

# QC de Freesurfer con VisualQC CLI

[https://raamana.github.io/visualqc/cli\\_freesurfer.html](https://raamana.github.io/visualqc/cli_freesurfer.html)

## Protocolo y criterios de QC

Para publicar cualquier tipo de resultados, ya sea dentro o fuera de ACE, deberá hacerse antes este QC. Los resultados publicados sin QC previo, no son válidos.

Solo reportaremos dos tipos de resultados:

1. **Fail:** errores graves, movimiento excesivo, falta parte de la imagen, etc
2. **Good:** el resto

Una vez en posesión del CSV con las calificaciones, debemos filtrar los archivos de resultados a los que estén calificados como *Good* o acompañar los resultados con este archivo, a modo de advertencia.

Para obtener el FSQC almacenado en XNAT hacemos, por ejemplo,

```
$ xnatapic get_fsqc --project_id facehbi
```

siendo en esta caso *facehbi* el nombre del proyecto en XNAT.

## Traer los sujetos de XNAT

La segmentación de Freesurfer actual se hace con XNAT, así que hay que sistematizar una manera para bajar los sujetos, hacer el QC, subir esta info a XNAT y borrar los que hemos bajado.

## Preparando el QC del proyecto

Antes de ejecutar el QC del proyecto necesitamos,

- Una lista de los sujetos que vamos a visualizar
- Los directorios de output de cada sujeto, colocados en la misma ubicación

Ejemplo, tengo algunos sujetos de la V5 de facehbi que quiero revisar,

```
[osotolongo@brick03 f5cehbi]$ ls -d /nas/data/subjects/f5cehbi*  
/nas/data/subjects/f5cehbi_0001 /nas/data/subjects/f5cehbi_0012  
/nas/data/subjects/f5cehbi_0023  
/nas/data/subjects/f5cehbi_0002 /nas/data/subjects/f5cehbi_0013  
/nas/data/subjects/f5cehbi_0024  
/nas/data/subjects/f5cehbi_0003 /nas/data/subjects/f5cehbi_0014  
/nas/data/subjects/f5cehbi_0025
```

```
/nas/data/subjects/f5cehbi_0004 /nas/data/subjects/f5cehbi_0015
/nas/data/subjects/f5cehbi_0026 /nas/data/subjects/f5cehbi_0016
/nas/data/subjects/f5cehbi_0005 /nas/data/subjects/f5cehbi_0017
/nas/data/subjects/f5cehbi_0027 /nas/data/subjects/f5cehbi_0018
/nas/data/subjects/f5cehbi_0006 /nas/data/subjects/f5cehbi_0019
/nas/data/subjects/f5cehbi_0028 /nas/data/subjects/f5cehbi_0020
/nas/data/subjects/f5cehbi_0007 /nas/data/subjects/f5cehbi_0021
/nas/data/subjects/f5cehbi_0029 /nas/data/subjects/f5cehbi_0022
/nas/data/subjects/f5cehbi_0008 /nas/data/subjects/f5cehbi_0033
/nas/data/subjects/f5cehbi_0030
/nas/data/subjects/f5cehbi_0009
/nas/data/subjects/f5cehbi_0031
/nas/data/subjects/f5cehbi_0010
/nas/data/subjects/f5cehbi_0032
/nas/data/subjects/f5cehbi_0011
/nas/data/subjects/f5cehbi_0033
```

obtener la lista es tan sencillo como,

```
[osotolongo@brick03 f5cehbi]$ ls -d /nas/data/subjects/f5cehbi* | awk -F"/"
'{print $5}' > fs_subject.list
[osotolongo@brick03 f5cehbi]$ head fs_subject.list
f5cehbi_0001
f5cehbi_0002
f5cehbi_0003
f5cehbi_0004
f5cehbi_0005
f5cehbi_0006
f5cehbi_0007
f5cehbi_0008
f5cehbi_0009
f5cehbi_0010
```

## Lanzando el CLI

Con todo preparado podemos lanzar el CLI con las opciones basicas,

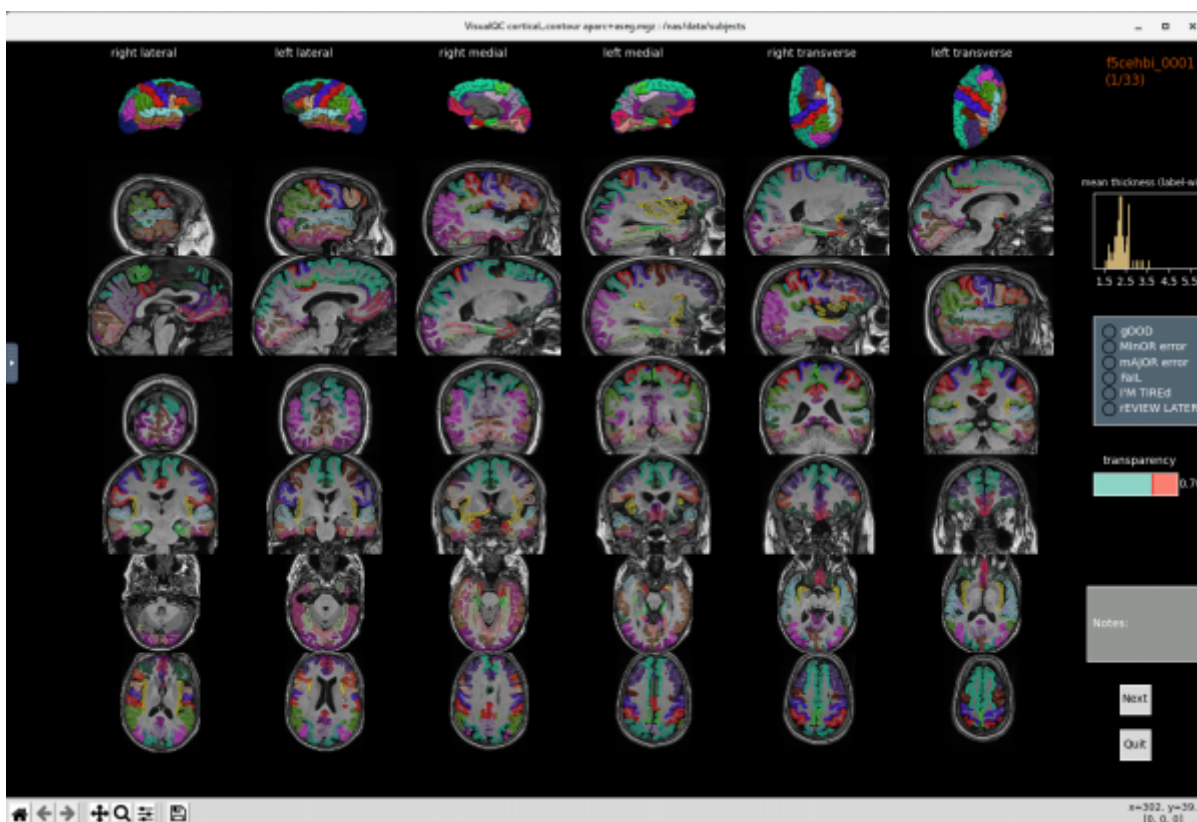
```
[fsluser@FSLVm7_64 ~]$ visualqc_freesurfer -i
/nas/data/f5cehbi/fs_subject.list -f /nas/data/subjects -o
/nas/data/f5cehbi/vfsqc
```

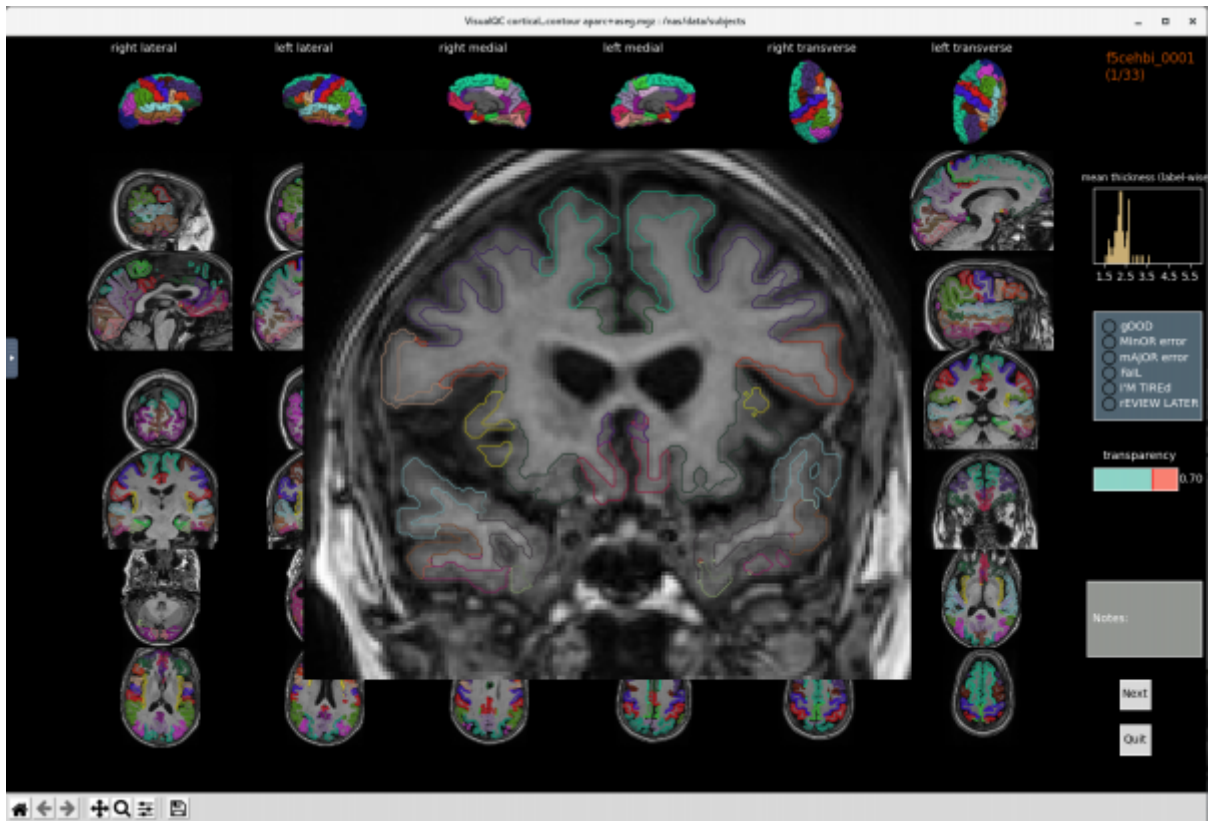
**Demora en procesar, asi que si ponemos una lista de un proyecto grande mejor nos damos una vuelta.** La parte buena es que el procesamiento solo se ejecuta una vez y se guarda en el directorio de *output* asi que podemos volver a lanzarlo en cualquier momento y sera mas rapido.

```
[osotolongo@brick03 f5cehbi]$ ls vfsqc/
annot_visualizations ratings
[osotolongo@brick03 f5cehbi]$ ls vfsqc/annot_visualizations/
f5cehbi_0001_left_lateral.tif f5cehbi_0012_left_medial.tif
```

```
f5cehbi_0023_left_transverse.tif  
f5cehbi_0001_left_medial.tif  
f5cehbi_0023_right_lateral.tif  
f5cehbi_0001_left_transverse.tif  
f5cehbi_0023_right_medial.tif  
f5cehbi_0001_right_lateral.tif  
f5cehbi_0023_right_transverse.tif  
f5cehbi_0001_right_medial.tif  
f5cehbi_0024_left_lateral.tif  
...  
f5cehbi_0012_left_transverse.tif  
f5cehbi_0012_right_lateral.tif  
f5cehbi_0012_right_medial.tif  
f5cehbi_0012_right_transverse.tif
```

Tras el procesamiento se muestra la pantalla principal donde podemos evaluar y puntuar la calidad de la segmentacion (haciendo doble click en una imagen se amplia). La interfaz es intuitiva y bastante agil.





Tras evaluar los sujetos, los resultados quedan almacenados en el directorio de output,

```
[osotolongo@brick03 f5cehbi]$ ls vfsqc/ratings/  
cortical_contour_rate_freesurfer_orig.mgz_ratings.all.csv  
id_list_rating_i_m_tired.txt  
id_list_rating_good.txt  
id_list_rating_minor_error.txt  
[osotolongo@brick03 f5cehbi]$ cat  
vfsqc/ratings/cortical_contour_rate_freesurfer_orig.mgz_ratings.all.csv  
f5cehbi_0001,gOOD,None  
f5cehbi_0002,gOOD,None  
f5cehbi_0003,I'M TIRED,None  
f5cehbi_0004,MIInOR error,None
```

## El script

tl;dr

```
$ review_vqcfs.pl -i experiments.list -d fsresults facehbi
```

## La explicacion

Primero que todo, esto hay que hacerlo en un entorno grafico. Así que vamos a hacer un chequeo simple al inicio,

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

Luego, las variables. Necesito una lista de los experimentos que quiero bajarme de XNAT (opcional), un directorio donde voy a procesar (opcional) y el nombre del proyecto en XNAT.

Si hay lista inicial, proceso solo los experimento listados, si no hay, todos los experimentos del proyecto.

```
my @pollos;
$file=$wdir.'/'.$xprj.'_experiment.list';
if ($ifile and -f $ifile){
    $file=$wdir.'/'.$xprj.'_experiment_custom.list';
    open IDF, "<$ifile";
    @pollos = <IDF>; chomp @pollos;
    close IDF;
    open TDF, ">$file";
    foreach my $pollo (@pollos){
        my $look = "xnatopic list_experiments --project_id $xprj --
subject_id $pollo --modality MRI";
        my $epollo = qx/$look/;
        chomp $epollo;
        print TDF "$epollo\n";
    }
    close TDF;
}
```

Ahora, si hay directorio a procesar, y este existe, entonces asumo que ya he procesado algunos y lo que hago es continuar el trabajo lo deje. Pero si no doy el nombre del directorio y este existe, entonces asumo que quiero reprocesar y muevo el directorio antiguo a otro sitio.

```
unless ($odir and -d $odir) {
    $odir=$wdir.'/'.$xprj.'_fsresults';
    if ( -d $odir ){
        $odir = $odir.'_' .getLoggingTime();
    }
    mkdir $odir;
    if ($ifile and -f $ifile){
        foreach my $pollo (@pollos){
            my $pre = "xnatopic prepare_fsqc --project_id $xprj
--outdir $odir --subject_id $pollo";
            system($pre);
        }
    }else{
        my $pre = "xnatopic prepare_fsqc --project_id $xprj --outdir
$odir";
        system($pre);
    }
}
```

Esto puede demorar un poco 😊 pero después que tengo todo en su lugar puedo lanzar el *vqcf*s,

```
if ($ifile and -f $ifile){  
    my $order = "vqcfs -i $efile -f $odir -o $vqcd";  
    system($order);  
}else{  
    my $order = "vqcfs -f $odir -o $vqcd";  
    system($order);  
}
```

Y cuando termine, se suben los resultados

```
my $post = "xnatapic upload_fsqc --qcdir $vqcd";  
system($post);
```

From:

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

Permanent link:

[https://imagen.fundacioace.com/wiki/doku.php?id=neuroimagen:visualqc\\_freesurfer](https://imagen.fundacioace.com/wiki/doku.php?id=neuroimagen:visualqc_freesurfer)

Last update: **2021/09/14 07:19**

