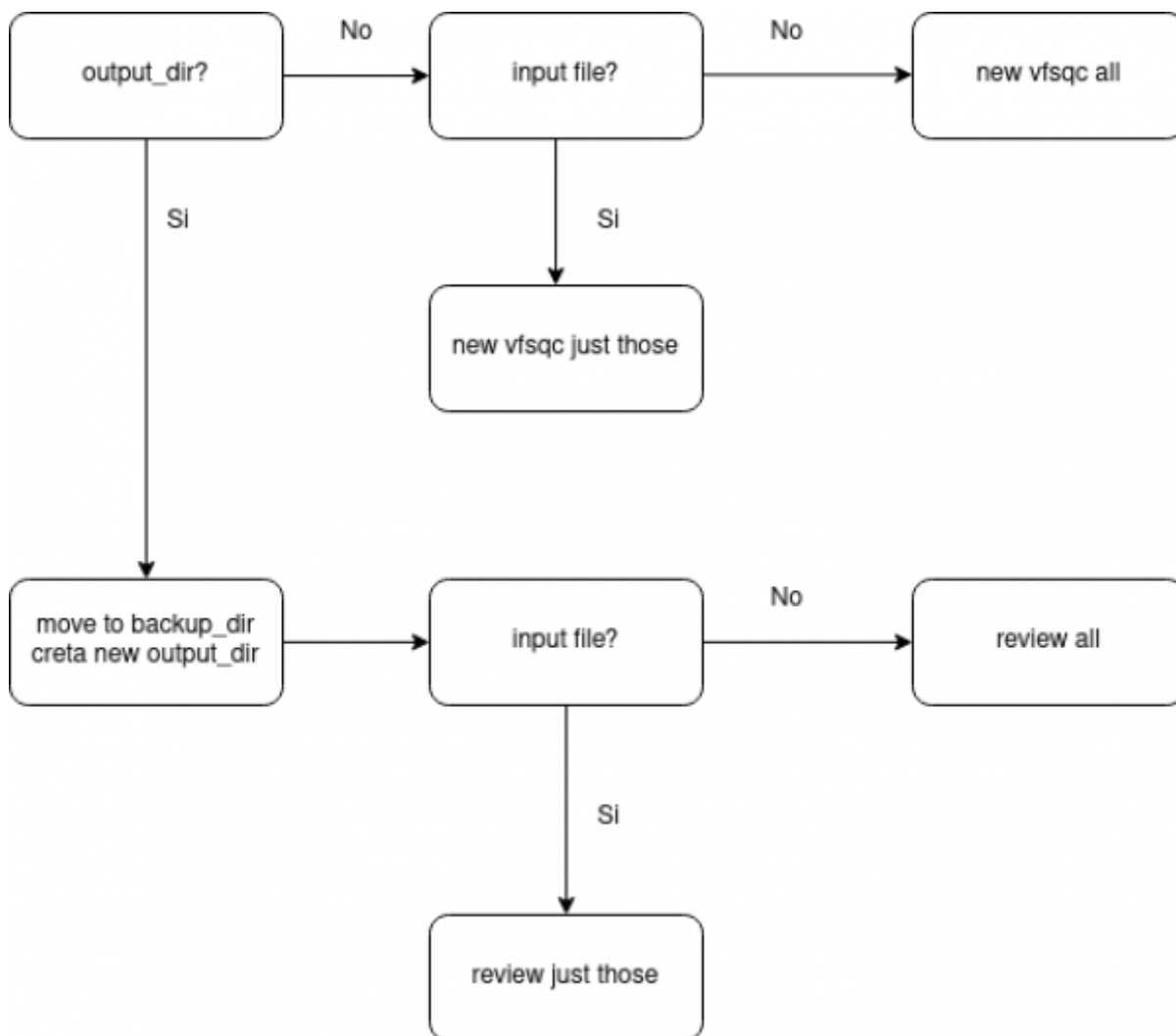
```

De esta manera los resultados quedan guardados y pueden consultarse en cualquier momento, ya sea visualmente mediante la interfaz web de XNAT, o programaticamente.

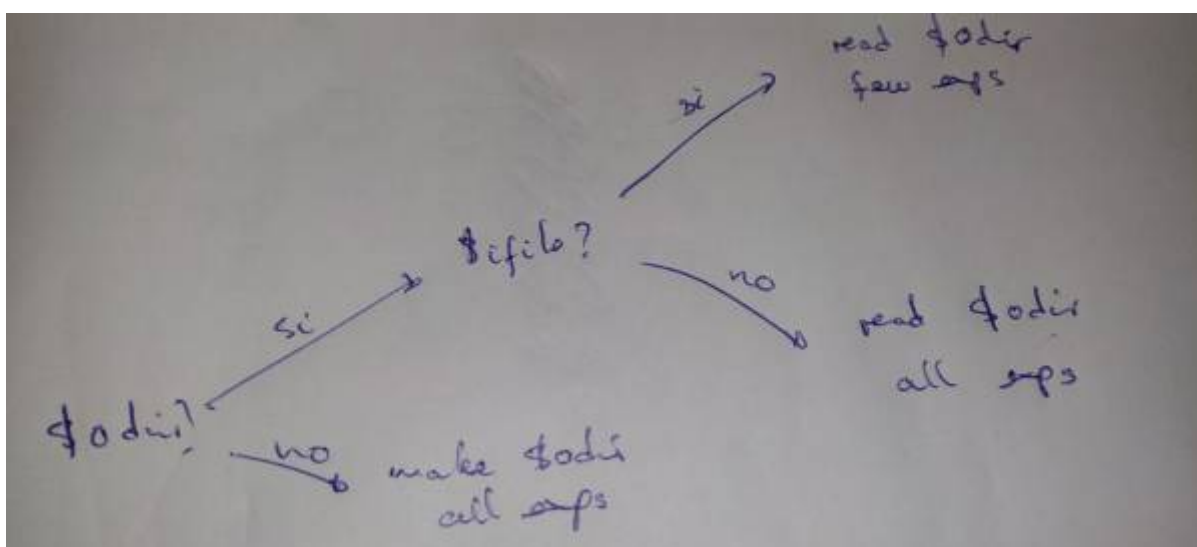
## un poco de logica (review FSQC)

Ahora lo que quiero hacer es algo para lo que visualQC no ha sido pensado. Quiero lanzar una herramienta que mire si los experimentos estan ya bajados y pasarle un lista de lo que quiero revisar. Si no paso la lista los revisa todos, y si la paso toma el directorio, escog ellos que quiero, los reviso y suvbe los resultados.

Esto es contraintuitivo, en relacion a los scripts de xnatapic que bajan y procesan el QC.



Voy a intentar revisar el FSQC despues incluso que se haya hecho. Tengo que hacerle [un frontend a esto](#).



**Importante:** el frontend hay que lanzarlo en un entorno grafico pues ademas de conectarse a XNAT y hacer cosas por debajo, va a lanzar la herramienta de evaluacion visual. He intentado regular esto añadiendo,

```
die "Should run this under X!\n" unless $ENV{'DISPLAY'};
```

pero todavia asi hay escenarios en que puede que este fijada la variable `$DISPLAY` y el script no encuentre el entorno grafico. O puedo fijar el variable `$DISPLAY` a un entorno grafico que ya exista desde una terminal que no lo tenga y me aparecera alli. 😊

Para el frontend tengo que contemplar la posibilidad de empezar de cero. Vamos, lo basico que es lanzarlo e intentar procesar todo. Es decir que si hago solo,

```
$ review_vqcfs.pl -x f5cehbi
```

el script intentara bajar *TODO* lo que exista en el proyecto `f5cehbi`, lanzará `vqcfs` para todos los que existan y despues subirá los resultados.

Luego puedo definir el directorio donde estan los sujetos y entonces no tendre que bajarme nada, sino leer los datos que esten alli.

## Consultando el QC

Se puede obtener un CSV con los resultados del QC haciendo,

```
[osotolongo@brick03 test_xnat_fsqc]$ xnatapic get_fsqc --project f5cehbi
```

El archivo de output se puede cambiar con las opciones (ver la ayuda) y obtenemos un archivo CSV de este tipo,

```
[osotolongo@brick03 test_xnat_fsqc]$ head f5cehbi_fsqc.csv
F005, "gOOD", "None"
F006, "gOOD", " Pues parece que est aben"
F007, "rEVIEW LATER", ""
F009, "MInOR error", " not realy an error"
F010, "gOOD", ""
F014, "rEVIEW LATER", ""
F015, "gOOD", ""
F001, "gOOD", "None"
F100, "gOOD", "None"
F145, "I'M TIREd", "None"
```

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

Permanent link:  
[https://mail.fundacioace.com/wiki/doku.php?id=neuroimagen:visualqc\\_freesurfer\\_xnat](https://mail.fundacioace.com/wiki/doku.php?id=neuroimagen:visualqc_freesurfer_xnat)

Last update: **2021/08/07 14:42**

