

Manual MyX

Versión 2.0 (marzo 2023)

Contenido

Instalación del programa.....	1
Preparación del entorno de trabajo.....	3
Funcionamiento del programa.....	4
Archivos resultantes del procesado	10
Preprocesado de las imágenes.....	11

Instalación del programa

1. Descomprimir la carpeta con los archivos de MyX, en el escritorio o en la ubicación desde la cual vayamos a utilizar el programa. El contenido de esta carpeta es el siguiente:

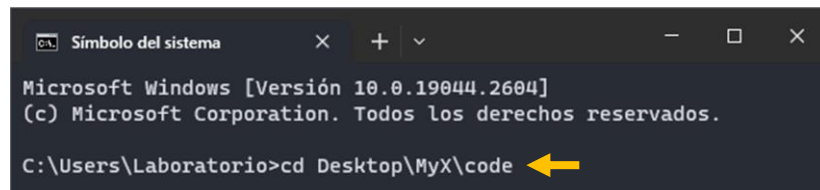
- Carpeta “**code**”: contiene los códigos del programa
- Carpeta “**myrelax**”: contiene el código de uno de los módulos que precisa el programa. Se moverá más adelante.
- El manual del programa y el archivo “colorbars.png”, en el que podemos consultar los nombres de diferentes escalas de colores para los mapas generados.

Para instalar el programa, abrimos la terminal (consola de comandos) del ordenador. En Windows, podemos acceder a ella tecleando “cmd” en el explorador de Windows.

2. En la terminal, nos desplazamos a la carpeta “**code**” de los archivos de MyX. Para ello, utilizamos el comando **cd ruta_a_code**.

MAC/Linux: `cd /Users/Laboratorio/Desktop/MyX/code`

Windows: `cd \Users\Laboratorio\Desktop\MyX\code`



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19044.2604]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\Laboratorio>cd Desktop\MyX\code
```

3. A continuación, vamos a descargar las librerías necesarias para que funcione el programa. Para ello, corremos el script `build_env` correspondiente a nuestro sistema operativo, como se muestra en el siguiente ejemplo. Tras un par de minutos se habrán instalado todas las librerías.

MAC/Linux: `sh build_env.sh`

Windows: `build_env_windows`

```
C:\Users\Laboratorio\Desktop\MyX\code>build_env_windows

C:\Users\Laboratorio\Desktop\MyX\code>python -m venv venv &&
.\venv\Scripts\activate && pip install -r requirements.txt
Collecting bruker2nifti==1.0.4
  Using cached bruker2nifti-1.0.4-py3-none-any.whl (38 kB)
Collecting cycler==0.11.0
  Using cached cycler-0.11.0-py3-none-any.whl (6.4 kB)
```

● ● ●

4. Por último, tendremos que reubicar la carpeta “**myrelax**”. Para ello, desde el explorador de archivos, la movemos a la carpeta “**site-packages**”. La ubicación de esta carpeta es la siguiente:

- **MAC/Linux:** MyX > code > venv > Lib > python3.8 > site-packages
- **Windows:** MyX > code > venv > Lib > site-packages

5. **(Optional)** Por defecto, el programa dará un output en la terminal en blanco y negro, pero es posible obtener el output con colores para facilitar la interacción con el programa. Si la terminal de nuestro ordenador permite mostrar colores (en bash, en Linux y MAC es posible, en Windows se puede descargar en la store la aplicación Windows terminal), para activar esta opción, tenemos que abrir con un editor de texto el script “**utils.py**” y comentar las líneas de output en blanco y negro (poner “#” delante) y descomentar las de output en color (quitar el “#”).

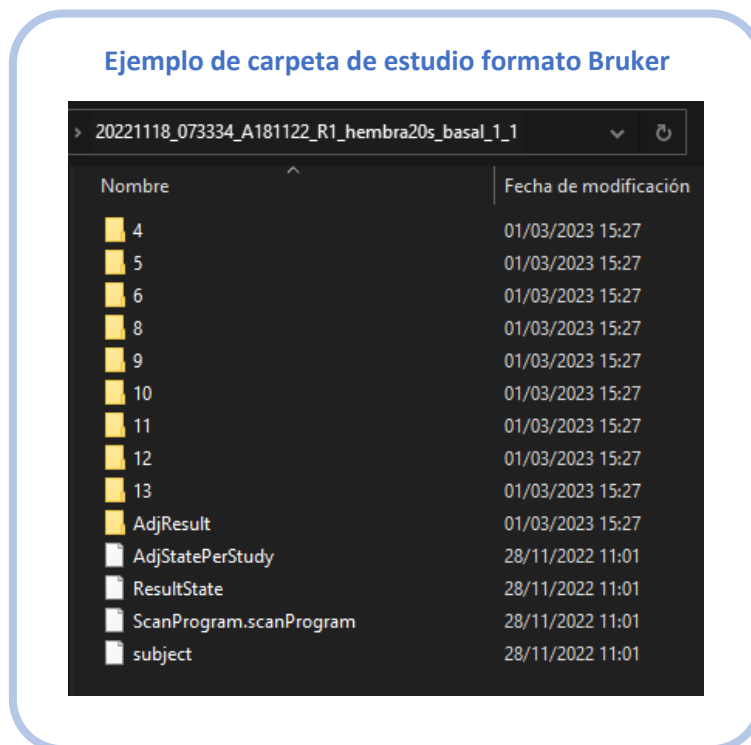
Script `utils.py` con modo a color activado

```
11 # --- Modo de output ---
12 class Headermsg:
13     # Output en blanco y negro
14
15     # info = '[INFO]'
16     # warn = '[WARNING]'
17     # error = '[ERROR]'
18     # success = '[SUCCESS]'
19     # pointer = '>>>'
20     # ask = '(?)'
21     # welcome = '\n-----\n' + \
22     #         '-- Welcome to MyX --\n' + \
23     #         '-----\n'
24     # new_patient1 = '* STUDY *'
25     # new_patient2 = '* STUDY *'
26     # new_modal = '> MODAL >'
27
28
29     # Para output en color, comentar las líneas anteriores a esta y descomentar las siguientes
30
31     info = '\x1b[0;30;44m [INFO] \x1b[0m'
32     warn = '\x1b[0;30;43m [WARNING] \x1b[0m'
33     error = '\x1b[0;30;41m [ERROR] \x1b[0m'
34     success = '\x1b[0;30;42m [SUCCESS] \x1b[0m'
35     pointer = '\x1b[5;36;40m>>>\x1b[0m'
36     ask = '\x1b[0;30;46m ? \x1b[0m'
37     welcome = '\n\x1b[0;30;46m \x1b[0m\n' + \
38             '\x1b[0;30;46m \x1b[0m \x1b[0;30;46m \x1b[0m\n' + \
39             '\x1b[0;30;46m \x1b[0m \x1b[0;36;40mWelcome to MyX\x1b[0m \x1b[0;30;46m \x1b[0m\n' + \
40             '\x1b[0;30;46m \x1b[0m \x1b[0;30;46m \x1b[0m\n' + \
41             '\x1b[0;30;46m \x1b[0m\n'
42     new_patient1 = '\x1b[0;30;47m * STUDY *'
43     new_patient2 = '* STUDY * \x1b[0m'
44     new_modal = '\x1b[0;30;47m > MODAL > \x1b[0m'
45 # -----
```

Preparación del entorno de trabajo

Una vez instalado el programa, podemos empezar a utilizarlo. Pero antes, debemos preparar una carpeta en la que colocar los estudios que queremos procesar y en la que se guardarán los archivos resultantes de dicho procesamiento. En lo sucesivo, nos referiremos a esta carpeta como “**main**”, pero se le puede poner otro nombre. Esta carpeta puede estar ubicada en cualquier lugar, pero es preferible que esté a mano para facilitar el acceso.

En la carpeta “**main**” tendremos por tanto las carpetas con los **estudios**. Se procesarán tantos estudios como haya en la carpeta. Todos ellos deberán tener formato *bruker*, por lo que tendrán una estructura similar a la indicada en la imagen.



Además, durante la ejecución del programa se generarán automáticamente las siguientes carpetas:

- Carpeta “**convertidos**”: contendrá los estudios originales convertidos a formato NIfTI. Las subcarpetas de un estudio con posible interés para el procesamiento tendrán un prefijo para poder identificarlos mejor (DT, MT, T1, T2 o T2E).
- Carpeta “**procesados**”: contendrá los resultados del procesamiento. Dentro de cada estudio habrá una subcarpeta con los resultados para cada modalidad procesada. Más adelante se especificará en qué consiste cada archivo en estas carpetas según la modalidad.
- Carpeta “**supplfiles**”: se genera para guardar archivos temporales durante la ejecución. No es necesario hacer nada con ella.

Funcionamiento del programa

Durante la ejecución del programa, tendremos que interactuar mediante la terminal y en algunas ocasiones mediante ventanas emergentes que aparecerán en pantalla. Si en algún momento quisiésemos interrumpir y parar el programa, podemos salir pulsando **Ctrl + C**.

1. (Inicio) Para iniciar el programa, abrimos la terminal y, del mismo modo que se especifica en el segundo paso de la instalación, nos movemos a la carpeta “**code**” (MyX > code). Desde allí, lanzamos el script run_myx correspondiente a nuestro sistema operativo, como se muestra a continuación.

MAC/Linux: sh run_myx.sh

Windows: run_myx_win



```
C:\Users\Laboratorio\Desktop\MyX\code>run_myx_win
C:\Users\Laboratorio\Desktop\MyX\code>.\env\Scripts\activate
&& python -u main.py

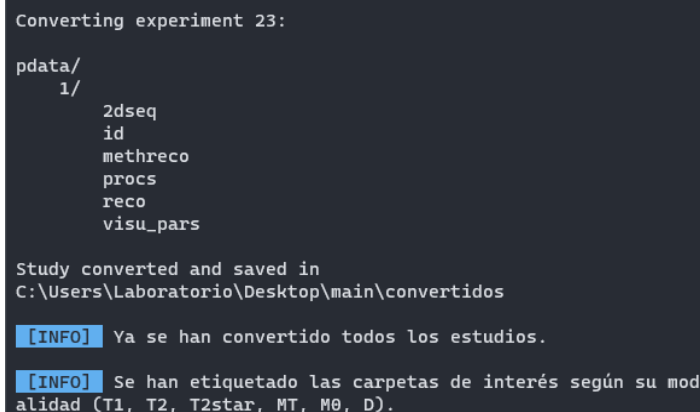
Welcome to MyX

? Selecciona la carpeta de trabajo en la ventana emergente.
```

2. (Selección de carpeta) Al iniciar el programa, se mostrará un mensaje de bienvenida en la terminal (como se muestra en el ejemplo anterior) y se abrirá una ventana en la que tendremos que elegir nuestra carpeta “**main**” y pulsar “Seleccionar carpeta”.

3. (Conversión a NIfTI) Una vez elegida la carpeta, se empezarán a convertir los estudios a formato NIfTI de manera automática (en el caso de que no hayan sido convertidos antes en otra ejecución y estén ya guardados). En la terminal aparecerá información similar a la mostrada en el ejemplo a continuación, pero esta no es relevante para el usuario. Cuando termine el proceso,

Conversión automática de ficheros a formato NIfTI



```
Converting experiment 23:

pdata/
  1/
    2dseq
    id
    methreco
    procs
    reco
    visu_pars

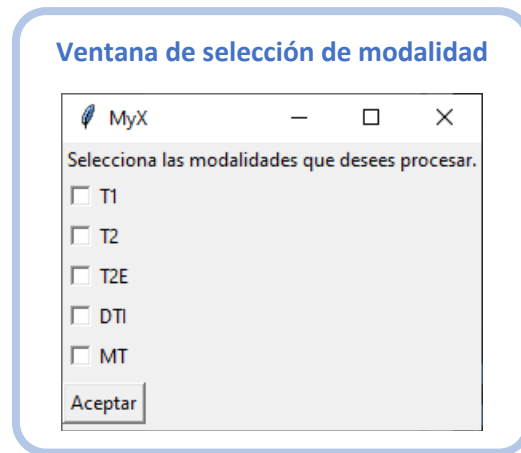
Study converted and saved in
C:\Users\Laboratorio\Desktop\main\convertidos

[INFO] Ya se han convertido todos los estudios.

[INFO] Se han etiquetado las carpetas de interés según su modalidad (T1, T2, T2star, MT, M0, D).
```

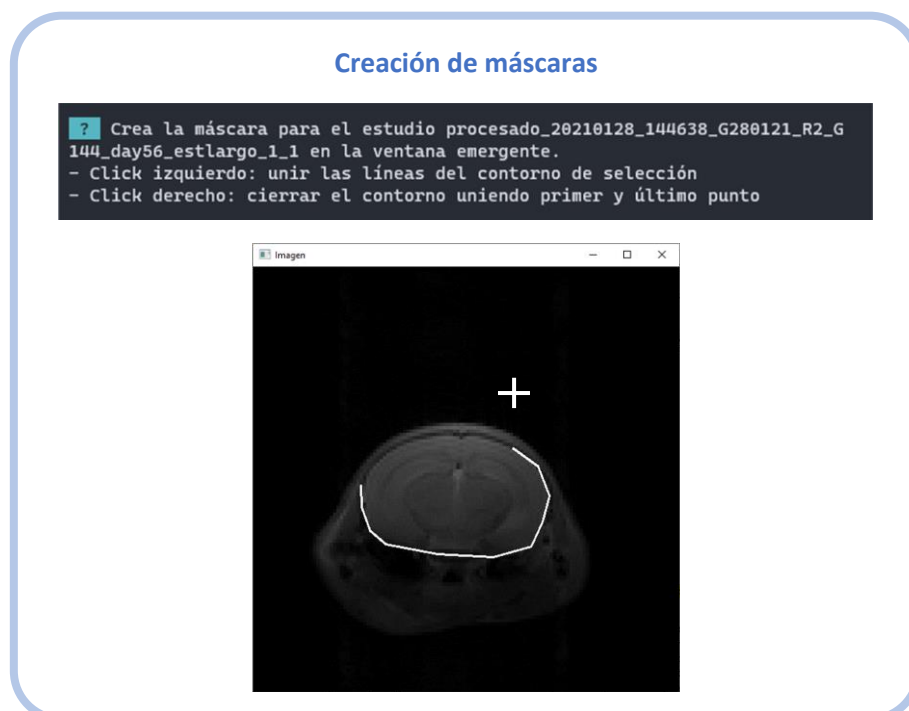
se mostrará un mensaje indicando que se han convertido todos los estudios. Si queremos acceder a estos archivos, estarán guardados en la carpeta **“convertidos”**.

4. (Selección de modalidad) A continuación, aparecerá una ventana en la que podremos elegir las modalidades que queramos procesar. Se procesarán, para cada estudio, las modalidades que hayamos elegido (en el caso de que estén disponibles).



En caso de que una modalidad seleccionada haya sido ya procesada para un estudio, se mostrará un mensaje dando la opción de volverla a procesar. Hay que tener en cuenta que volverla a procesar supone borrar los resultados del primer procesado. Por eso, si queremos conservarlos, habrá que moverlos a otra carpeta antes de continuar (obtendremos un mensaje de advertencia antes de continuar para tener tiempo de hacerlo).

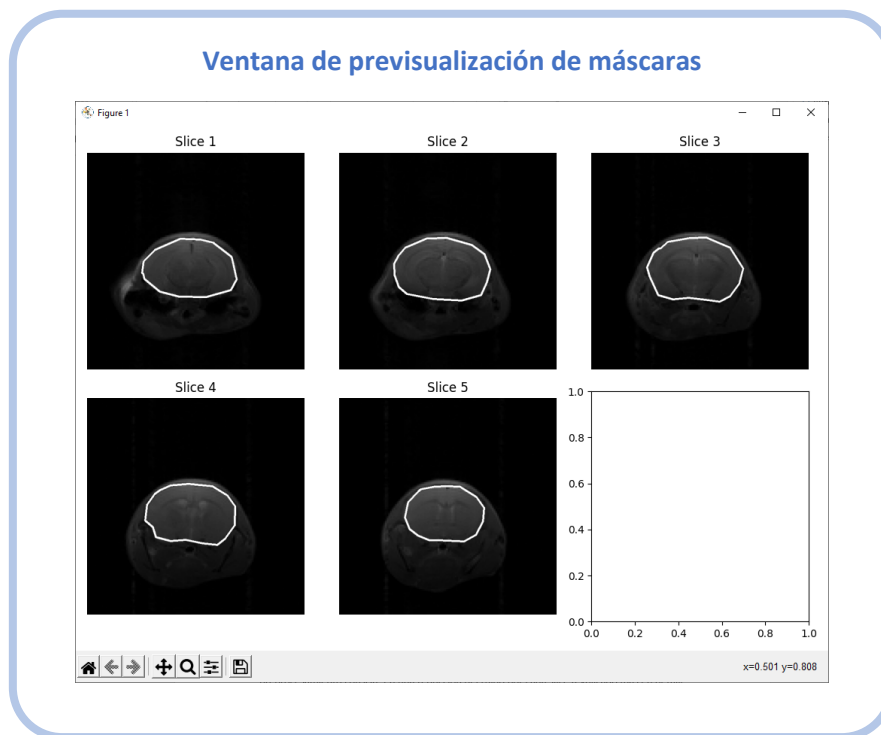
5. (Creación de las máscaras) Llegados a este punto, comenzará el procesamiento de los estudios correspondientes. Se irá anunciando cuando se empiece con cada estudio, y con cada modalidad dentro de éste.



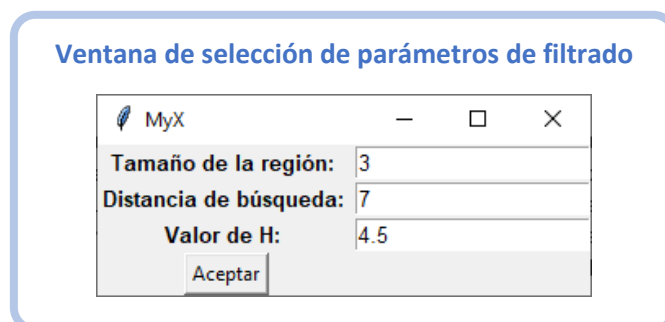
El primer paso para cada instancia será crear las máscaras o ROIs donde queramos que se realice el procesamiento. En el caso de crear una nueva máscara, se irán mostrando ventanas emergentes para cada slice, y seguiremos los pasos mostrados en la terminal.

Si ya se ha procesado el estudio alguna vez, y está disponible su máscara correspondiente, se preguntará si queremos reutilizarla, para no tener que crearla de nuevo.

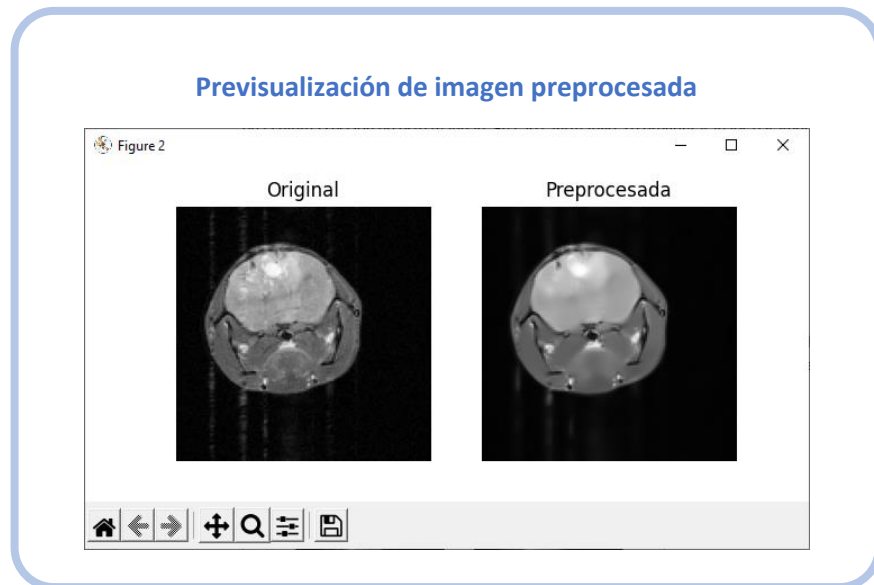
Después de crear todas las máscaras, se mostrará una ventana emergente con la previsualización de todas las slices (mostrado a continuación). Una vez visto, pulsar **enter** (estando la ventana seleccionada, podemos hacer clic encima antes para asegurarnos), y se preguntará en la terminal si es correcta. Si no nos convencen las máscaras creadas, podemos repetir el proceso tantas veces como sea necesario.



6. (Preprocesamiento) Antes de comenzar el procesamiento, tendremos la opción de preprocesar las imágenes antes de generar el mapa paramétrico. En concreto, el preprocesamiento consiste en una reducción del ruido por cada slice (explicado más en detalle en la última sección de este manual). En caso afirmativo, se preguntarán los parámetros para el filtrado. El primero y el segundo deben ser números enteros, mientras que el último puede ser tanto entero como decimal.



A continuación, aparecerá una ventana que mostrará la imagen original y la imagen preprocesada, a la vez que se preguntará si está todo correcto a través de la terminal. En caso de que no se estemos satisfechos con el resultado, se puede repetir el preprocesamiento tantas veces como se desee.



7. (Procesamiento) Tras estos pasos, comenzará el procesamiento de la modalidad correspondiente. Según la modalidad, se irán sucediendo diferentes pasos:

- **DTI**

En este caso, en primer lugar, se preguntará al usuario los parámetros de adquisición de las imágenes (número de b valores, número de imágenes basales y número de direcciones).

Especificación de parámetros DTI

```
? ¿Número de b valores para cada dirección?
>>> 3

? ¿Número de imágenes basales?
>>> 3

? ¿Número de direcciones?
>>> 13

[WARNING] Has introducido un número de b valores diferente al adquirido.
Has especificado 3, cuando se adquirieron 2. Se va a continuar con el valor real.

[INFO] Número de imágenes basales correcto.

[WARNING] Has introducido un valor de direcciones diferente al adquirido.
Has especificado 13, cuando se adquirieron 15.

? ¿Deseas eliminar 2 direcciones? [y/n]
>>> y

[INFO] Se van a eliminar 2 direcciones.

? Por favor, especifica las direcciones que desees eliminar.
(1) >>> 2
(2) >>> 5
```

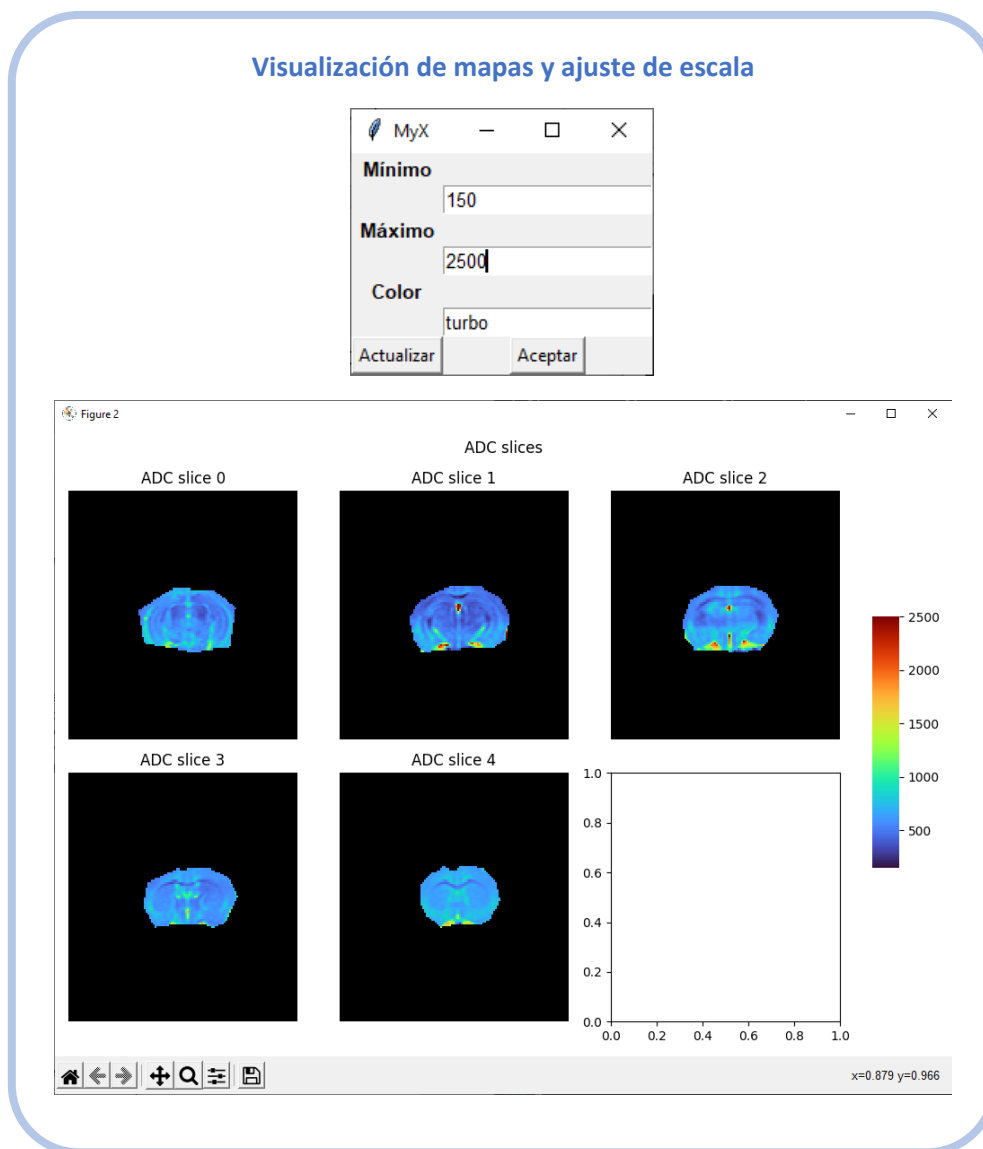
En el caso de las imágenes de MT, no hace falta especificar ningún parámetro previamente. Es posible que haya varias imágenes diferentes en el estudio original, si hemos modificado el slope. Si es así, el programa preguntará qué carpeta o carpetas deseamos procesar (normalmente, la 1 será la original y las siguientes serán las que tengan la slope modificada). Si sólo hay una carpeta, se procesará directamente. El procesamiento de estas imágenes es muy rápido, porque no requiere realizar el ajuste a ninguna ecuación. A continuación, podremos guardar el resultado.

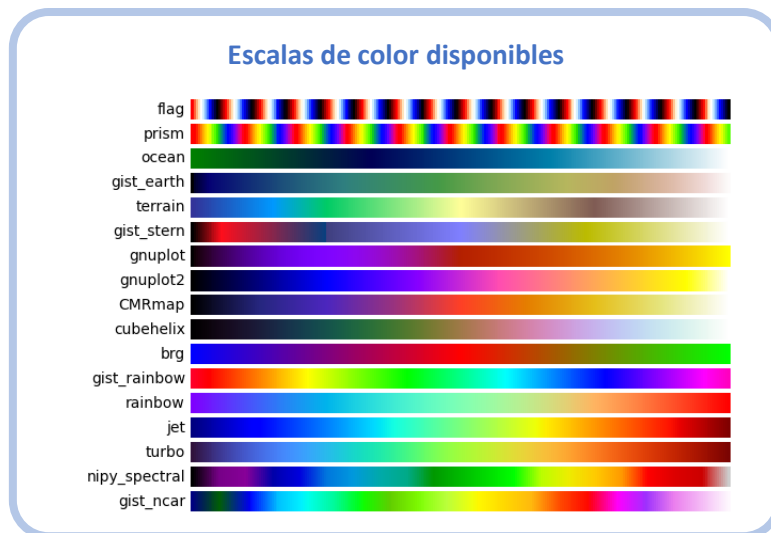
- **Mapa de T1/T2/T2***

Estos mapas no necesitan ninguna especificación adicional antes de comenzar el procesamiento. Una vez realizado, se preguntará si se desea usar un filtro de R^2 y se podrán guardar los mapas.

Guardado de mapas: cada vez que se genere un mapa o un resultado, en cualquiera de las modalidades, se abrirá una ventana emergente mostrando el resultado. Además, se permitirá modificar la escala de colores en la que se muestra, especificando el valor mínimo y máximo, así como el nombre de la paleta de colores. Los distintos mapas de colores y sus nombres se encuentran en el archivo “colorbars.png” (además de en este mismo manual). Los recomendados son “turbo”, que está por defecto, y “jet”. Podremos probar diferentes combinaciones y visualizándolas pulsando el botón “Actualizar”. Una vez estemos satisfechos, pulsaremos aceptar para guardar el mapa.

Cuando se hayan procesado todos los mapas que se hayan seleccionado de todos los estudios incluidos en la carpeta “main”, el procesamiento habrá terminado.





Archivos resultantes del procesamiento

A continuación se describe la ubicación y contenido de algunos de los archivos más relevantes que se producen tras utilizar el programa.

convertidos > convertido_estudio > convertido_estudio_subcarpeta

- **convertido_estudio_subcarpeta_(subscan).nii.gz** -> imágenes adquiridas originalmente. Si hay más de una, normalmente nos interesará la subscan_0, excepto en la modalidad de MT, en la que cada una corresponderá a las diferentes slopes (en orden de adquisición).
- **convertido_estudio_method.txt** -> parámetros de adquisición.
- **acquisition_method.txt** -> nombre secuencia.

procesados > procesado_estudio > modalidad_procesado_estudio_subcarpeta

En estas carpetas encontraremos de nuevo los archivos descritos anteriormente, que se copian a esta carpeta durante el procesamiento. Sin embargo, el archivo de las imágenes, en caso de haber realizado el preprocesado, estará filtrado. Si no, serán las originales. El resto de archivos que encontraremos en esta carpeta variará según la modalidad:

DTI (procesado realizado usando el paquete [Dipy](#))

- **procesado_estudio_subcarpeta_DwEffBval** -> vector de b valores efectivos
- **procesado_estudio_subcarpeta_DwGradVec** -> matriz de direcciones
- **ADC_map.nii** -> mapa de ADC
- Carpetas “Dir_x”:
 - Imágenes png del mapa de ADC para la dirección x
 - **R2_map.nii** -> mapa de R^2 (bondad de ajuste) para la dirección x del mapa
- Carpeta “AD” (Axial Diffusivity):
 - Imágenes png del mapa de AD
 - **AD_map.nii** -> mapa de AD
- Carpeta “RD” (Radial Diffusivity):
 - Imágenes png del mapa de RD
 - **RD_map.nii** -> mapa de RD

- Carpeta “MD” (Mean Diffusivity):
 - Imágenes png del mapa de MD
 - **MD_map.nii** -> mapa de MD
- Carpeta “FA” (Fractional Anisotropy):
 - Imágenes png del mapa de FA
 - **FA_map.nii** -> mapa de FA

MT (si hemos procesado más de una, con diferentes slopes, los siguientes archivos se encontrarán dentro de una carpeta cuyo nombre será el número original del orden de la adquisición)

- Imágenes png del mapa de MT
- **MT_map.nii** -> mapa de MT

T1/T2/T2* (procesado realizado utilizando del paquete [MyRelax](#))

- Carpeta “mapas”:
 - Imágenes png del mapa de T1/T2/T2*
 - **R2_map.nii** -> mapa de R^2 (bondad de ajuste) del mapa
 - **..._TxyME.nii** -> mapa de T1/T2/T2*
 - **..._SSEME.nii** -> fitting sum of squared errors
 - **..._SOME.nii** -> T1w proton density, with receiver coil field bias
 - **..._ExitME.nii** -> exit code (1 = successful fitting, 0 = background, -1 = unsuccessful fitting)

Además, encontraremos las máscaras para seleccionar las ROIs tanto en formato NIfTI (**mask.nii**) como en imágenes para cada slice, para poder visualizarlas superpuestas a la imagen de resonancia (**shape_slice_x.png**). Estos archivos se encontrarán en la modalidad en la que se hayan creado. También se guarda el archivo NIfTI en la carpeta general del estudio procesado, para los estudios en los que se reutiliza la máscara.

Preprocesado de las imágenes

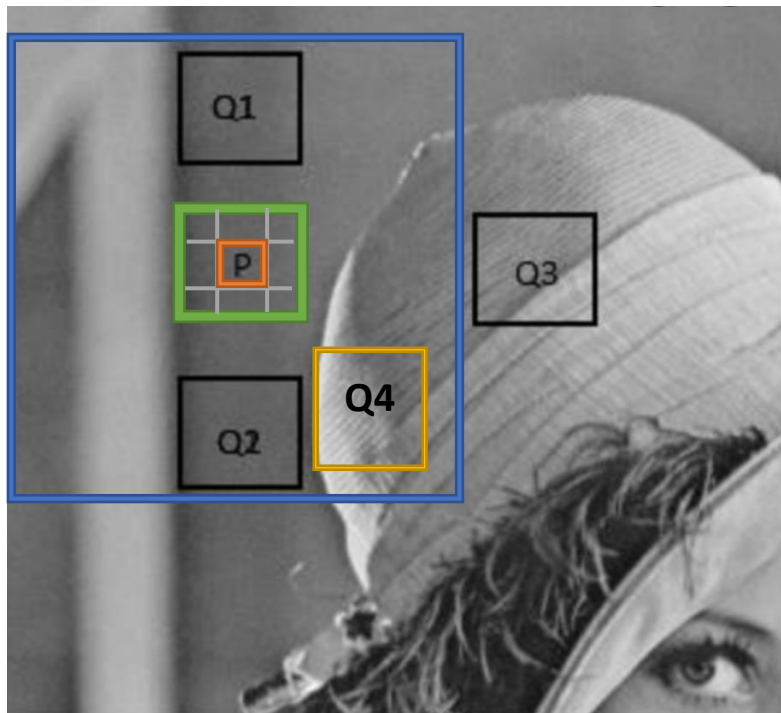
El preprocesado realizado incluye únicamente una reducción del ruido en la imagen empleando el algoritmo *non-local means*. Este algoritmo se basa en remplazar el valor de intensidad de un píxel con el valor promedio de las intensidades de píxeles similares. Como esos píxeles similares no necesariamente tienen que estar cerca del píxel objetivo, este algoritmo busca por toda la imagen (por ello es no local).

Como buscar por toda la imagen supone un coste computacional elevado, generalmente se trabaja por vecindarios de píxeles. Para cada píxel objetivo (píxel al que se va a reemplazar su valor de intensidad) se toma:

- Un tamaño de región T alrededor del píxel objetivo.
- Una zona de búsqueda a una distancia D en píxeles que defina un vecindario alrededor del píxel objetivo. En este vecindario se van a buscar regiones de un tamaño T con las que poder promediar.
- Una distancia de intensidad H que permitirá promediar aquellos píxeles que tengan un valor de intensidad similar al píxel objetivo. Sirve a modo de valor de tolerancia.

En la imagen inferior se muestra, en azul, el vecindario definido por una distancia D; en naranja, el píxel objetivo P; y en verde, una región con un tamaño T de 3x3 alrededor de P. El algoritmo

lo que hace es ir situando una región de tamaño T alrededor de cada píxel Q que se encuentre dentro del vecindario y que esté en el rango de valores permitido por H . En la imagen, se observa que $Q1$ y $Q2$ servirán para obtener el nuevo valor de P , pero $Q3$ no porque esta fuera del vecindario. En el caso de $Q4$, como los valores de intensidad de su región son distintos a los de P , será una región que también se descarte. Para obtener el nuevo valor de P , se hará una suma ponderada de los valores promedio de los píxeles contenidos en cada una de las regiones Q , donde cada región Q tendrá un peso asociado en función de un valor de distancia (distancia de color, es decir, de cómo de similares son los valores de gris) entre ella misma y la región P .



En el programa, se han de introducir 3 parámetros:

- Tamaño de la región, que hace referencia a T . Ha de ser un entero, porque con este parámetro se forman regiones de tamaño $T \times T$ píxeles. Por ejemplo, 3×3 .
- Distancia de búsqueda, que hace referencia a D . Ha de ser un entero, porque con este parámetro se forman regiones de tamaño $D \times D$ píxeles. Por ejemplo, 7×7 .
- Valor de H . Puede ser entero o decimal. Cuanto mayor sea, más permisivo se es a la hora de incluir regiones con intensidades distintas, por lo que la imagen quedará más difuminada/emborronada.

Es recomendable hacer pruebas al principio para ver cuáles son los valores que más convienen en cada estudio e, incluso, en cada modalidad. Por ejemplo, si se trabaja con imágenes de un tamaño mayor, será conveniente aumentar la distancia de búsqueda o el tamaño de región.

El artículo que describe el método se encuentra [aquí](#). La implementación en código de este algoritmo se ha empleado la librería SciPy. Para más información acudir [aquí](#) y [aquí](#).