

tags: Tutorial

Tutorial FEAT

Luis Concha

Instituto de Neurobiología, UNAM Campus Juriquilla

lconcha@unam.mx (<mailto:lconcha@unam.mx>).

★ Los datos pueden descargarse desde esta liga.

(<https://drive.google.com/drive/folders/1WrHzbLMBLu2qEW2mD2epE2mZ7ZENOVNp?usp=sharing>).

📺 El video del tutorial está en youtube. (<https://youtu.be/h3w6FotGvgQ>).

ℹ Los datos fueron adquiridos en un resonador GE Discovery MR750 de 3 teslas. La tarea del sujeto consistió en abrir/cerrar su mano derecha en bloques de reposo/actividad de 30 segundos, comenzando por reposo.

En este tutorial estamos asumiendo que se tiene instalado fsl versión 6 o superior en linux. Las rutas de las carpetas que se usarán son específicas del tutorial y deberán ser modificadas por el usuario acordemente a su sistema.

Abrimos una terminal (<https://www.softzone.es/linux/tutoriales/terminal-linux/>). Esto depende de tu computadora. En una Mac debe llamarse así, o consola, no lo sé. En linux casi siempre se llama terminal, y lo encuentras en las herramientas, y en muchas distribuciones se puede abrir tecleando simultáneamente `Ctrl + Alt + t`.

Conversión de formato

(en caso de tener los archivos en formato NIFTI, ignorar esta sección y pasar directamente a la sección FEAT (https://hackmd.io/nHFpcNZ_R56S7ncBQBbr-w#FEAT)).

Ahora vamos a convertir nuestros archivos desde formato DICOM hacia formato NIFTI (nii). Para ello usaremos el programa dcm2nii (<https://github.com/rordenlab/dcm2nii>), pero primero debemos navegar hacia nuestra carpeta.

ℹ Para fines de este tutorial, todos los archivos estarán grabados en la ruta `/misc/nyquist/lconcha/tmp/fMRI2022`, pero si estás haciendo este tutorial en tu propia computadora, la ruta será diferente.

Por lo tanto, en la terminal tecleamos:

```
1 | cd /misc/nyquist/lconcha/nobackup/fMRI2022
```

Si queremos ver el contenido de la carpeta, usamos el comando:

```
1 | ls
```

Y veremos que existe una carpeta llamada `datos` . Ahí están los Dicoms. Podemos ver el contenido:

```
1 | ls datos
```

Vemos que tiene dos carpetas, una para los datos anatómicos `1401_T1W_3D_SENSE` y otra para los datos funcionales `1101_fmRI_SMA` . Adentro de cada una de esas carpetas hay muchos, muchos archivos con extensión `.dcm` (uno para cada rebanada).

Los archivos dicom serán compartidos por todos en el tutorial, pero cada quien requiere su carpeta para todos sus resultados. Necesitamos hacer una carpeta personal, a la que llamaremos de acuerdo a nuestro nombre de usuario (en este tutorial, es `fmRI_test`) y la haremos dentro de este mismo directorio, así que es fácil:

```
1 | mkdir fmRI_test
```

⚠ No escribas `fmRI_test` , sino el nombre que quieras darle a la carpeta.

Para convertir los datos de DICOM a NIFTI, el programa `dcm2niix` espera una ENTRADA (carpeta donde están los dicoms), y opcionalmente una SALIDA (carpeta donde queremos grabar los niftis, `-o` ; el default es la carpeta de entrada, pero no es recomendable mezclar ambos tipos de archivos). Por lo tanto, para convertir los datos anatómicos el comando queda como:

```
1 | dcm2niix -o fmRI_test datos/1401_T1W_3D_SENSE/
```

Igualmente, convertimos los datos funcionales:

```
1 | dcm2niix -o fmRI_test datos/1101_fmRI_SMA/
```

Si vemos el contenido de nuestra carpeta personal, encontraremos archivos nifti:

```
lconcha@mansfield:/misc/nyquist/lconcha/fMRI2022$ ls fmRI_test
1101_fmRI_SMA_fmRI_SMA_SENSE_20121004193706_1101.nii
1401_T1W_3D_SENSE_T1W_3D_SENSE_20121004193706_1401_brain.nii.gz
1401_T1W_3D_SENSE_T1W_3D_SENSE_20121004193706_1401.nii
```

❗ Los nombres de los archivos se generan a través de los meta-datos de los archivos DICOM, que por default son la fecha de adquisición y tipo de imagen. Por lo tanto, los nombres de los archivos quedan muy largos. El comportamiento de dcm2niigui se puede modificar en Help-Preferences, pero por ahora dejaremos los valores por defecto.

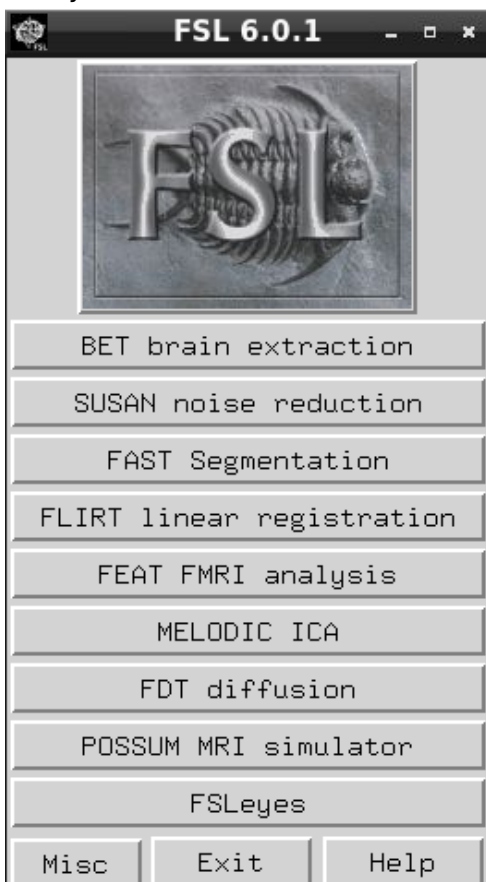
😊 A mí me gusta cambiarle los nombres a los archivos, por ejemplo algo como `sujeto_experimento_bold.nii.gz` o `sujeto_experimento_t1.nii.gz`. Si notas que en el tutorial tengo nombres distintos a los arriba mencionados, es porque los cambié.

🧠 Lo más recomendable es adherirse al estándar internacional que se ha desarrollado en los últimos años, llamado BIDS. Esto hace homogéneos los nombres de los archivos pero además abre la puerta a herramientas de análisis que comparten un mismo ecosistema. [Aprende más acerca de BIDS aquí.](https://bids.neuroimaging.io/) (<https://bids.neuroimaging.io/>).

Ahora que tenemos nuestros datos, podemos procesarlos.

FEAT

En la terminal, escribimos `fsl`. Se abrirá el menú de fsl. Seleccionamos el quinto, `FEAT FMRI analysis`.



En la parte superior, dejaremos puesto `First-level analysis` y `Full analysis`. Si quisiéramos hacer análisis de grupo, cambiaríamos esa opción a `Higher-level analysis`, y si quisiéramos brincarnos algunos pasos del análisis, quitaríamos `full analysis` y seleccionaríamos alguna de las otras opciones.

Primero, indicamos nuestro archivo con las imágenes funcionales. Para ello, presionamos el botón que está en la pestaña *Data*, que dice *Select 4D Data*. Usamos el ícono del folder para navegar a donde está nuestro archivo, que es en la carpeta

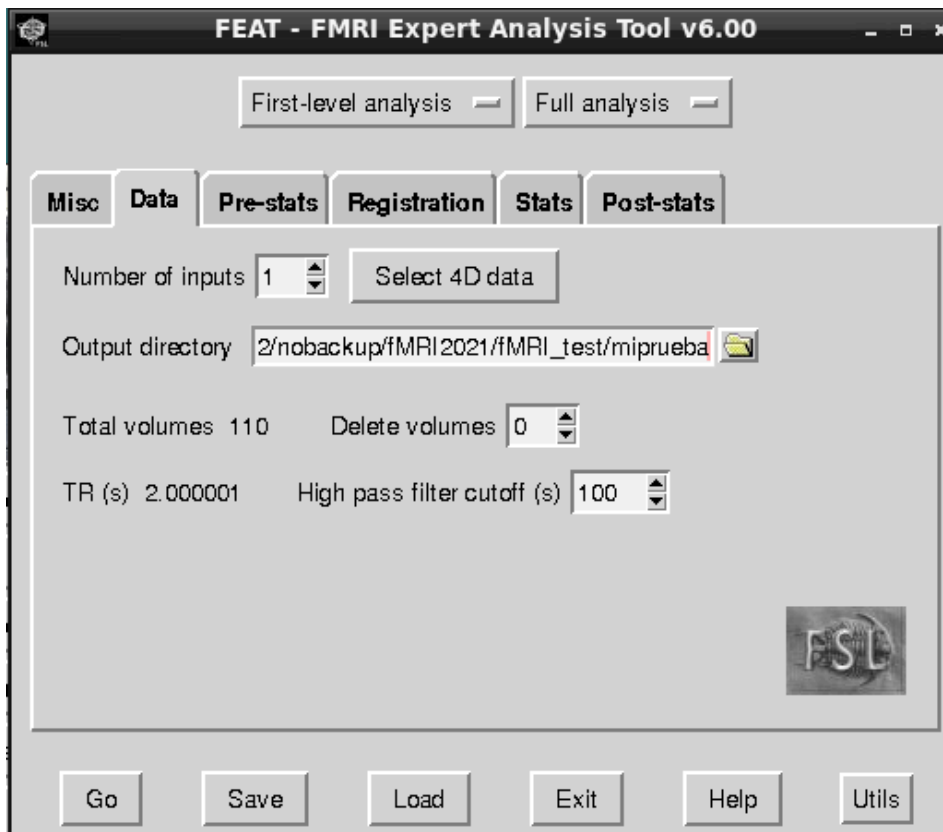
`/misc/nyquist/lconcha/tmp/fMRI2022/USUARIO` (recuerda cambiar `USUARIO` por el nombre de la carpeta que hicimos). El archivo se llama

`1101_fMRI_SMA_fMRI_SMA_SENSE_20121004193706_1101.nii`. Seleccionamos *OK*, y nuevamente *OK* en la ventanita de *Select Input Data*.

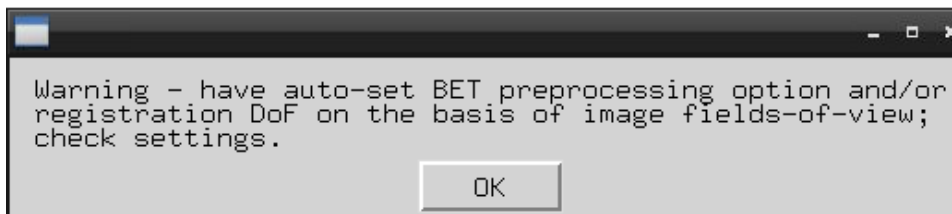
Aprovechamos y ponemos la ruta del *Output directory*, que en este tutorial será en `/misc/nyquist/lconcha/tmp/fMRI2022/USUARIO/miprueba` (recuerda cambiar `USUARIO` por el nombre de la carpeta). Para ello podemos escribirlo directamente en el recuadro de texto, o utilizar el ícono del folder y navegar a la carpeta.

i El programa FEAT siempre agregará el sufijo `.feat` a la carpeta output que nosotros indiquemos. Sin embargo, esto no lo muestra en la caja de texto de *Output directory*. Así, el resultado quedará en la ruta: `/misc/nyquist/lconcha/tmp/fMRI2022/USUARIO/miprueba.feat`

i los nombres de archivos en linux son sensibles a mayúsculas/minúsculas, y no deben contener espacios.

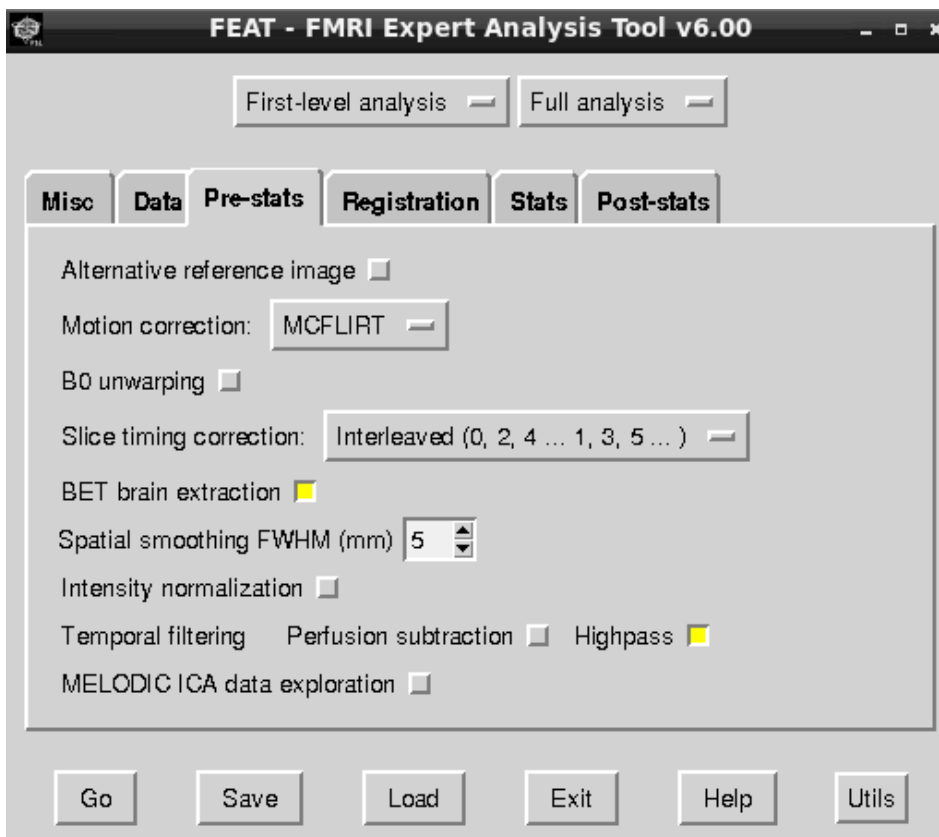


Saldrá una advertencia:



Esto nos está diciendo que el default del programa es hacer extracción de cerebro en las imágenes anatómicas, así que debemos poner imágenes anatómicas también con extracción de cerebro (ver adelante). Ahora, decimos OK . Podemos ver que en la pestaña *Data*, el software ya reconoció que nuestro archivo tiene 110 volúmenes, y que el TR es de 2 segundos. Si no lo hubiera reconocido, tenemos la opción de escribirlo (este dato lo sabríamos de acuerdo a nuestra adquisición). Dejamos las otras opciones intactas, pues no necesitamos eliminar volúmenes, y el filtro pasa-altas está en 100s (de forma predeterminada aparece 60, o 100), que es un buen valor por default.

Vamos a la pestaña *Pre-stats*. Lo único que cambiaremos es *Slice Timing correction* , a *Interleaved* . Este parámetro depende de cada tipo de adquisición. Podríamos cambiar el nivel de suavizado de la imagen mediante FWHM, pero el valor 5 mm por ahora es bueno (normalmente se utilizan FWHM con valores de 1.5 a 2 veces la dimensión del voxel).



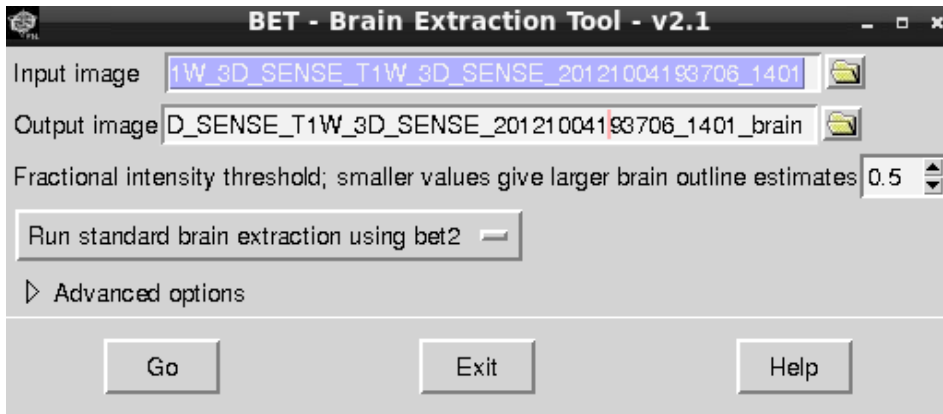
Pasamos a la pestaña *Registration*.

Activamos el botón *Main Structural image* . Aquí vamos a poner una imagen anatómica de nuestro sujeto, pero primero debemos hacerle extracción de cerebro.

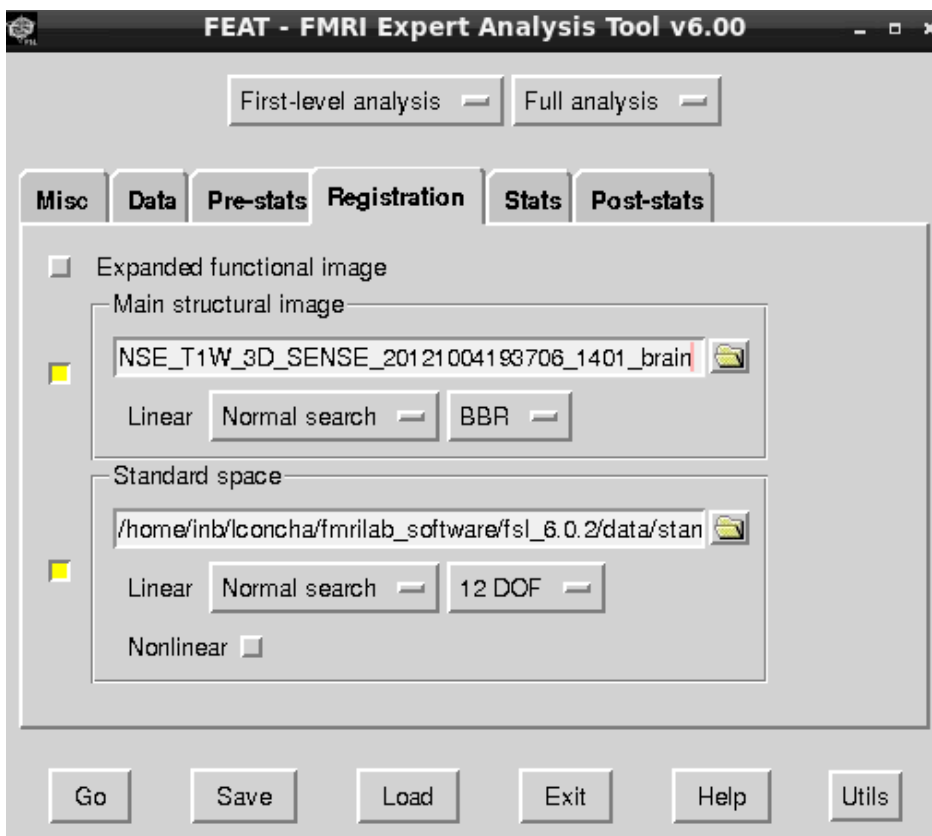
Extracción de cerebro

Regresemos a la ventana principal FSL y presionemos el primer botón, *BET brain extraction* .

En **Input Image** , usemos el botón del folder para ir a buscar nuestra imagen anatómica, que se llama `1401_T1W_3D_SENSE_T1W_3D_SENSE_20121004193706_1401.nii` . La salida se pone automáticamente, con el mismo nombre del archivo de entrada, pero con un `_brain` agregado.

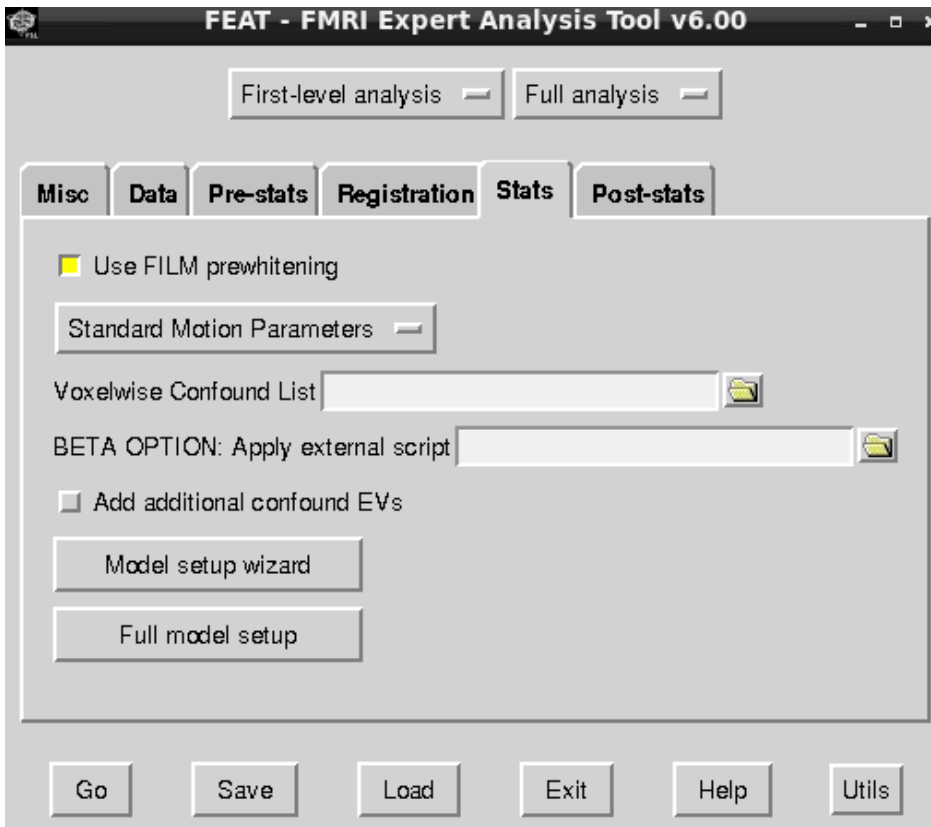


Regresamos a la ventana FEAT, pestaña *Registration*. Usemos el botón de folder para incluir una **Main Structural Image** , y seleccionamos nuestra nueva imagen (poner la que termina con `_brain`), presionamos **OK** .

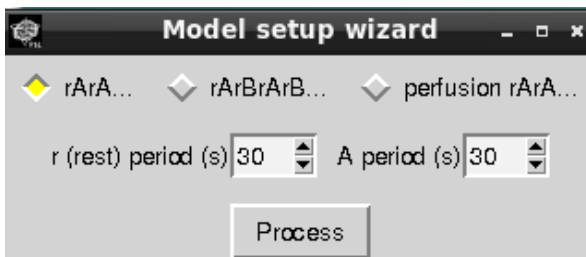


Vamos a la pestaña *Stats*.

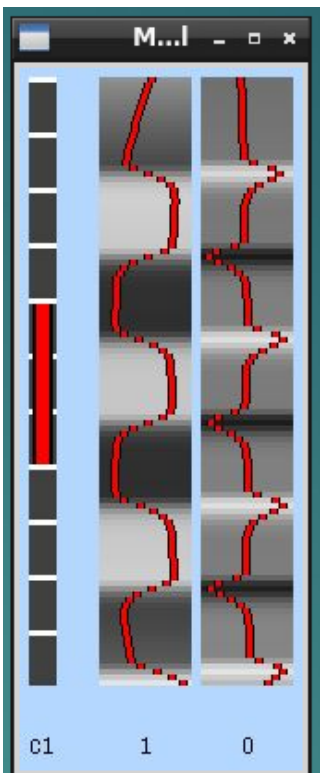
Dejamos activado **Use FILM prewhitening** . En el primer botón, apretamos el botón **Don't Add Motion Parameters** , y en el menú que se abrirá, seleccionamos **Standard Motion Parameters** . Y ahora presionamos **Model setup wizard** .



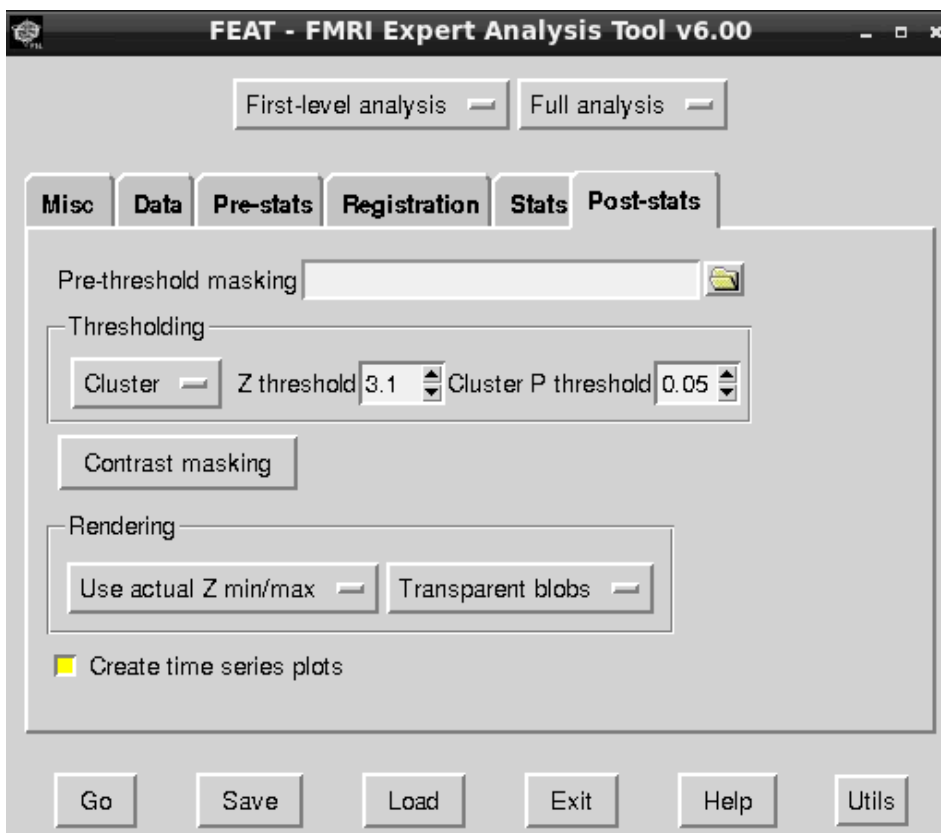
Nuestro paradigma fue de movimiento de mano, 30 seg sí, 30 seg no, iniciando con reposo. Por lo tanto, seleccionamos el modelo `rArA...`, y dejamos en 30 los periodos `rest` y `A`.



Presionamos `Process`. Veremos el paradigma, con la línea de tiempo en la vertical. Hay tres columnas, la primera es el tiempo, con barras blancas cada 20 segundos. La segunda columna es nuestro paradigma experimental (reposo-actividad), pero convolucionado con la respuesta hemodinámica. La tercer columna es la derivada de la primer columna, y se calcula automáticamente.

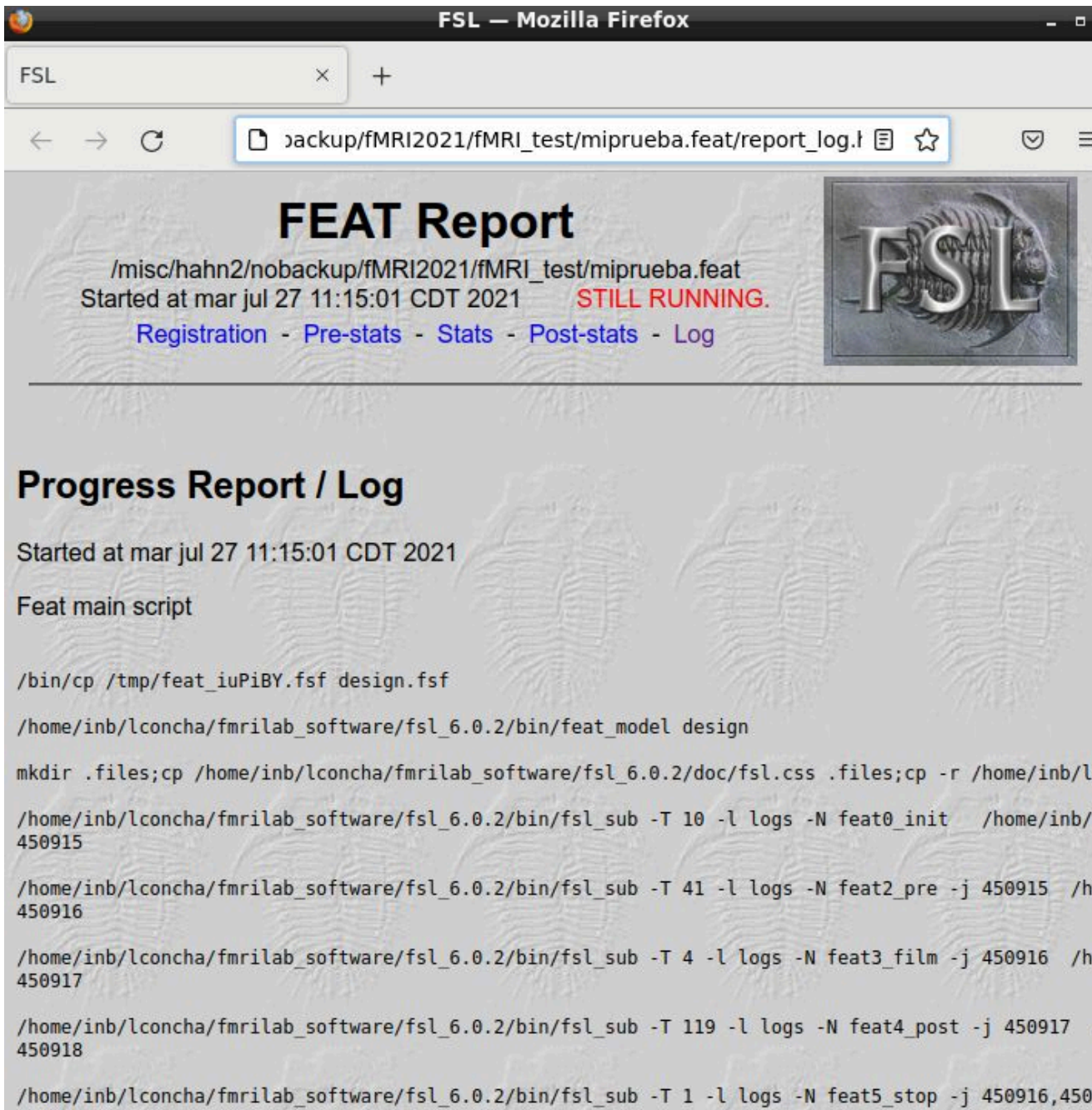


Por ahora no haremos nada en la pestaña *Post-stats*.



Estamos listos! Presionamos **GO**. El proceso tardará 10 a 15 minutos. Se abrirá una página web (local) en el navegador de internet, donde veremos el progreso. En caso de que cerremos la ventana, o que ésta no se haya abierto por default, podemos encontrar el archivo `.html` en la siguiente ruta:

`/misc/nyquist/lconcha/tmp/fMRI2022/fMRI_test/miprueba.feat/report.html`



Revisando resultados.

Esta página tiene 5 pestañas. *Registration* mostrará los resultados del registro de las imágenes funcionales con las T1, y éstas con el atlas. *Pre-stats* nos mostrará resultados de la corrección de movimiento y otros pre-procesos. *Stats* nos indica el modelo estadístico que estamos probando. *Post-stats* nos muestra el resultado de la evaluación estadística por voxel, después de haber hecho las correcciones pertinentes por comparaciones múltiples (Random-field theory, en nuestro caso). Finalmente, *Log* es una bitácora de los procesos que hizo el programa, y es útil para revisarlo en caso de que algo haya fallado.

En la pestaña *post-stats* podemos ver que nuestro modelo muestra un muy buen ajuste con la señal BOLD en la región motora izquierda, correspondiente al paradigma experimental (mover la mano derecha).

FEAT Report

/misc/hahn2/nobackup/fMRI2021/fMRI_test/miprueba.feat
Finished at Tue Jul 27 11:24:57 CDT 2021

[Registration](#) - [Pre-stats](#) - [Stats](#) - [Post-stats](#) - [Log](#)



Post-stats

Analysis methods

FMRI data processing was carried out using FEAT (FMRI Expert Analysis Tool) Version 6.00, part of FMRIB's Software Library, www.fmrib.ox.ac.uk/fsl. Z (Gaussianised T/F) statistic images were thresholded non-parametrically using clusters determined by $Z > 3.1$ and a (corrected) cluster significance threshold of $P = 0.05$ [Worsley 2001].

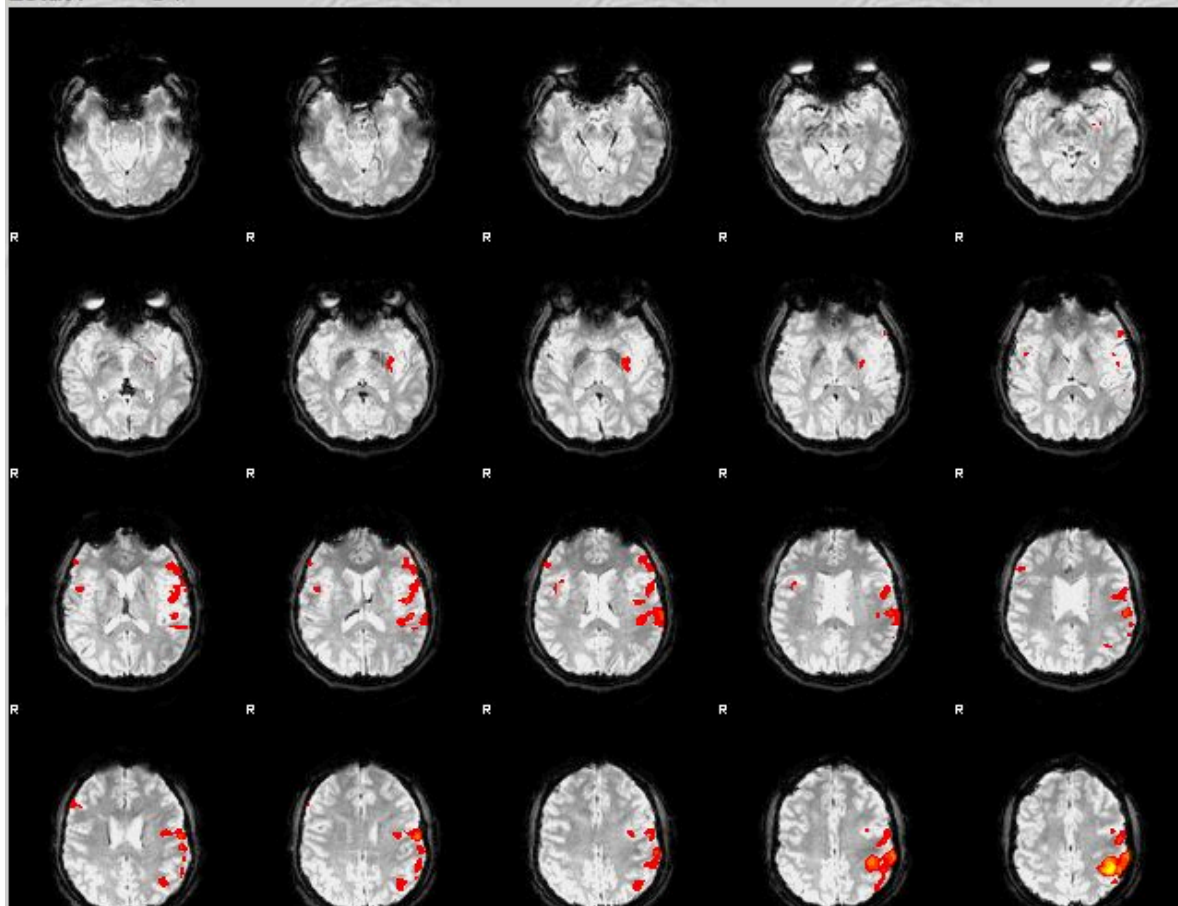
References

[Worsley 2001] K.J. Worsley. Statistical analysis of activation images. Ch 14, in Functional MRI: An Introduction to Methods, eds. P. Jezzard, P.M. Matthews and S.M. Smith. OUP, 2001.

Thresholded activation images

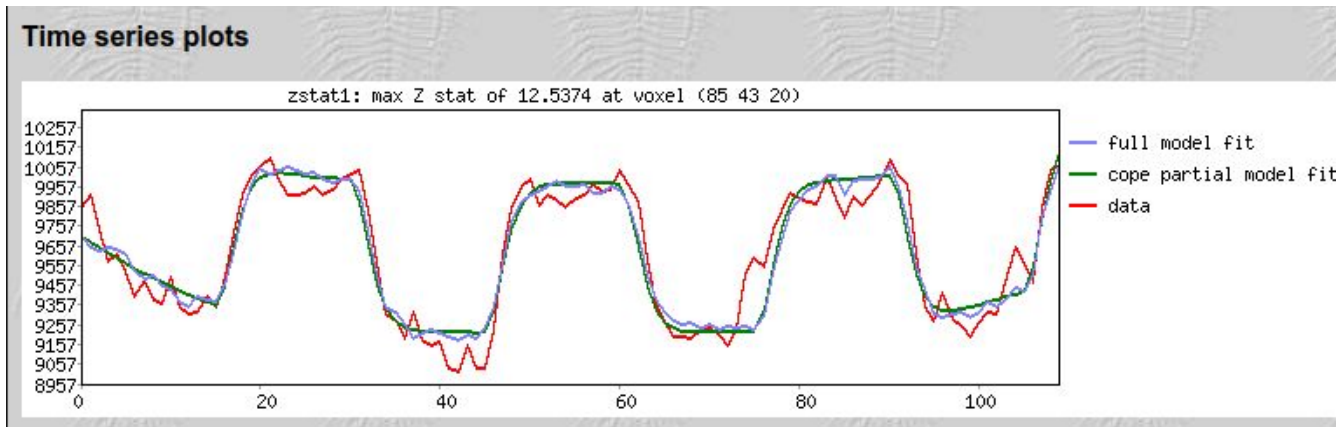
3.1  12.5

zstat1 - C1

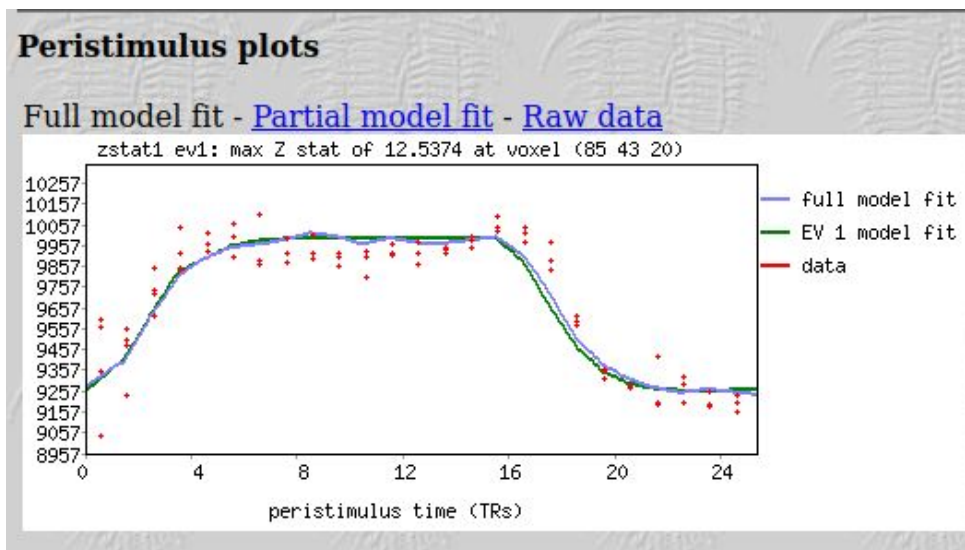


Si damos clic en el mapa de resultados, nos aparecerá una tabla que indica los clústers encontrados, su volumen, sus coordenadas, y su significancia estadística.

Regresando a la página de *post-stats*, más abajo, podemos ver la señal temporal BOLD en el voxel que mayor significancia estadística tuvo de todo el cerebro. En rojo la señal BOLD de nuestro sujeto, y en azul y verde el modelo experimental (es decir, el diseño de bloques convolucionado con la respuesta hemodinámica canónica).



Si damos clic en la serie temporal, veremos más información relevante. En el caso de modelos de bloques, es fácil ver la respuesta hemodinámica asociada al bloque, si se alinean todos los eventos para iniciar en $t=0$.



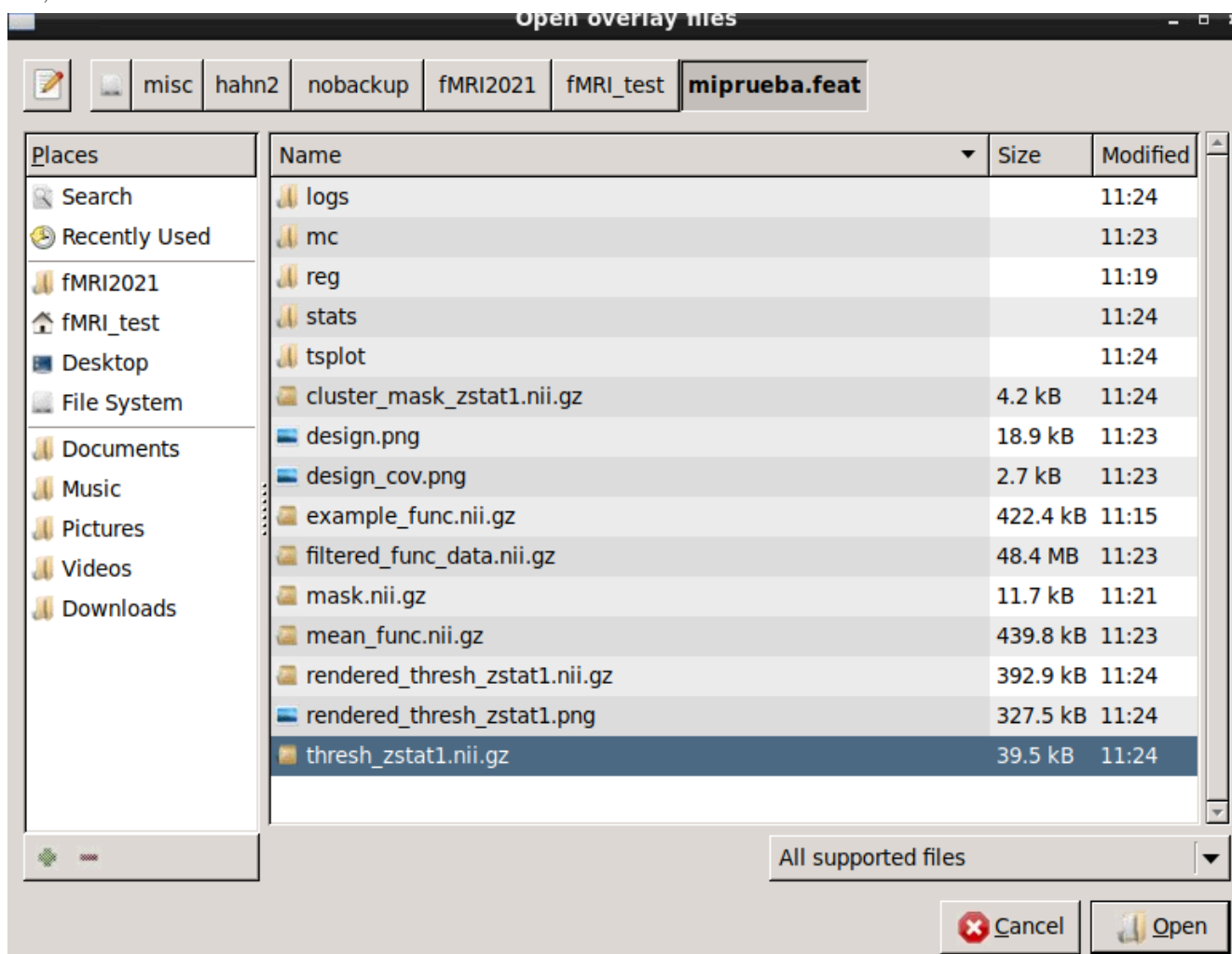
Visualización interactiva de resultados

Usaremos el programa `fsleyes`, dándole como argumento (entrada), la carpeta `.feat` que se hizo como resultado de nuestro análisis.

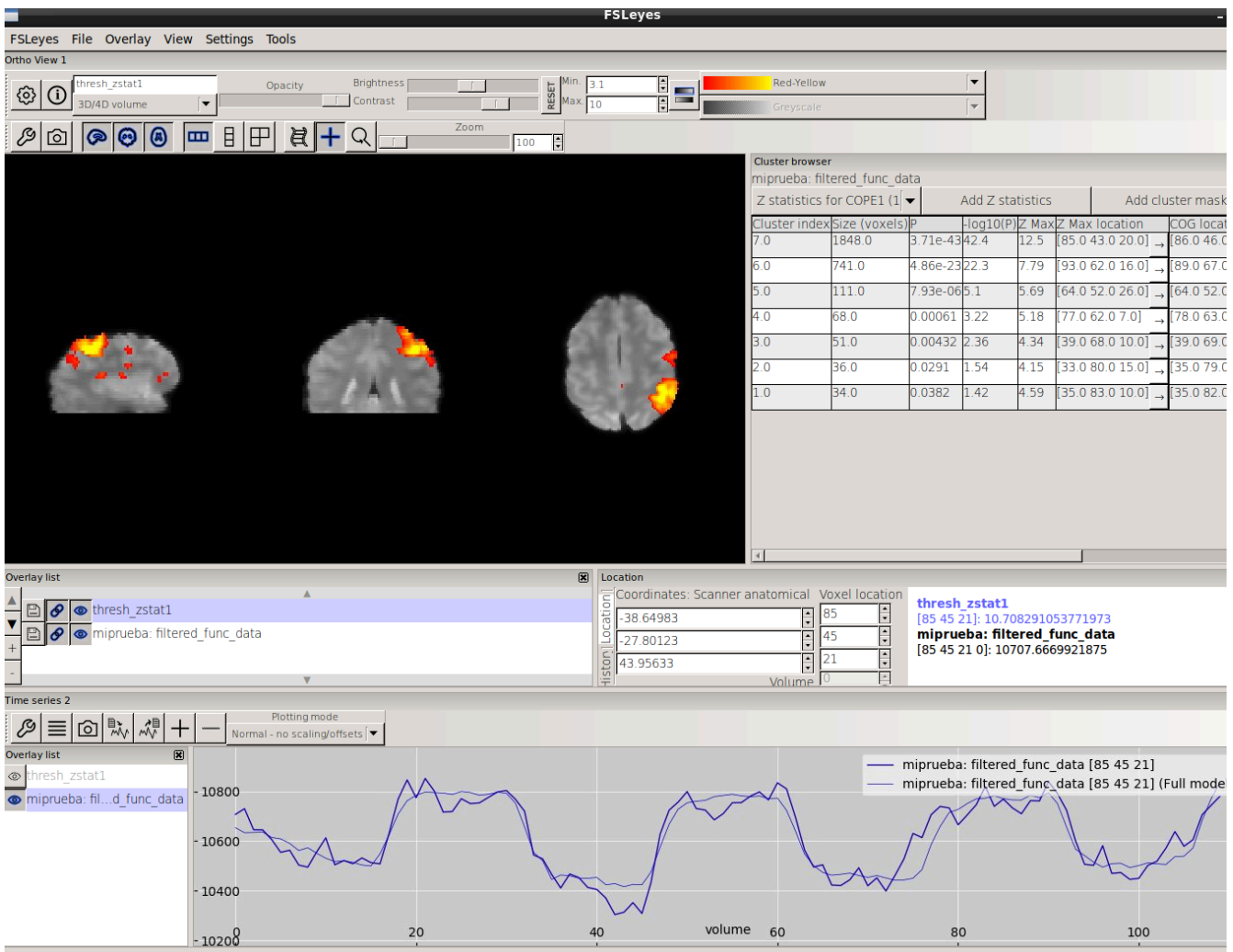
```
1 | fsleyes -s feat fMRI_test/miprueba.feat
```

👤 La opción `-s feat` le dice a `fsleyes` que vamos a querer ver los resultados de un análisis de FEAT, así que automáticamente abre varias sub-ventanas para hacernos la vida fácil.

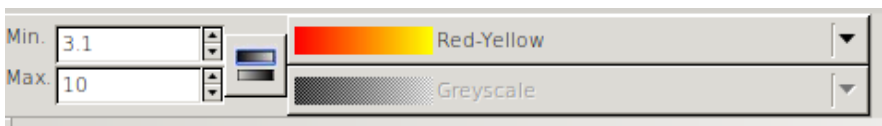
Agregaremos nuestro mapa estadístico, en `File->Add from file->`, y navegamos hacia dentro de nuestra carpeta de resultados `.feat`, a buscar el archivo `thresh_zstat1.nii.gz`.



Teniendo seleccionada la imagen `thresh_zstat1` en la sub-ventana *Overlay list*, vamos a la parte superior derecha, donde vemos los diferentes mapas de color. Seleccionamos el segundo (Red–Yellow), o algún otro que nos agrade.



Y cambiemos el rango de intensidades. Dado que lo que vemos en color es el valor z del ajuste estadístico para cada voxel, y habíamos elegido en el software que para ser considerado parte de un cluster, el voxel tendría que tener $z > 3.1$, pongamos dicho valor como mínimo. El máximo depende de los valores resultantes de nuestro mapa, y el elegido automáticamente fue el valor máximo encontrado. Podemos cambiarlo, si queremos.



Ahora podemos navegar por el volumen, y ver en tres planos el resultado, de manera interactiva. Podemos dibujar un recuadro con el botón derecho del mouse, y se hará un zoom automáticamente a esa zona. Para restablecer el zoom, podemos poner 100 en el factor zoom, o usar el botón de la lupa que tiene una flecha.

Podemos usar la tabla de los clústers que habíamos visto en la página html, pero ahora interactiva. Para ir a ver cada cluster, podemos dar clic en el ícono de la flecha que apunta a la derecha, y nos llevará al centro del clúster (técnicamente, al voxel con z más alto).

Cluster browser ✕						
miprueba: filtered_func_data						
Z statistics for COPE1 (1 ▼)		Add Z statistics		Add cluster mask		
Cluster index	Size (voxels)	P	-log10(P)	Z Max	Z Max location	COG location
7.0	1848.0	3.71e-43	42.4	12.5	[85.0 43.0 20.0] →	[86.0 46.0 20.0]
6.0	741.0	4.86e-23	22.3	7.79	[93.0 62.0 16.0] →	[89.0 67.0 15.0]
5.0	111.0	7.93e-06	5.1	5.69	[64.0 52.0 26.0] →	[64.0 52.0 25.0]
4.0	68.0	0.00061	3.22	5.18	[77.0 62.0 7.0] →	[78.0 63.0 7.0]
3.0	51.0	0.00432	2.36	4.34	[39.0 68.0 10.0] →	[39.0 69.0 11.0]
2.0	36.0	0.0291	1.54	4.15	[33.0 80.0 15.0] →	[35.0 79.0 15.0]
1.0	34.0	0.0382	1.42	4.59	[35.0 83.0 10.0] →	[35.0 82.0 11.0]