

NOEL FCD TEXTURE PIPELINE - VERSION 2.0

This brochure includes the instructions for installing and using the texture pipeline developed at the Neuroimaging of Epilepsy Laboratory (<https://noel.bic.mni.mcgill.ca>). You are granted access to this pipeline because you attended the International Training Course on Neuroimaging of Epilepsy.

The name of the pipeline is **NoelTexturesPy**. **NoelTexturesPy** generates 3D volume-based *Relative Intensity* and *Gradient Magnitude* maps derived from 3D T1-weighted MRI with isotropic resolution for computer-aided detection of focal cortical dysplasia (FCD).

Disclaimer

- 1) *This pipeline **should be used only when a focal cortical dysplasia type II is suspected. It is not recommended to use this pipeline for the detection of any other epileptogenic lesions, such as hippocampal sclerosis, heterotopias or tumors.***
- 2) *The use of this pipeline for clinical decision-making is not the responsibility of NOEL. The intended use is solely the responsibility of the operator.*
- 3) *Before you make use of the pipeline, please read the paper that describes the methodology and the interpretation: "Texture analysis and morphological processing of magnetic resonance imaging assist detection of focal cortical dysplasia in extra-temporal partial epilepsy" (Bernasconi et al., Annals of Neurology, 2001). See pdf at the end of the handbook (page 20).*
- 4) *Cases presented in this document (page 15) should serve as didactic material only and not be used for other purposes. They should not be shared or published in any form.*

General technical information

NoelTexturesPy also co-registers 3D T2-weighted MRI (or FLAIR) to the T1-weighted MRI. In other words, at the end of the processing pipeline, all maps and MRIs are co-registered to a common, stereotaxic space. This guarantees anatomical correspondence.

To facilitate the use of this tool, we chose to rely on Docker, a service that delivers software across multiple operating systems (Windows, MacOS and Linux) in packages called *containers*.

The user needs to proceed through the following steps:

1. Install Docker and **NoelTexturesPy** (see details below).
2. After installation, **NoelTexturesPy** will run as a website, accessible via a web-browser window (e.g., Firefox or Google-chrome). The pipeline is installed on the user's computer and does not request nor send any information to the internet. The user will have to "upload" a T1-MRI to the website, wait for the processing to run in the background, and "download" the texture maps.
3. To review the MRI and maps simultaneously, we recommend installing *Register*. If you are not able or do not wish to use *Register* (see installation page 10), we recommend the use of *ITK-SNAP*. Alternatively, you can use any viewer of your choice.

1. INSTALLATION

1.1 Installation of Docker

Before installing NoelTexturesPy, you need to install *Docker*. Depending on your operating system (OS), the procedure might be slightly different.

WARNING. It is essential to use a recent version of your operating system. Docker used to have a version for older systems (Docker Toolbox), but it is no longer available, so no longer maintained or upgraded. If you cannot update the OS to its most recent version, your technical support with Docker will be limited.

1.1.1 MacOS

To know which version of MacOS you are running, please click on the “Apple” menu icon (top left of the screen) and go to “About This Mac”. You’ll see the MacOS version name (e.g., Mojave), followed by its version number (official instructions by Apple can be found at the following address: <https://support.apple.com/en-us/HT201260>).

1.1.1.1. MacOS 10.14 or higher

You can download *Docker* from: <https://desktop.docker.com/mac/stable/amd64/Docker.dmg>.

Installation is like for any other MacOS Application: double-click on Docker.dmg, drag and drop Docker.app into the Application folder and then double-click on Docker.app in the Applications folder to start the program (<https://docs.docker.com/docker-for-mac/install/>).

1.1.1.2. MacOS 10.8 to 10.13

Download the link:

<https://github.com/docker/toolbox/releases/download/v19.03.1/DockerToolbox-19.03.1.pkg>

WARNING: Docker Toolbox is discontinued. No further support is provided.

1.1.2 Windows

To know which version of Windows your computer is running, type the following:
<https://support.microsoft.com/en-ca/help/13443/windows-which-version-am-i-running>

1.1.2.1. Windows 10

Download the link:

<https://download.docker.com/win/stable/Docker%20Desktop%20Installer.exe>

Installation is done following the installation wizard, as any other windows program.
Windows 10 Home user should also follow the instructions contained in the following link:
<https://docs.docker.com/docker-for-windows/install/>

1.1.2.2. Windows 8.1 and 7

Download the link:

<https://github.com/docker/toolbox/releases/download/v19.03.1/DockerToolbox-19.03.1.exe>

Installation is done by following the installation wizard, as for any Windows program.

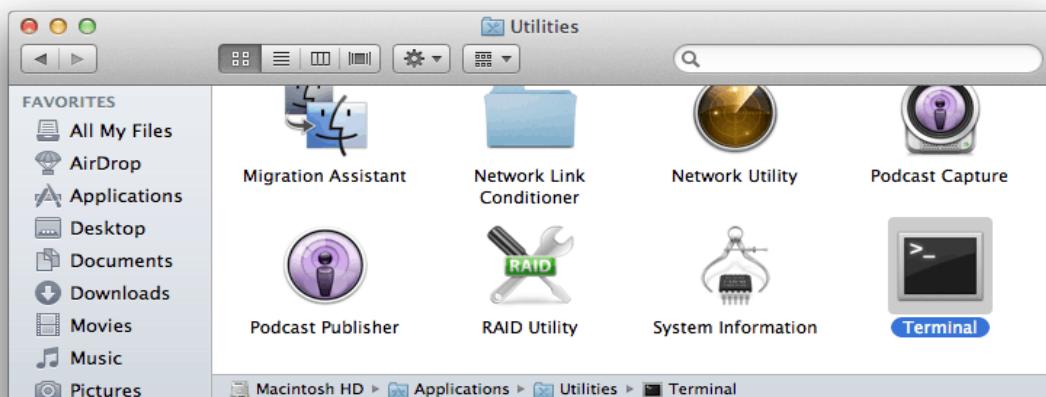
WARNING: Docker Toolbox is deprecated. No further support is provided.

1.2. Launch a terminal window

If you are running MacOS 10.12 or Windows 10 Pro, Enterprise or Education, you can open a terminal.

1.2.1. MacOS 10.14 or higher

You will find the terminal application in “Application/Utilities”.



1.2.2. MacOS 10.8 to 10.13

Open the launch pad and look for the Docker Quickstart Terminal.



When started, the terminal will run several programs in background and display its activities. Let it run as it sets a clean terminal environment for you.

1.2.3 Check the default shell on MacOS

Something important to consider is the default **shell** used in the terminal. A shell is a user interface to the various operating system services. The terminal is the program that opens a graphical window providing access to the shell in the form of a *command-line interface* (CLI). The most common shells are: sh (Bourne Shell), bash (Bourne Again shell), zsh (Z-shell), csh (C-shell), ksh (Korn-shell), etc.

Important. The rest of this document assumes that the default shell used for MacOS is *bash*.

To check which shell is the current default, open the ‘Terminal’ application and type the following command: echo \${SHELL}

```
[benoit@Avishai:~$ echo ${SHELL}
/bin/bash
benoit@Avishai:~$ ]
```

If the result of the command is: /bin/bash, you are good to go.

If something else appears, you must change the default bash by typing: **chsh -s /bin/bash**

```
Last login: Mon Jun  7 13:56:42 on ttys001
[benoit@Avishai ~ % chsh -s /bin/bash
Changing shell for benoit.
[Password for benoit:
benoit@Avishai ~ % ]
```

As a change in setting the default bash is an administration task, you will be asked to enter your MacOS administrator password in the terminal. In order to effectively modify the bash, after you enter the password, you still need to **close the terminal, reopen it and verify that the default shell is bash**.

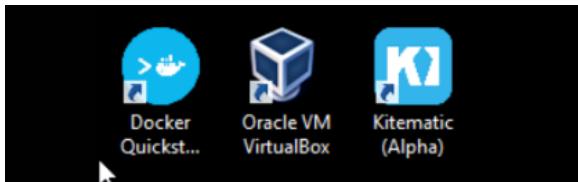
1.2.4. Windows 10

Make sure *Docker* is run with administrator privileges. To do so, right-click in task bar, use “run as administrator”. Press Windows Button + X to open power user menu and click on “Command Prompt”.



1.2.5. Windows 7 and 8.1

On your desktop, locate the Docker Quickstart Terminal icon and double click on it.



1.3. Verification that Docker is running on the terminal window

Type: docker version and/or docker info or docker run hello-world as shown below:

```
[benoit@Avishai:~/Boulot/PostDoc/08_SoftWares$ docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Congratulations - You went through the hardest part!

1.4. Installation and running of NoelTexturesPy

To install NoelTexturesPy, type: docker pull noelmni/pynoel-gui-app:latest.
Download needs to be performed only once.

```
[benoit@Avishai:~/Boulot/PostDoc/08_SoftWares$ docker pull noelmni/pynoel-gui-app:latest
latest: Pulling from noelmni/pynoel-gui-app
bf5952930446: Pull complete
385bb58d08e6: Pull complete
ab14b629693d: Pull complete
7a5d07f2fd13: Pull complete
25a245937421: Pull complete
4ecbe2605027: Pull complete
cec7ecad6125: Downloading [=====] 16.62MB/88.81MB
5d0edb70e87: Downloading [=====] 10.66MB/15.62MB
ad85070c4a5b: Download complete
430c0ad4b747: Download [>] 2.124MB/303.6MB
d62ea9637c73: Waiting
bdb37f99f666: Waiting
8a6c2b2232ea: Pulling fs layer
9bb5b43c3ab1: Waiting
46885987e63c: Waiting
8e424ece72e8: Waiting
17c38a98c710: Waiting
3b41e1a3a6d5: Waiting
5813957cdc39: Waiting
93ccde180a9a: Waiting
```

You can now run noelPyTextures from the terminal or from Docker Desktop.

1.4.1. Start noelPyTextures from the terminal

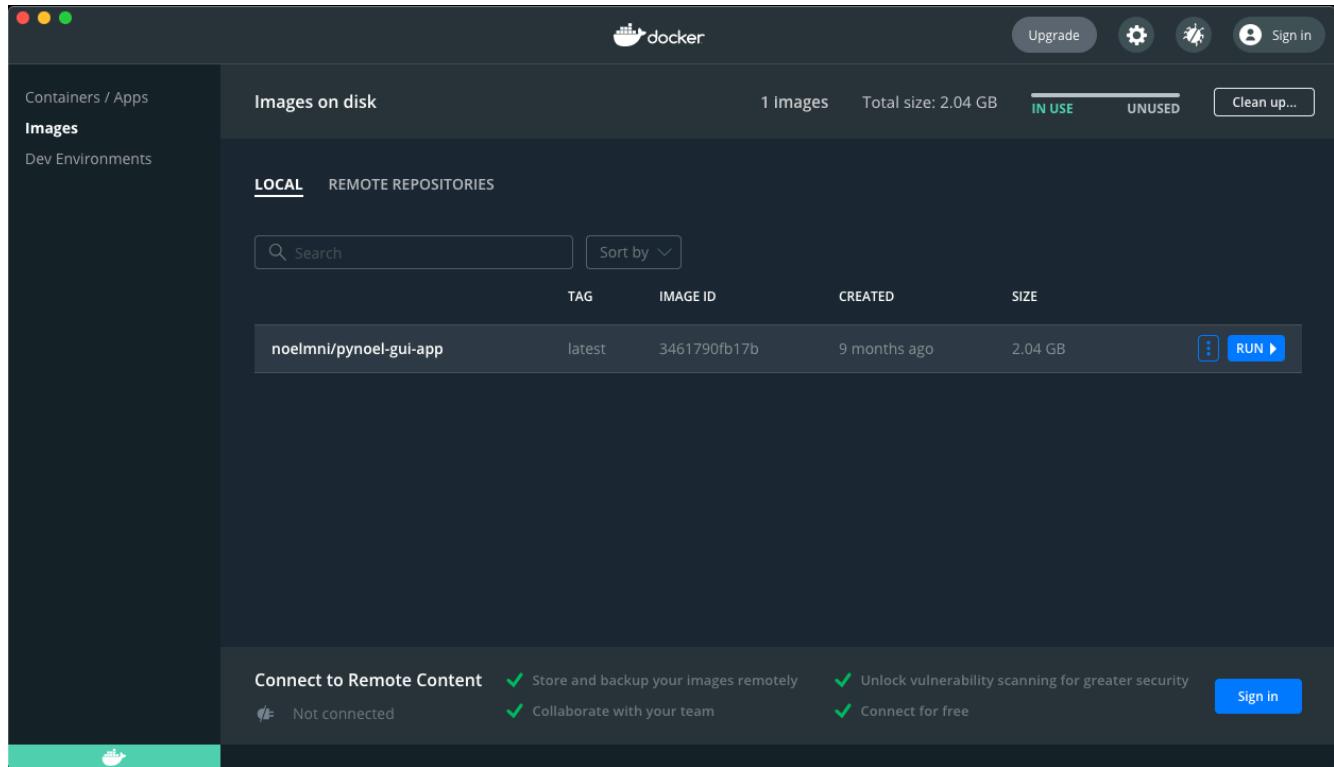
To run the application, type: `docker run --rm -p 9999:9999 noelmni/pynoel-gui-app:latest`

```
[benoit@Avishai:~/Boulot/PostDoc/08_SoftWares$ docker run --rm -p 9999:9999 noelmni/pynoel-gui-app:latest
Error: ./kxr_6675.log - No such file or directory.
Warning: ./uploads - No such file or directory.
Dash is running on http://0.0.0.0:9999/
* Serving Flask app "dash_flaskapp" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:9999/ (Press CTRL+C to quit)
```

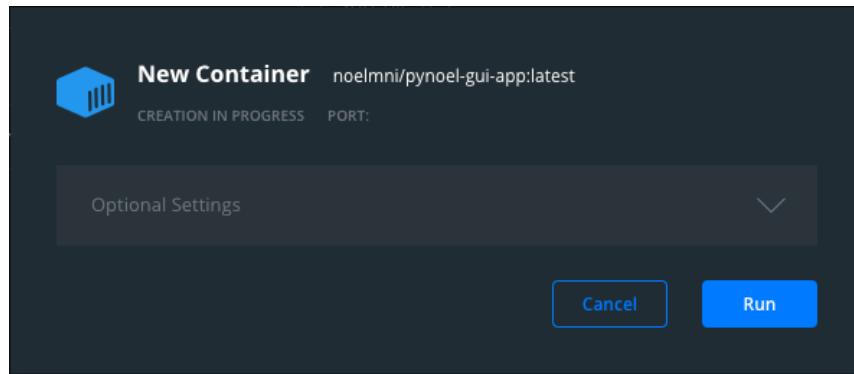
The command executes a container running the noelPyTextures program in background.

1.4.2. Start noelPyTextures from Docker Desktop

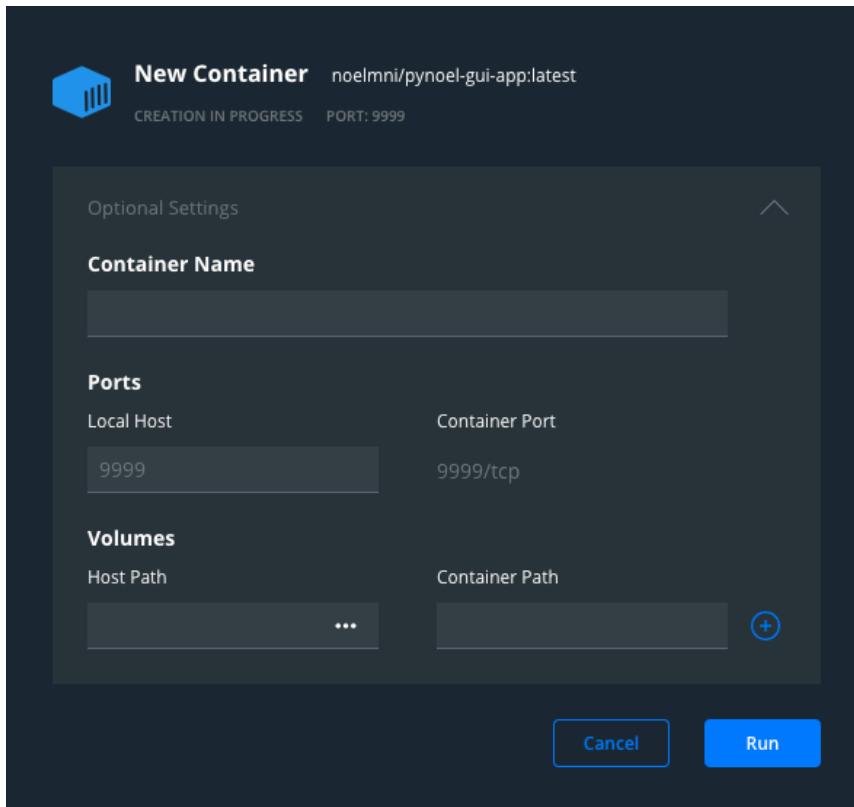
To run noelPyTextures from Docker Desktop, go to “Images” tab and click “RUN” on the line corresponding to the application (the “RUN” button will appear when the mouse cursor hovers over the line).



On the new window, access the drop-down “Optional Settings”.



Once additional options are revealed, in the “Ports” section, fill the ‘Local Host’ field with 9999, and hit “Run”.

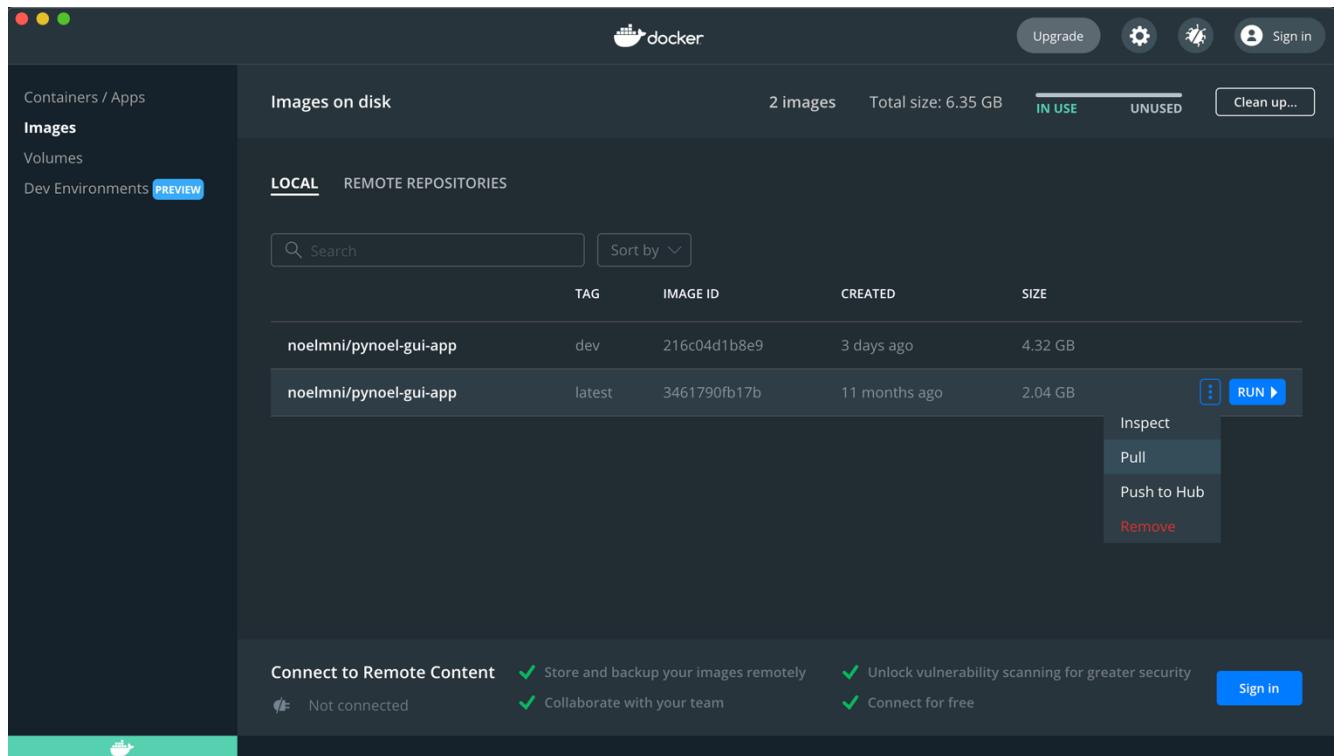


1.4.3 Update of noelPyTextures

From time to time, we might inform you about upgrades to the processing pipeline (which is a different process than the regular updates of Docker Desktop itself).

You will need to update noelPyTextures docker image as follows:

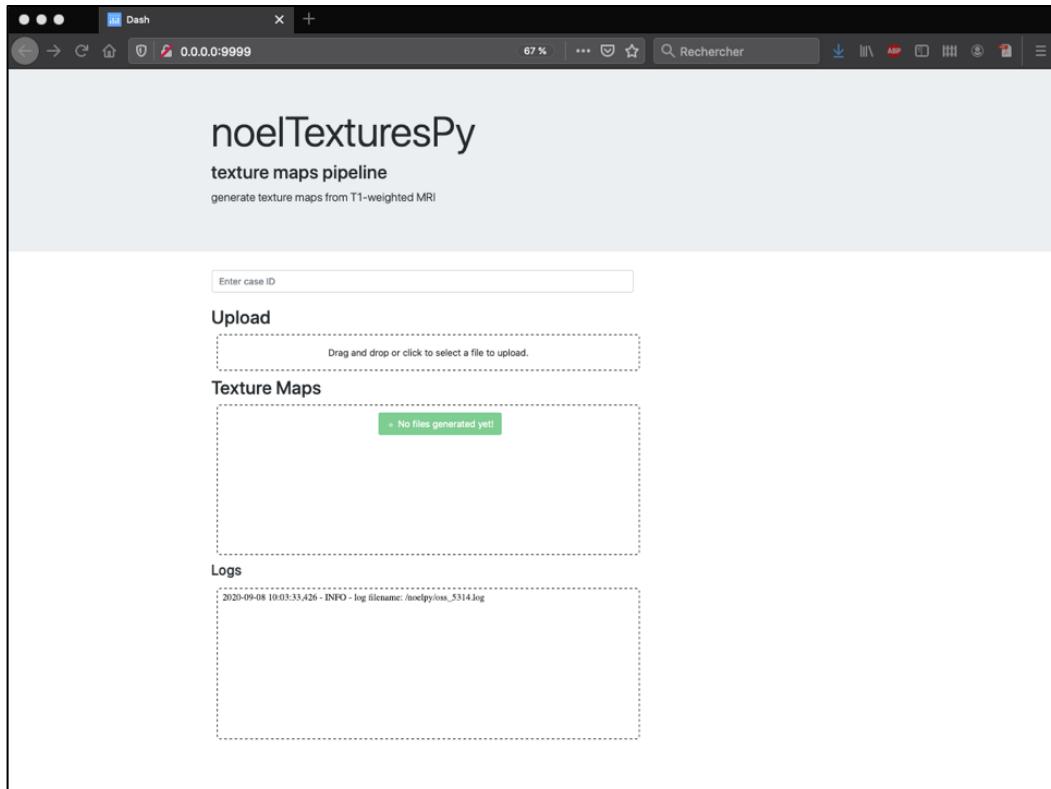
In the “Images” tab in Docker Desktop, click on the three dots next to the “RUN” button (a drop-down menu will appear). Click on “Pull”, and the software will be updated in background.



1.4.4 Access and use of noelPyTextures

This application can be accessed by opening this link in your web-browser:

<http://localhost:9999>



WARNING. To access the website, the container running the program has to be ACTIVE. You can verify the activity by checking the tab “Containers/Apps” in Docker Desktop.

For each case, provide the 3D T1-weighted alone or together with a 3D T2-weighted (or FLAIR).

MR images must be in the Nifti (.nii or .nii.gz) format.

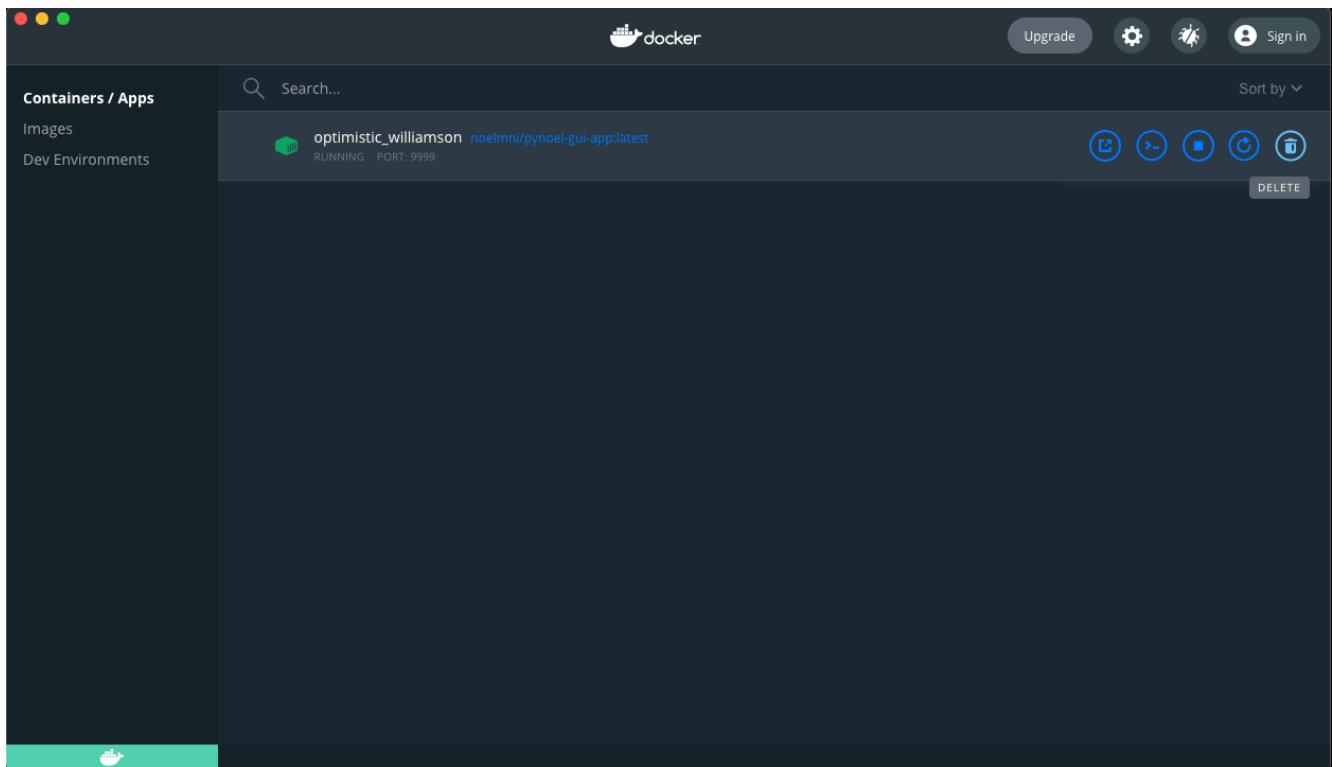
Prior to uploading the files, ensure the naming is correct. The T1-weighted files must include “t1” or “T1” in their filename, and the T2-weighted (or FLAIR) MRI must include either “t2”, “T2”, “flair”, or “FLAIR”. Please ensure all files are renamed before uploading.

Example: john_smith_T1.nii and john_smith_FLAIR.nii

Click the “Upload” section to open a file manager. This allows you to select which images you want to upload to NoelTexturesPy. You can also directly drag and drop your images from the operating system’s file manager (Finder on MacOS) into the “Upload” section.

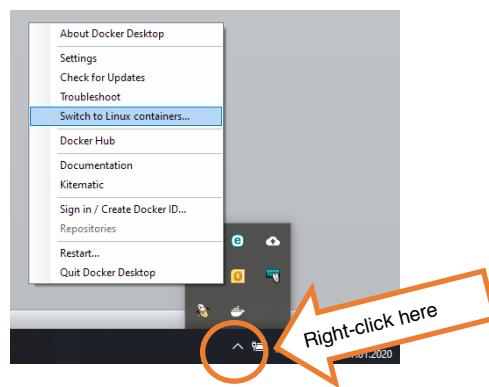
WARNING. If you want to upload two images, make sure you select both of them at once. You can NOT upload a T1-weighted image first and then the FLAIR images.

To stop the processing pipeline, press Ctrl+C in the terminal at any time, or click on the ‘Delete’ button corresponding to the container on the tab “Containers/App” in Docker Desktop. ‘Delete’ option only deletes the container running the program, not the images to be processed.



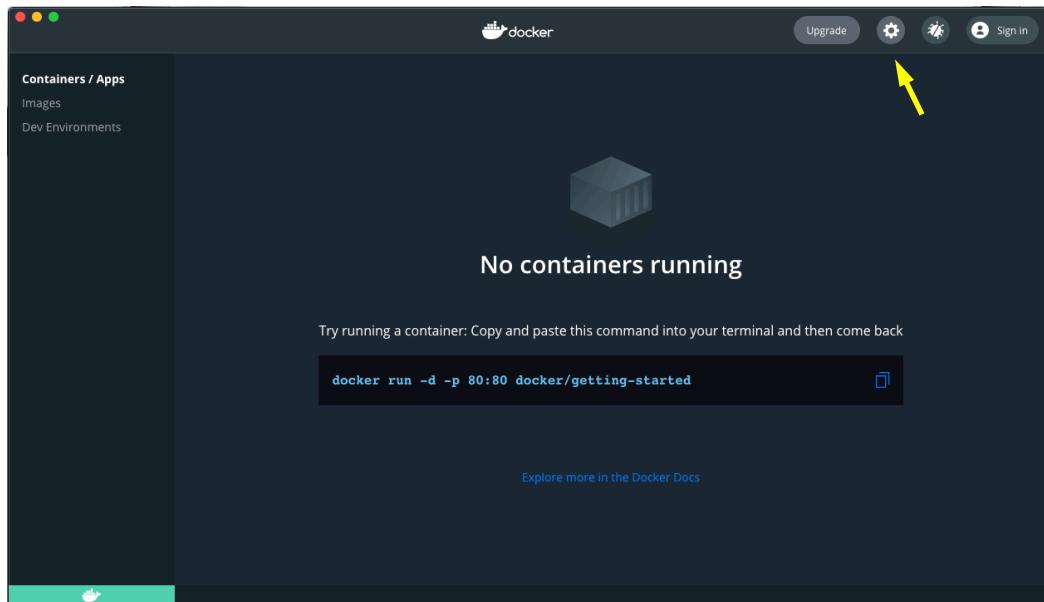
Windows Users

If an error appears during download, “Switch to Linux containers”. Access this option by right-clicking on the docker icon in the lower right corner of your screen.

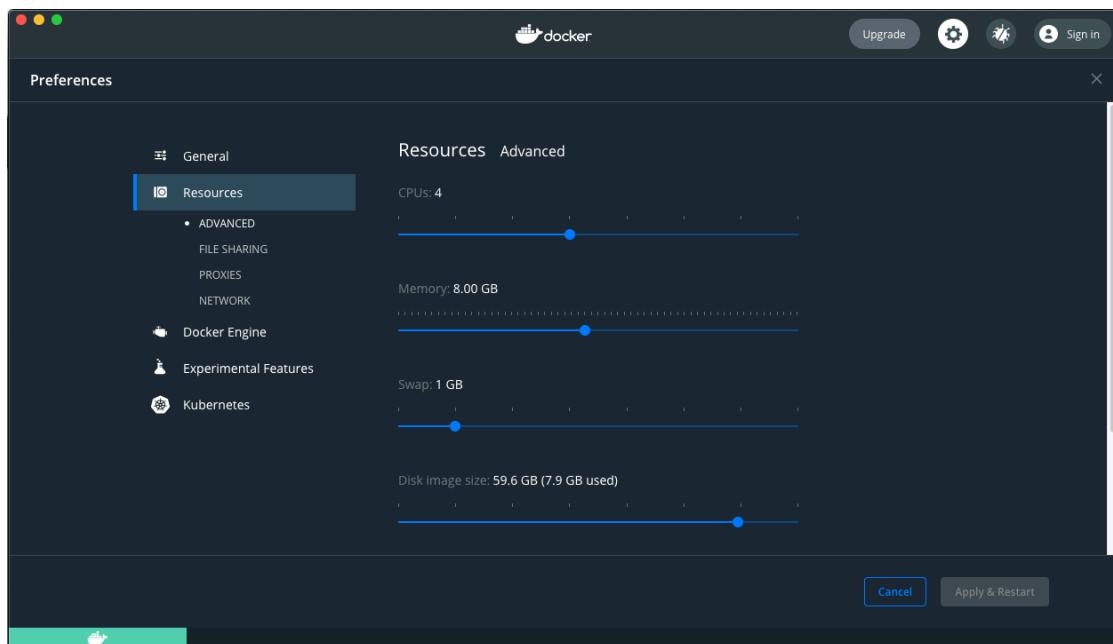


1.4.5 Processing high resolution images, beyond 1mm

If you are using images with a high resolution (e.g., 0.5 mm³ isotropic), the container might stop processing. This is normal, as Docker limits resources used by the container. Limits can be overridden in Docker Desktop, click on the “Settings” button (yellow arrow below).



Then, click on the “Resources” tab.



You can adjust the amount of Memory and Swap available for the containers. 2GB of memory should be enough for millimetric images. For higher resolution, you might want to increase this value to 6GB-8GB.

1.5. How to convert DICOM to NII format

Usually, MRI data are provided in DICOM format.

As NoelTexturesPy accepts only NifTi (.nii or .nii.gz) format, you need to convert the images.

Here are two third-party tools that can be easily installed on all major operating systems:

- If you feel comfortable with command line interface, you can download and install **dcm2niix**: <https://github.com/rordenlab/dcm2niix/releases>. This software is actively maintained.
- If you are more comfortable with a graphical interface, you can download and install **MRIcroGL**: https://www.nitrc.org/frs/?group_id=889. It includes a conversion tool, based on dcm2niix.

2. PROCESSING AND QUALITY CHECK (QC)

2.1 Processing procedure

The processing steps include:

- Registration of the images to MNI152 template
- N4 intensity non-uniformity correction
- Brain extraction
- Tissue segmentation (CSF, white matter, gray matter)
- Computation of the *gradient magnitude* map (modeling FCD blurring)
- Computation of the *relative intensity* map (modeling FCD hyperintensities)
- Generation of a PDF report, which allows you to quickly inspect the output of each processing step (see details pages 15-19)

The processing files are generated in the background. The “Logs” section (see below) lists the processing steps.

Once the procedure is finished, links to the results become available in the “Texture Maps” section (see below). You will be able to download results from your browser and visualize them on your computer using a dedicated MRI visualization tool of your choice (*e.g.*, register, ITK-SNAP, fslview, freeview).

In total, 5 files are generated (with a prefix you decide, for example the name or numeric ID):

- **Prefix_t1_final.nii.gz**: T1-weighted MRI registered to the MNI152 template
- **Prefix_flair_final.nii.gz**: FLAIR MRI registered to the MNI152 template
- **Prefix_t1_gradient_magnitude.nii.gz**: gradient magnitude derived from the t1w image
- **Prefix_t1_relative_intensity.nii.gz**: relative intensity derived from the t1w image
- **Prefix_QC_report.pdf**: PDF report to check the accuracy of the different processing steps

Typically, the pipeline takes about **10-15 minutes** (depending on the CPU core count) to run, and files will appear once all the relevant outputs have been generated.

noelTexturesPy

texture maps pipeline

generate texture maps from T1-weighted MRI

Enter case ID

Upload

Drag and drop or click to select a file to upload.

Texture Maps

[vop_7028_QC_report.pdf](#)

[vop_7028_t1_final.nii.gz](#)

[vop_7028_t1_gradient_magnitude.nii.gz](#)

[vop_7028_t1_relative_intensity.nii.gz](#)

[vop_7028_t2_final.nii.gz](#)

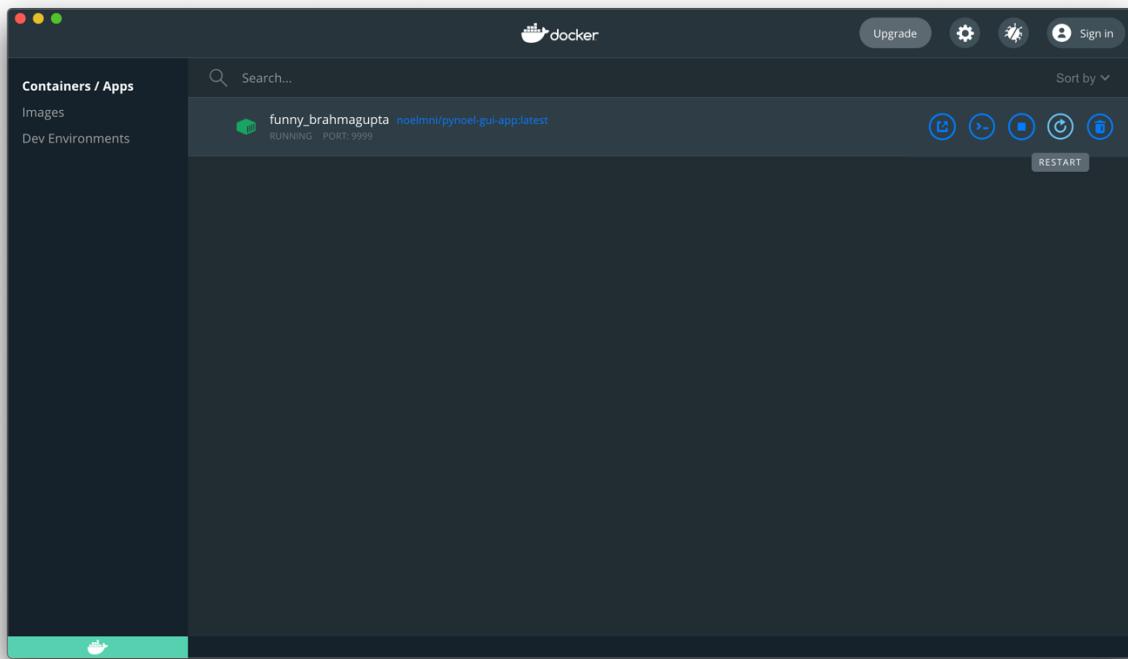
Logs

```
2020-09-08 09:43:47,883 - INFO - T1-weighted image detected: mcd_129_1_t1_nii.nii
2020-09-08 09:43:47,883 - INFO - T2-weighted image detected: mcd_129_1_flair_nii.nii
2020-09-08 09:43:47,883 - INFO - assigning randomly generated case ID
2020-09-08 09:43:47,883 - INFO - case ID: vop_7028
2020-09-08 09:43:47,883 - INFO - loading nifti files
2020-09-08 09:43:48,428 - INFO - registration to MNI template space
2020-09-08 09:44:26,018 - INFO - performing N4 bias correction
2020-09-08 09:45:24,622 - INFO - performing brain extraction
2020-09-08 09:45:42,368 - INFO - computing GM, WM, CSF segmentation
2020-09-08 09:46:47,104 - INFO - computing gradient magnitude
2020-09-08 09:46:49,988 - INFO - computing relative intensity
2020-09-08 09:46:52,980 - INFO - generating QC report
2020-09-08 09:48:49,710 - INFO - pipeline processing time elapsed: 301.8 seconds
2020-09-08 09:48:49,710 - INFO - ****
```

WARNING. When the processing is finished your case, don't forget to download results. If you hit Ctrl-C in the terminal, or the "Restart" or "Delete" buttons, the running container will be deleted, along with all its content including the processed MRI outputs and the PDF report.

To process another case, shut the container down (Ctrl-C in the terminal) and restart noelTexturesPy using the command: docker run --rm -p 9999:9999 noelmni/pynoel-gui-app:latest

Alternatively, hit the restart button corresponding to the container in the Docker Desktop window, as displayed in the following illustration.



In any case, do not forget to refresh NoelTexturesPy webpage.

2.2. How to verify the quality of the processing

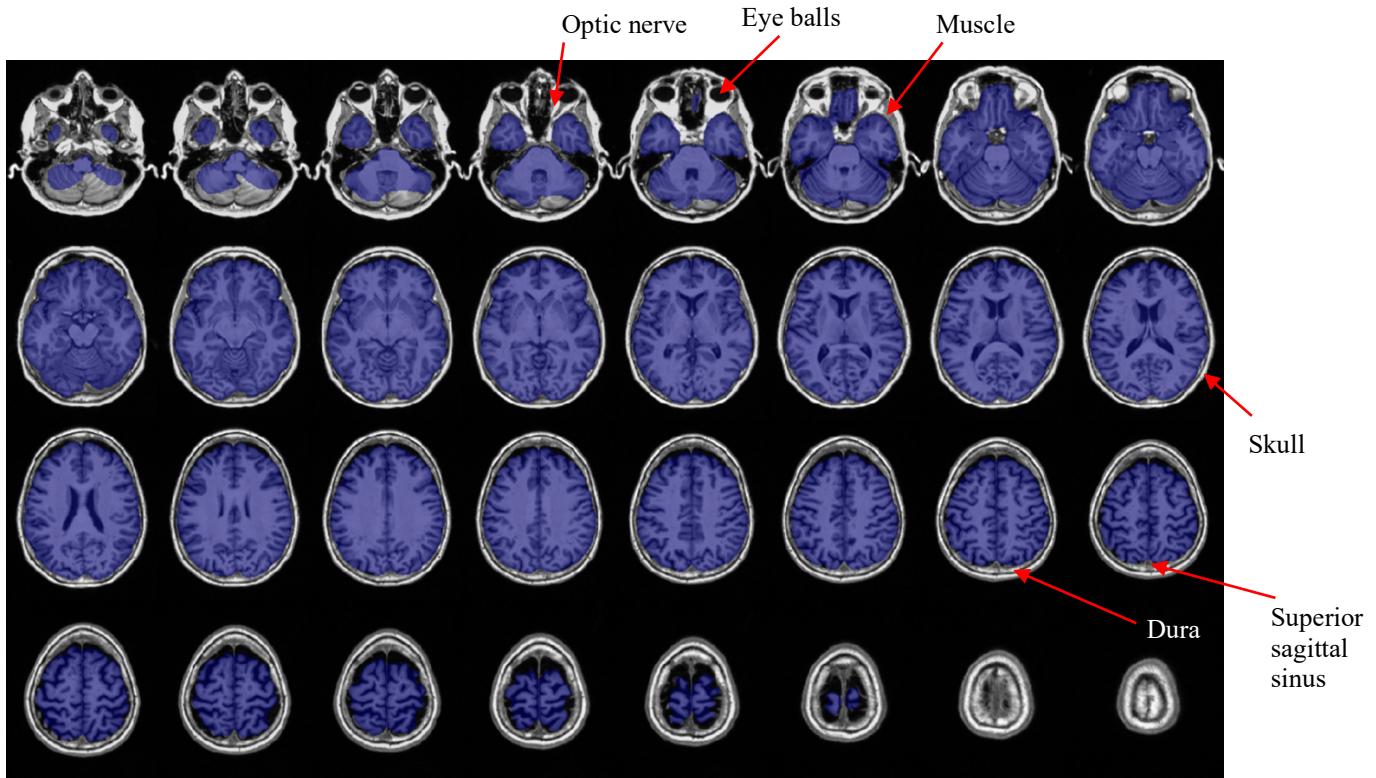
The file <Case-ID-Prefix>_QC_report.pdf includes 2 pages that allow assessing whether the skull stripping and tissue segmentation (separation of the brain into CSF, GM, and WM) are accurate.

The purpose of **skull stripping** is to mask out non-brain tissue (skull, dura, optic nerve, eye balls, muscles; Figure A, page 16) in order to identify only brain tissue. However, errors may occur (Figure B and C).

Aside from inaccuracies in skull stripping, another reason for poor brain tissue segmentation (such as WM labeled as GM, and vice-versa) may be due to local intensity variations that have not been sufficiently reduced by the correction of field inhomogeneity. Generally, in this scenario, errors are negligible and do not modify results.

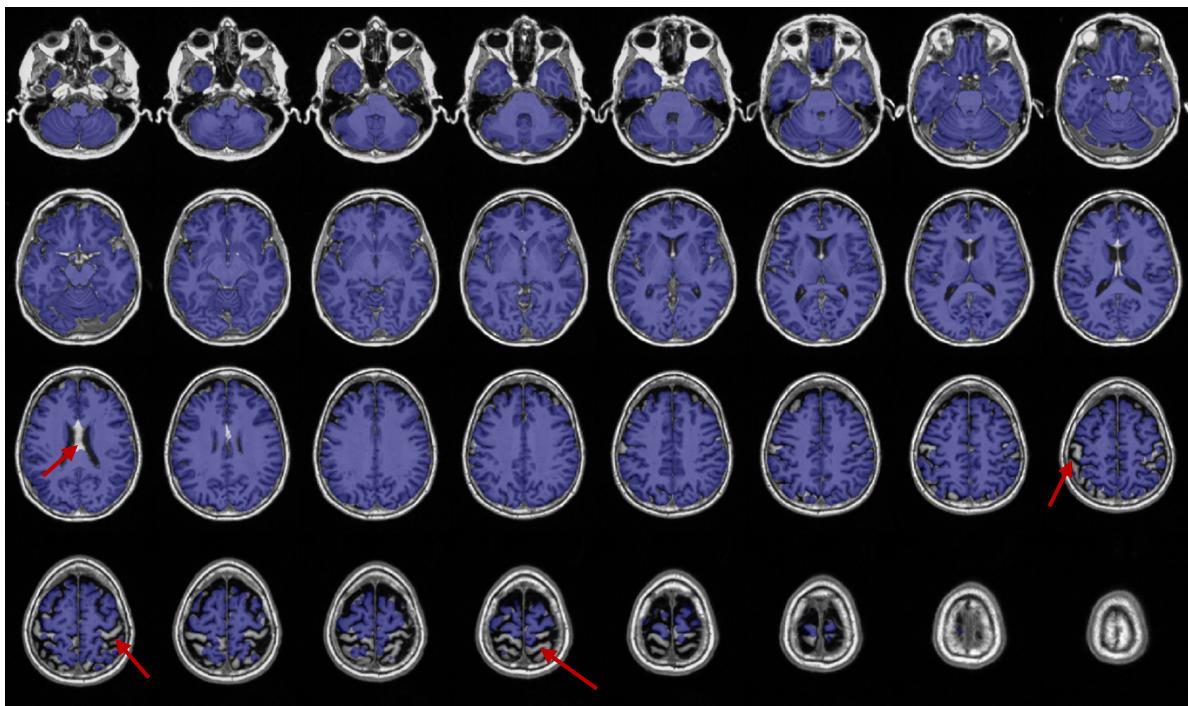
A) Successful skull stripping

The purple mask only includes GM, WM and CSF.



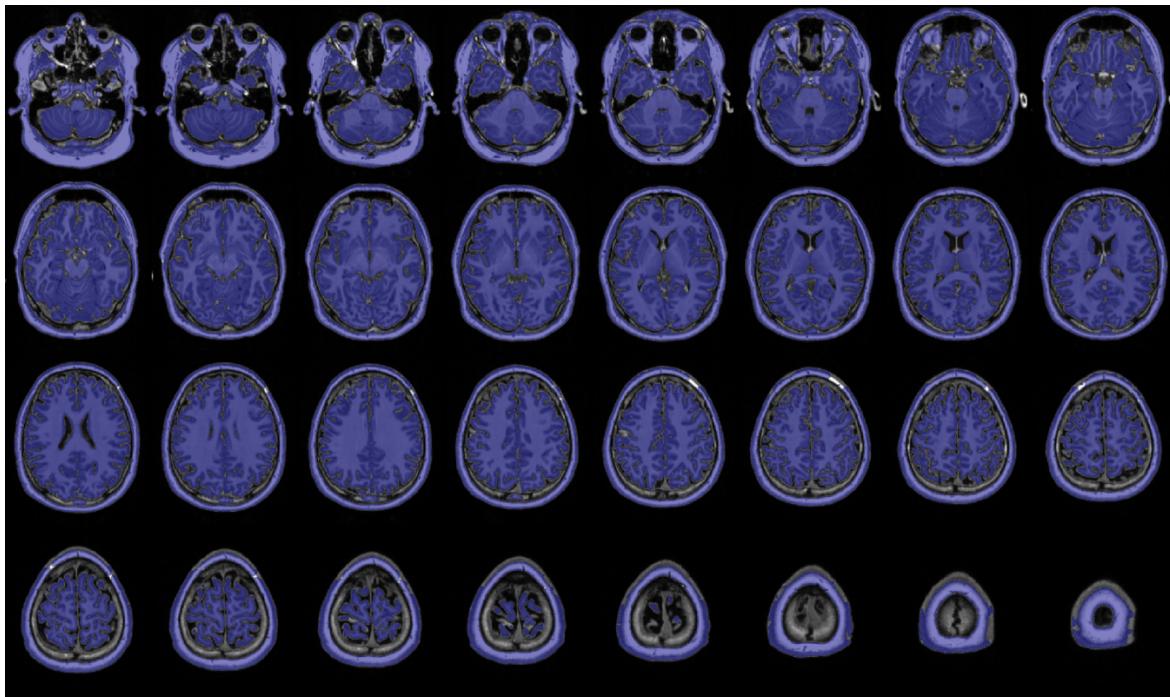
B) Inaccurate skull stripping

The purple mask misses part of the brain due to over-estimation of the skull stripping



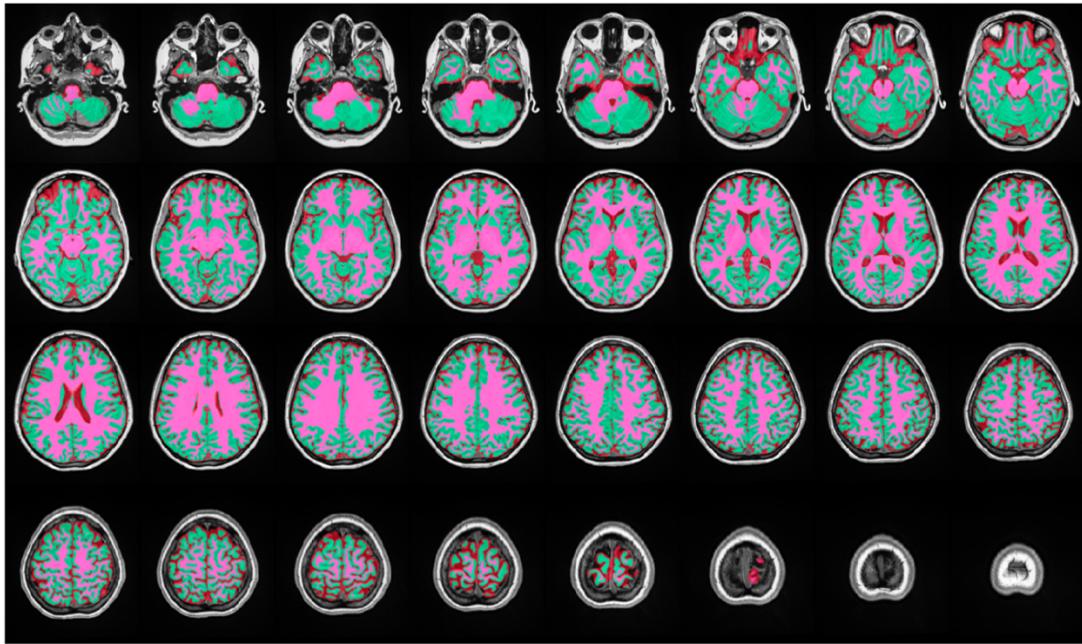
C) Inaccurate skull stripping

Non-brain tissue is not sufficiently excluded (only portions of the dura are included in the stripping process).

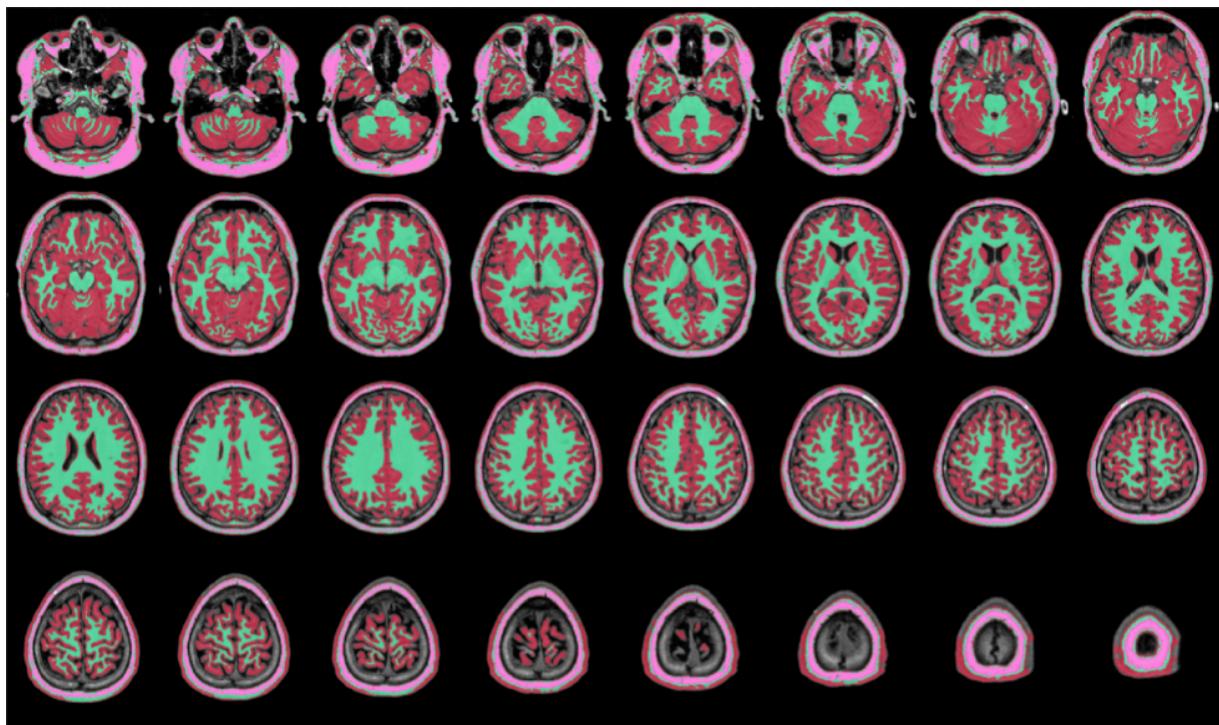


The skull stripping influences the quality of tissue segmentation into CSF, WM and GM.

Accurate segmentation. WM (pink), GM (green) and CSF (red).



Erroneous segmentation. The skull stripping did not exclude the skull, which was segmented as WM. The GM is erroneously classified as CSF. This type of tissue miss-classification may lead to inaccurate results in the texture maps!

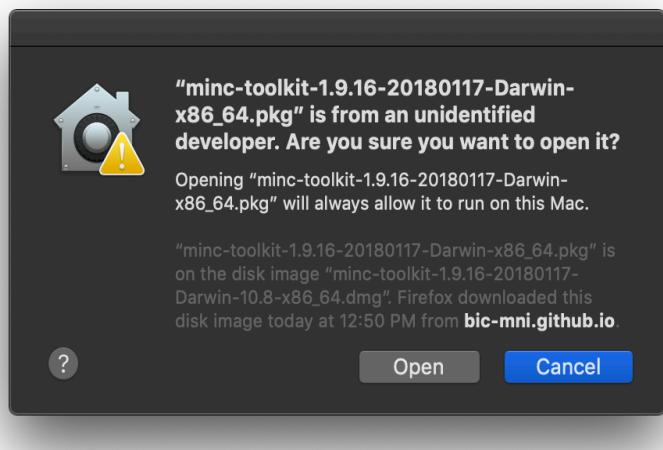


3. INSTALLATION/USE OF REGISTER + MINC-TOOLKIT V2 [FOR MACOS ONLY]

Download register [as part of a series of neuroimaging tools] here:
http://packages.bic.mni.mcgill.ca/minc-toolkit/MacOSX/minc-toolkit-1.9.18-20200825-Darwin-10.9-x86_64.dmg

Double-click on the downloaded file to run the installer.

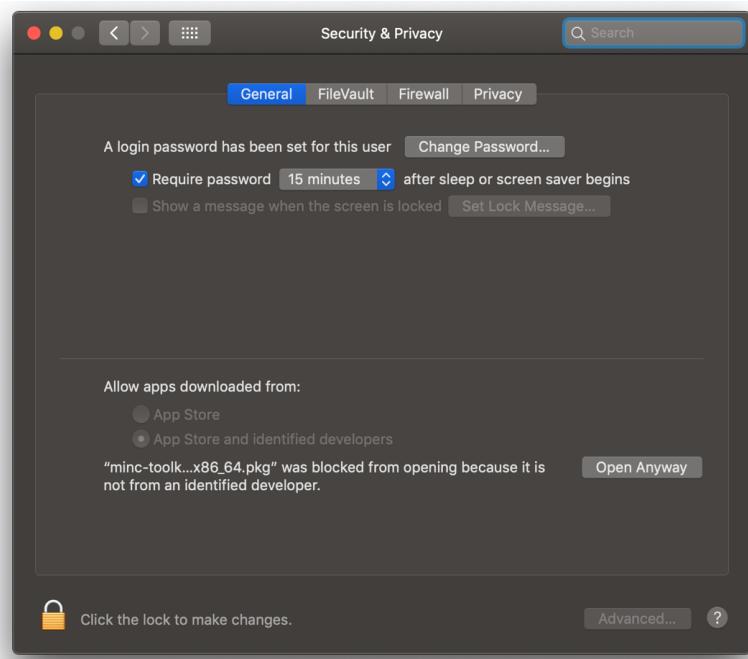
Refer the following screenshots to confirm the program installed successfully. The software versions may differ slightly between the text and screenshots without affecting any of the subsequent steps.

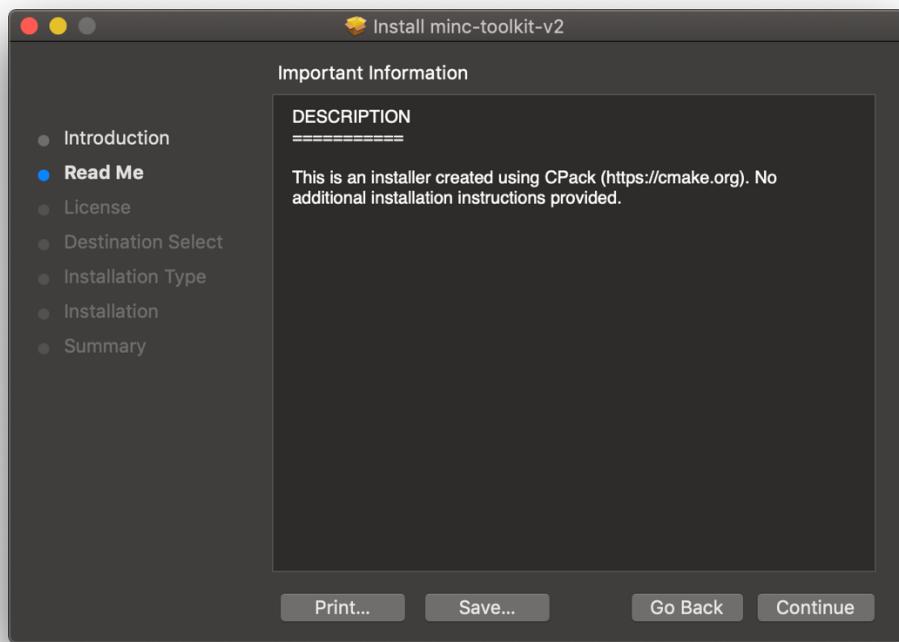
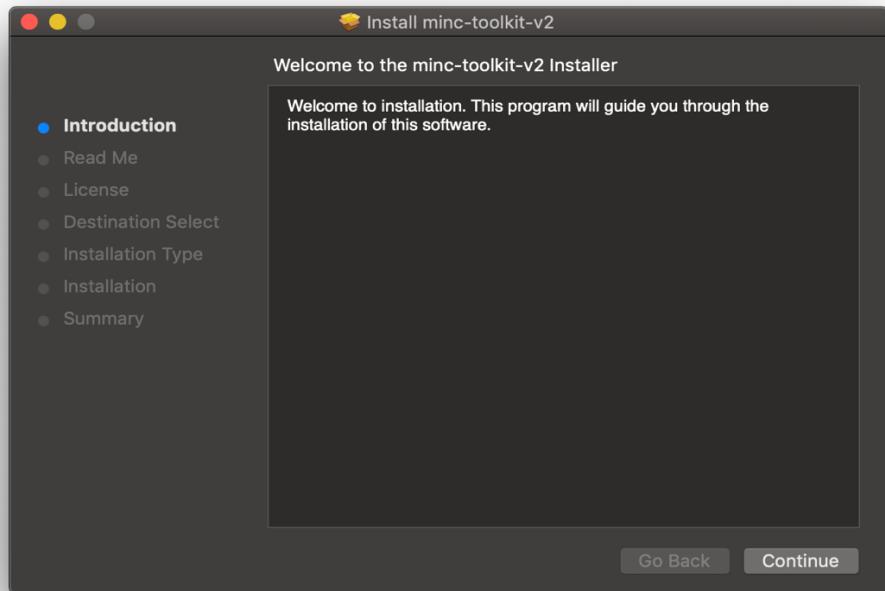


Click "Open"

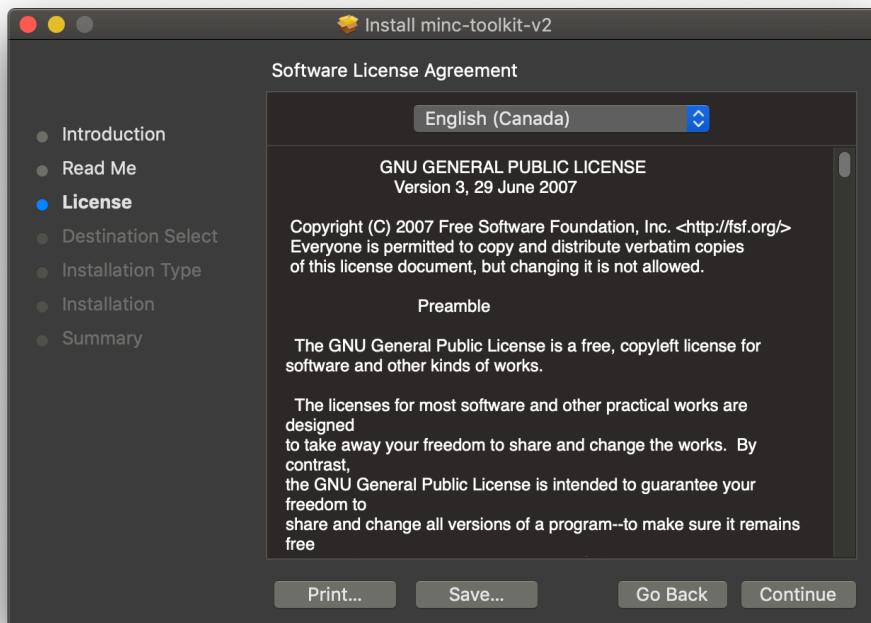
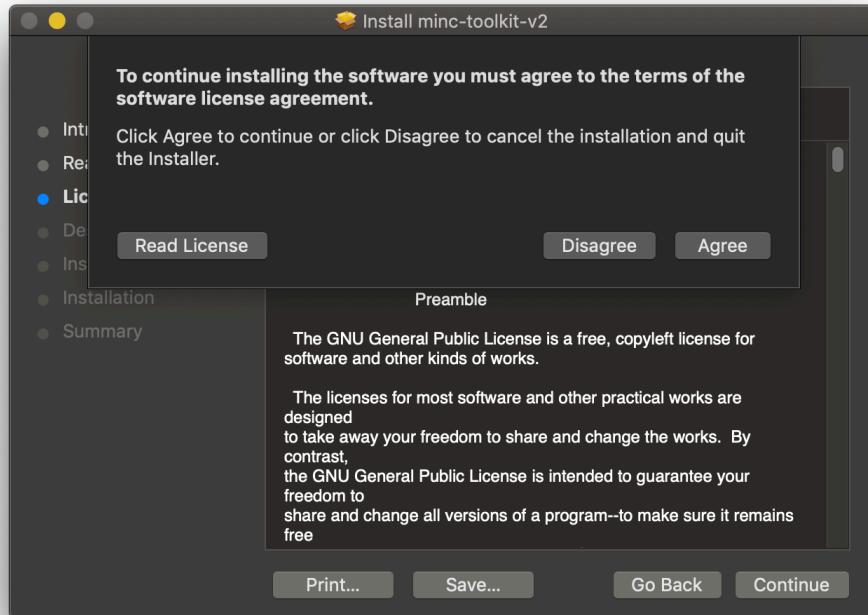


If you see the above prompt, open System Preferences > Security & Privacy > General, and click “Open Anyway”, as shown below.

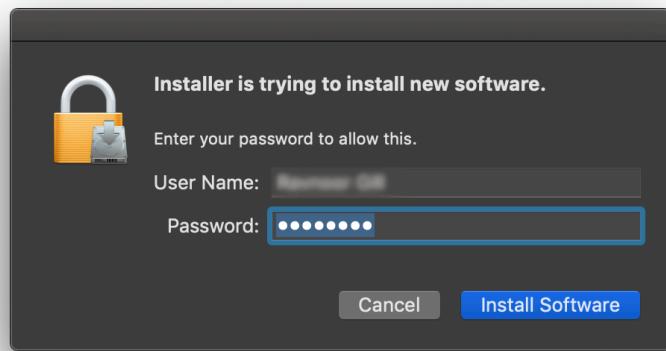
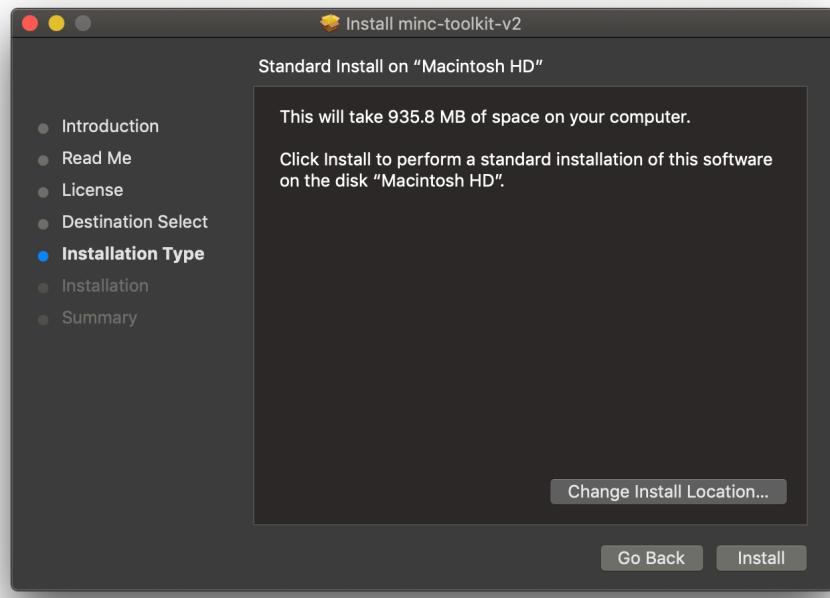




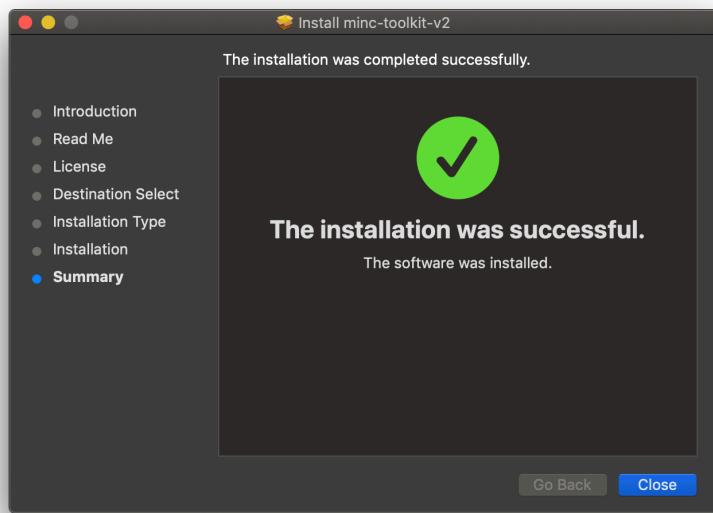
Press “Continue”



Press “Continue”



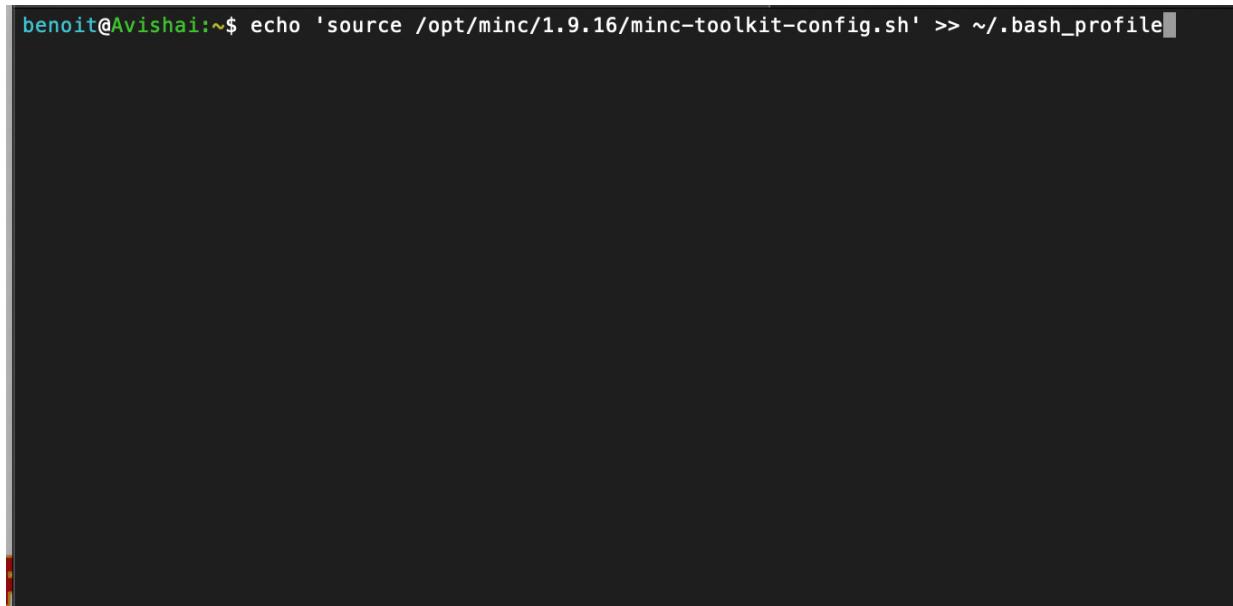
Click "Install", enter your MacOS username and password, and then press 'Install Software'



You have now successfully installed the BIC-MNI minc-toolkit-v2.

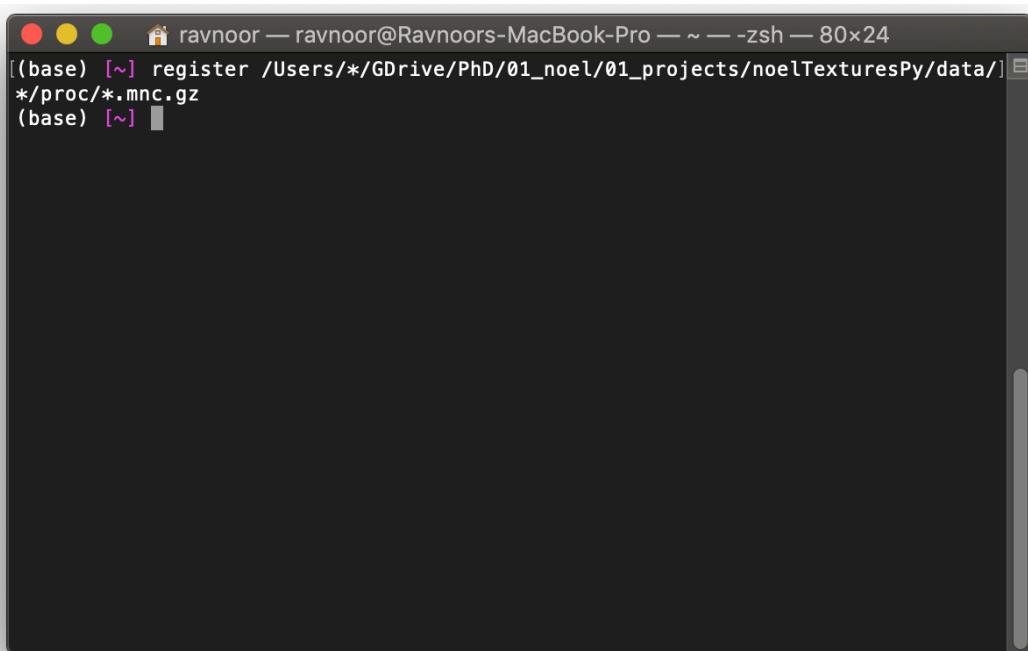
Open the ‘Terminal’ application.

Type: echo ‘source /opt/minc/1.9.16/minc-toolkit-config.sh’ >> ~/.bash_profile

A screenshot of a terminal window with a black background and white text. The command 'benoit@Avishai:~\$ echo 'source /opt/minc/1.9.16/minc-toolkit-config.sh' >> ~/.bash_profile' is typed at the prompt. The cursor is visible at the end of the command line.

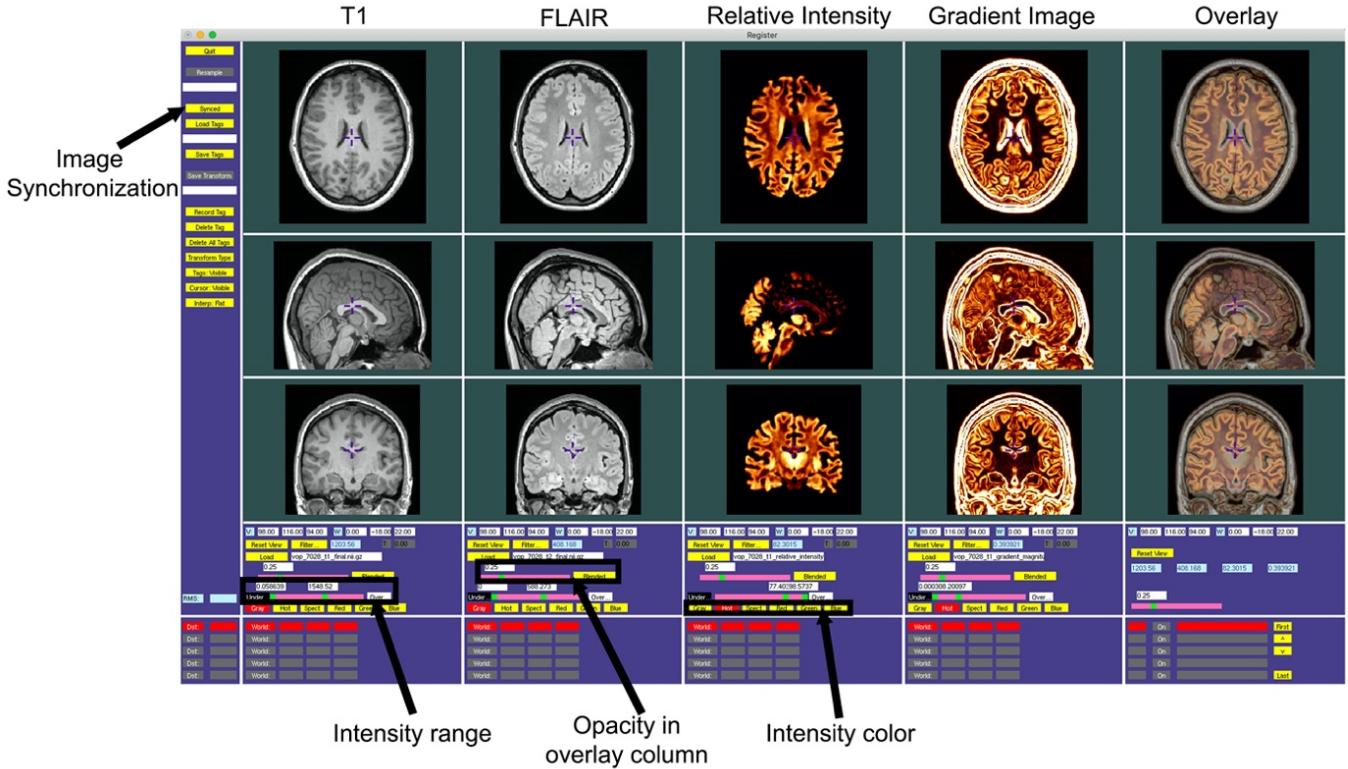
```
benoit@Avishai:~$ echo 'source /opt/minc/1.9.16/minc-toolkit-config.sh' >> ~/.bash_profile
```

You can access register simply by typing register in the Terminal app.

A screenshot of a terminal window with a dark gray background and light gray text. The command 'register' is typed at the prompt. The window has a title bar with the user name 'ravnoor' and the host name 'Ravnoors-MacBook-Pro'. The title bar also shows the path '/Users/*/*GDrive/PhD/01_noel/01_projects/noelTexturesPy/data/' and the file '*/proc/*.mnc.gz'.

```
ravnoor — ravnoor@Ravnoors-MacBook-Pro — ~ — -zsh — 80x24
[(base) [~] register /Users/*/*GDrive/PhD/01_noel/01_projects/noelTexturesPy/data/]
*/proc/*.mnc.gz
(base) [~] ]
```

Navigate to the folder where your images are stored before launching register if you don’t want to type the whole path.



Images are simultaneously viewed in the three orthogonal axes (axial, sagittal, coronal).

From left to right: T1, FLAIR, Relative intensity and intensity (modeling blurring). The last set of images on the right present all images overlaid (you can scroll through them using the green cursor).

Intensity range and intensity colors can be individually adjusted.

Since images are co-registered and in the same anatomical space, you can scroll through them simultaneously by clicking on the "Synced/Not Synced" button in the upper left area. By doing so, the cursor will be in the same voxel across all images and views.

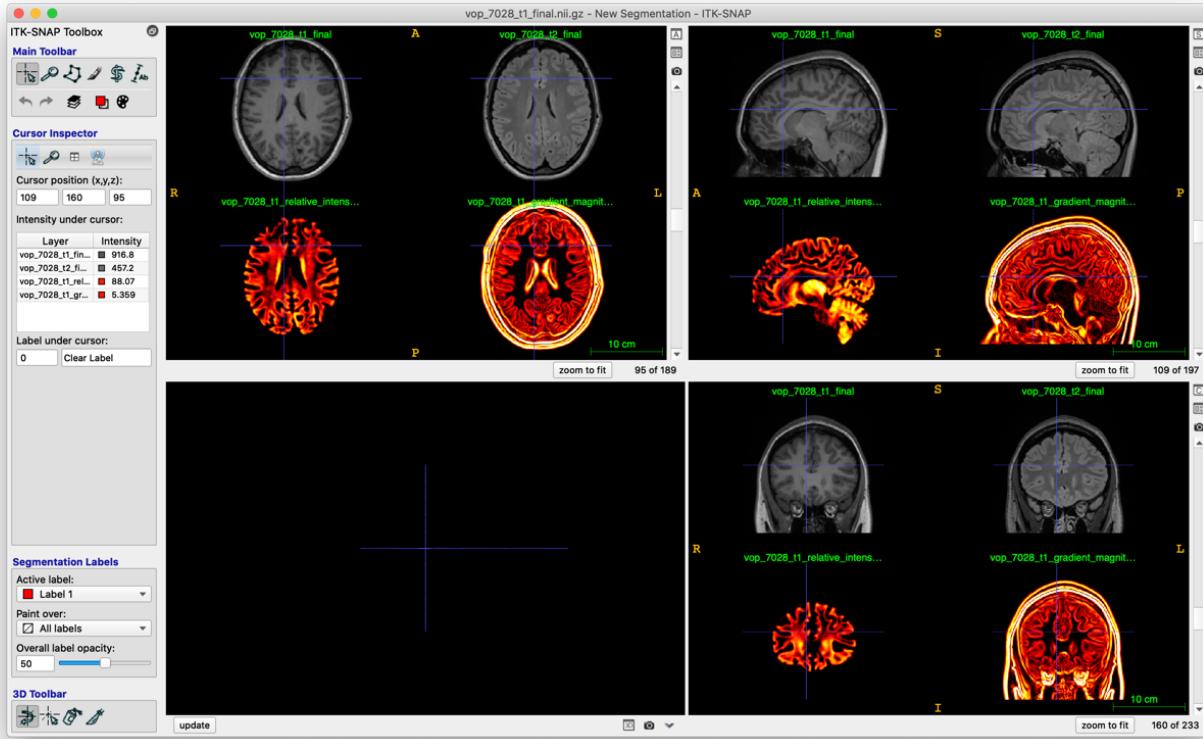
WARNING. When you set your folder and file names, avoid the use of whitespaces. For example, use 'Texture-Analysis' or 'Texture_Analysis' instead of 'Texture Analysis'. It is much easier to navigate the file system in the terminal when there are no spaces in the naming structure. The reason is that you have to introduce backslashes ('\\') or quotes in the command to take these spaces into account.

For example:

```
register '/Users/benoit/Texture Pipeline/* & or register
/Users/benoit/Texture\ Pipeline/* &
instead of
register /Users/benoit/Texture_Pipeline/* &
```

Note the positioning of the quotes in the first version.

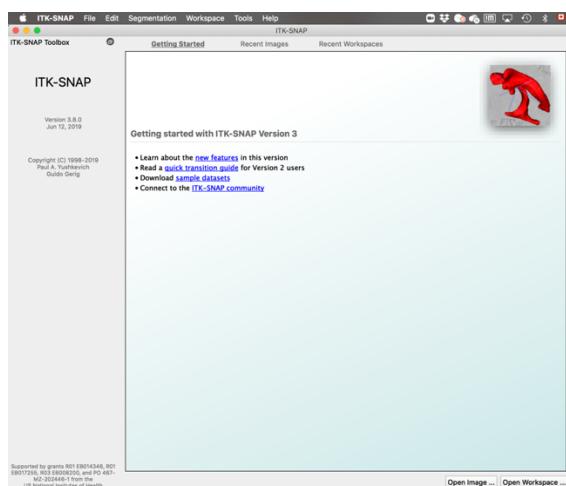
4. INSTALLATION AND USE OF ITK-SNAP [ALL SYSTEMS]



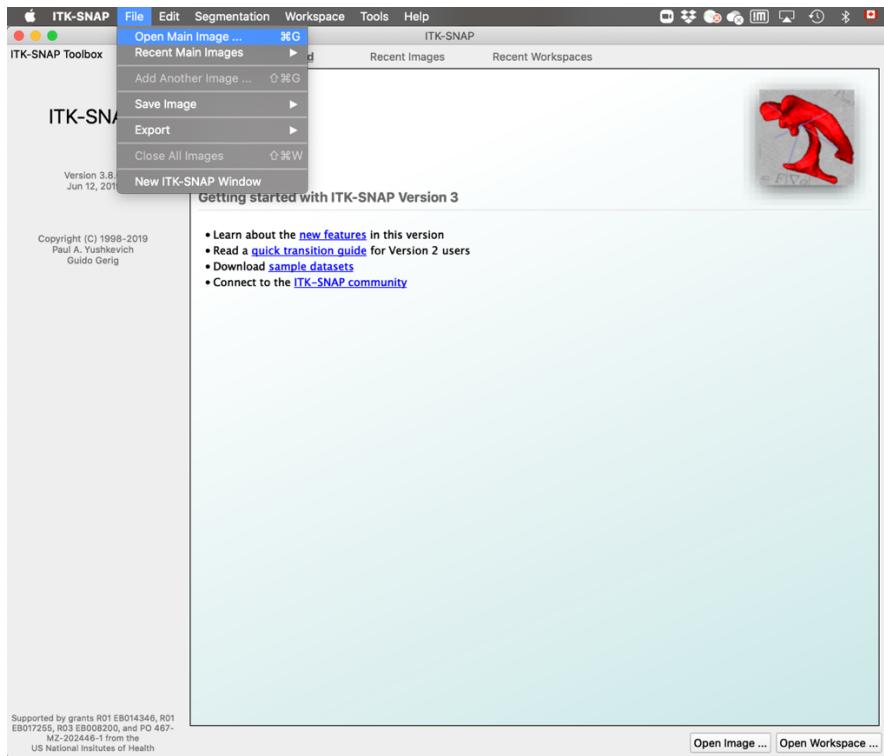
ITK-SNAP is a visualization software available on major platforms (Windows, MacOS and Linux). Download here: <http://www.itksnap.org/pmwiki/pmwiki.php?n=Downloads.SNAP3>
Select the latest version corresponding to your platform.

4.1. Load Images

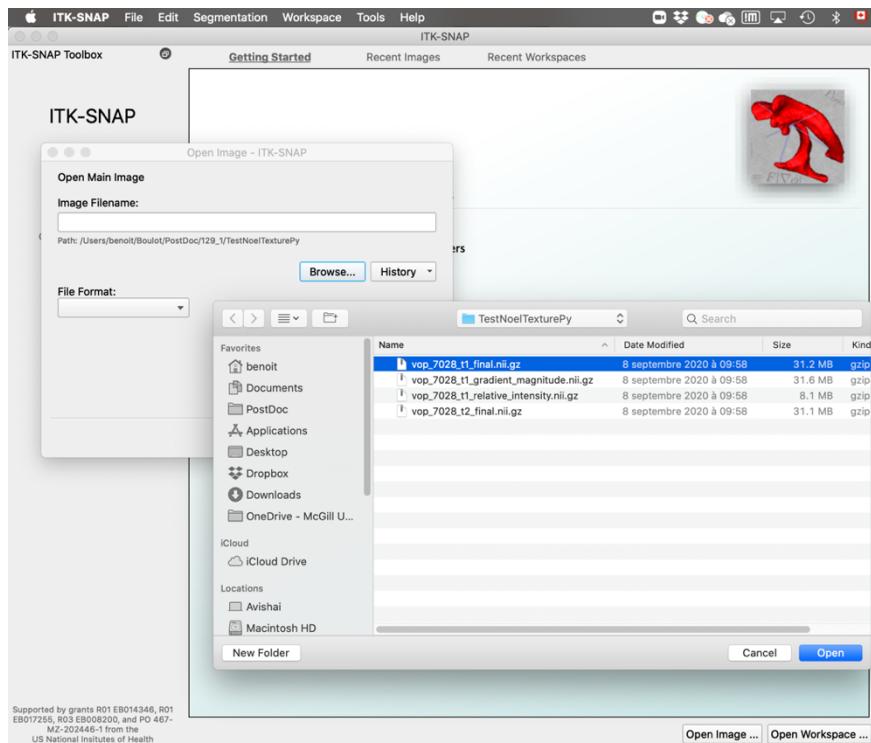
When first opening ITK-SNAP, you will be presented with the following screen.



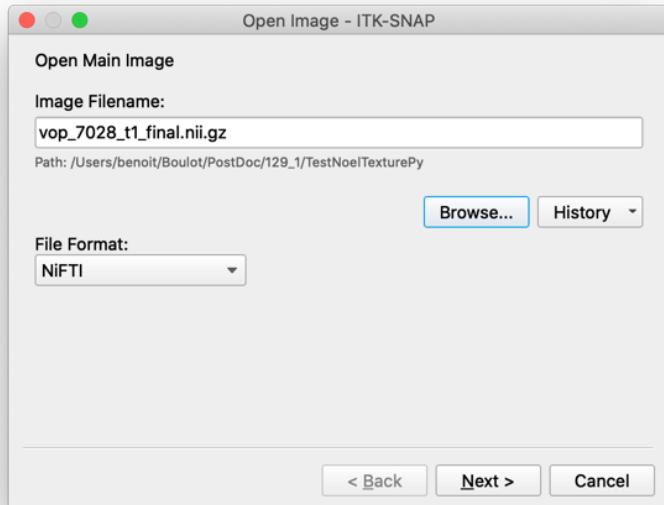
To open the first image, select “File > Open Main Image”.



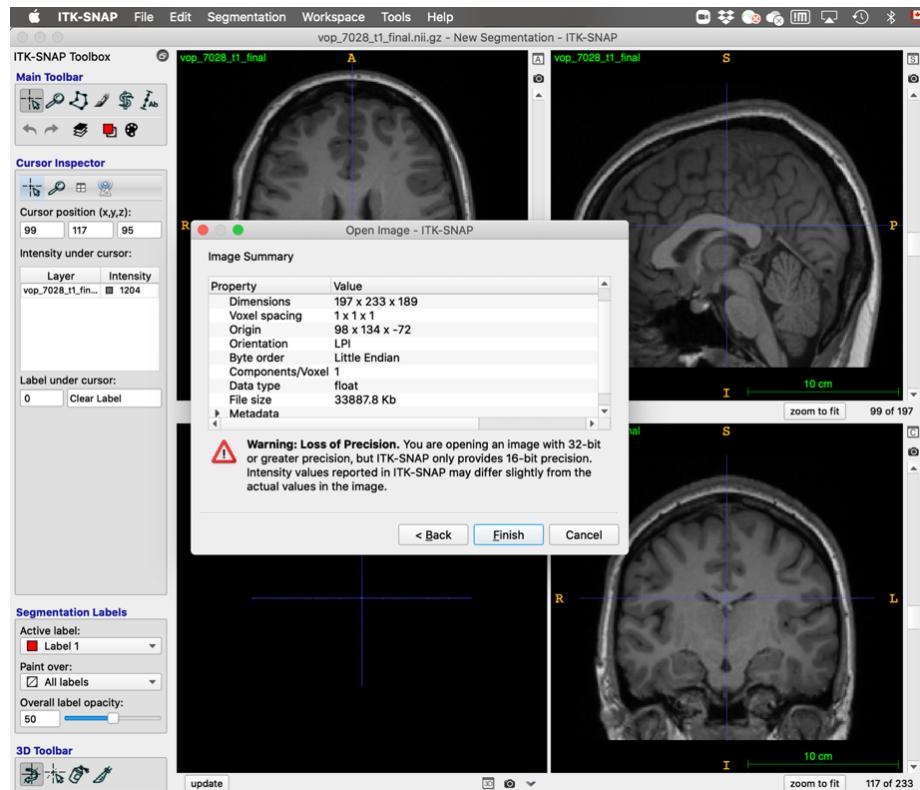
The dialog box “Open Image – ITK-SNAP” will open. Click on “Browse” to search for the file you want to load (load the T1-weighted image at this stage).



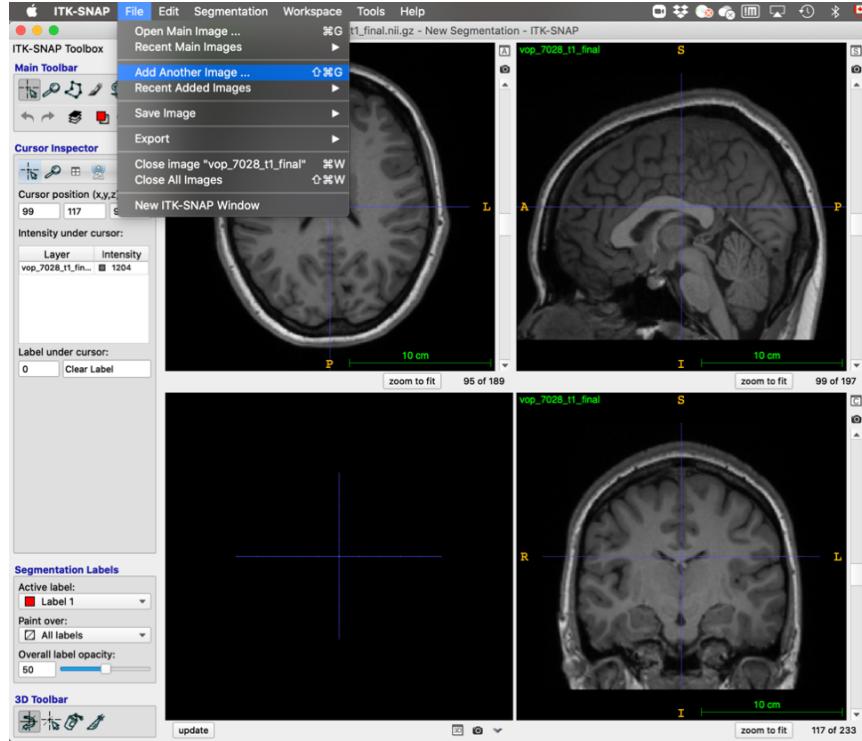
Once you click “Open”, the following dialog box will appear:



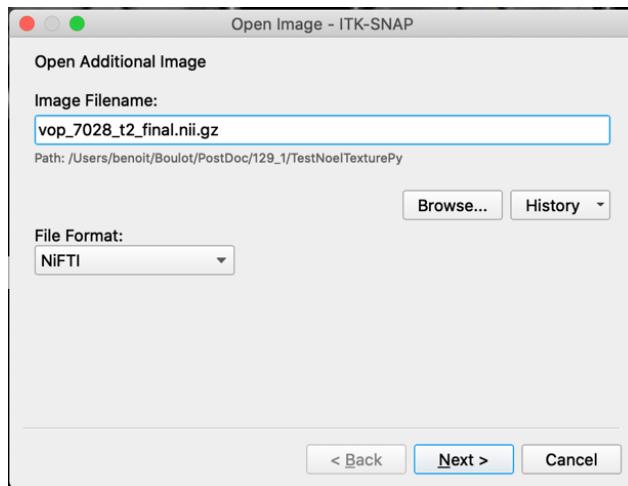
Click “Next”. The image will be loaded, and a final dialog box with the image characteristics will be displayed. Click on “Finish”.



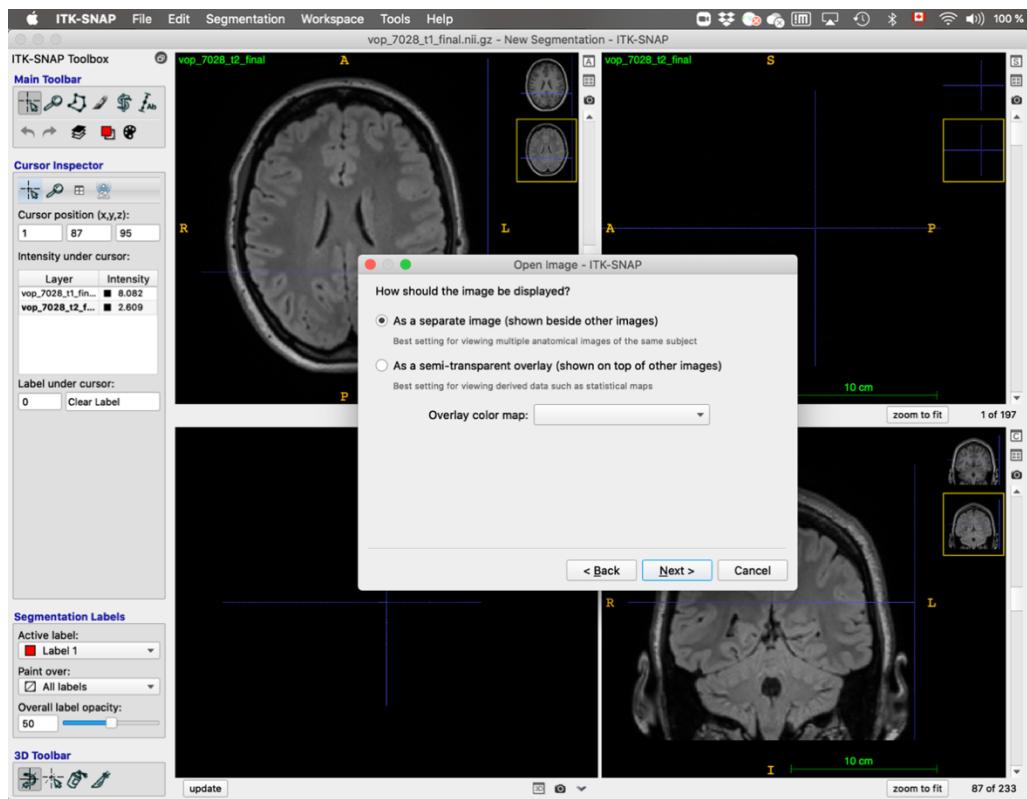
ITK-SNAP allows juxtaposition of multiple images. To add a new image, select “File > Add Another Image ...”.



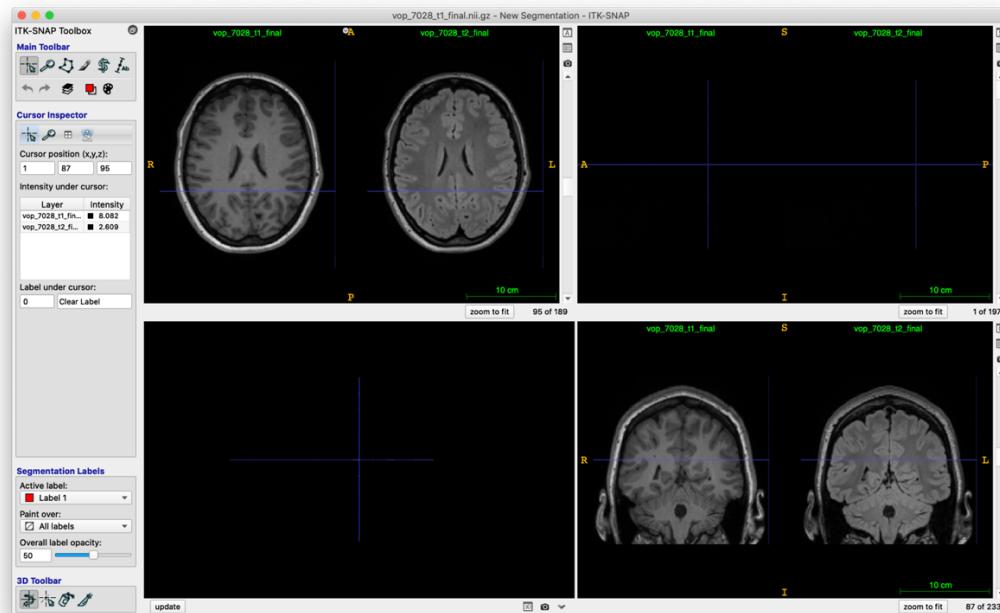
The same dialog box presented earlier will appear. We recommend loading the T2-weighted file. Until the selection of the image, the procedure is exactly the same.



When clicking “Next”, ITK-SNAP will ask to display the additional image as a separate image, or as an overlay. Choose “As a separate image (shown beside other images)”. Click “Next”, and a dialog box summarizing the additional image’s properties will appear. Click on “Finish” to proceed.



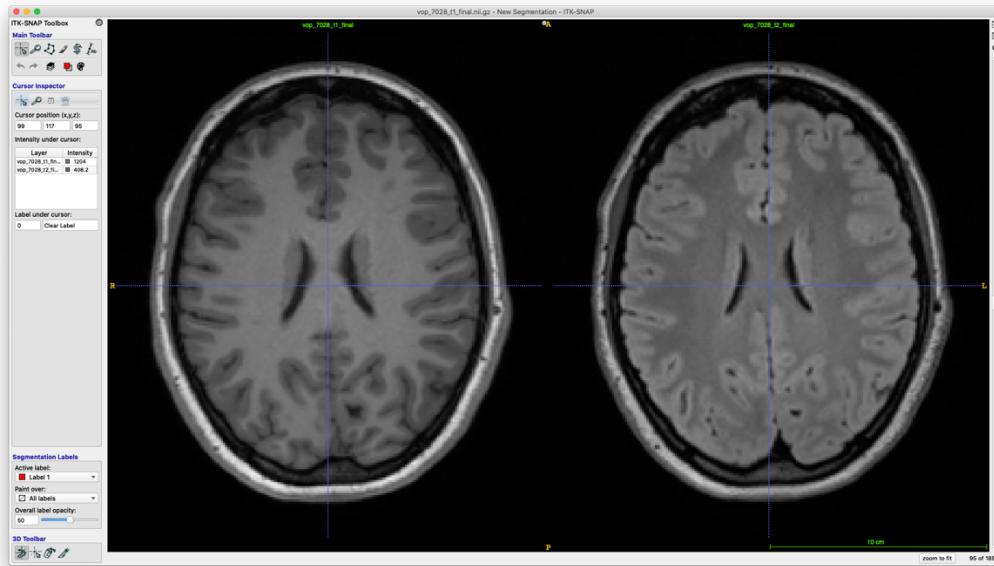
Notice that both images are **not** juxtaposed next to one another.
To enable this display, click the button highlighted in red as follows.



To explore one orientation more closely, you can extend its view to full screen by clicking one of the highlighted red buttons.

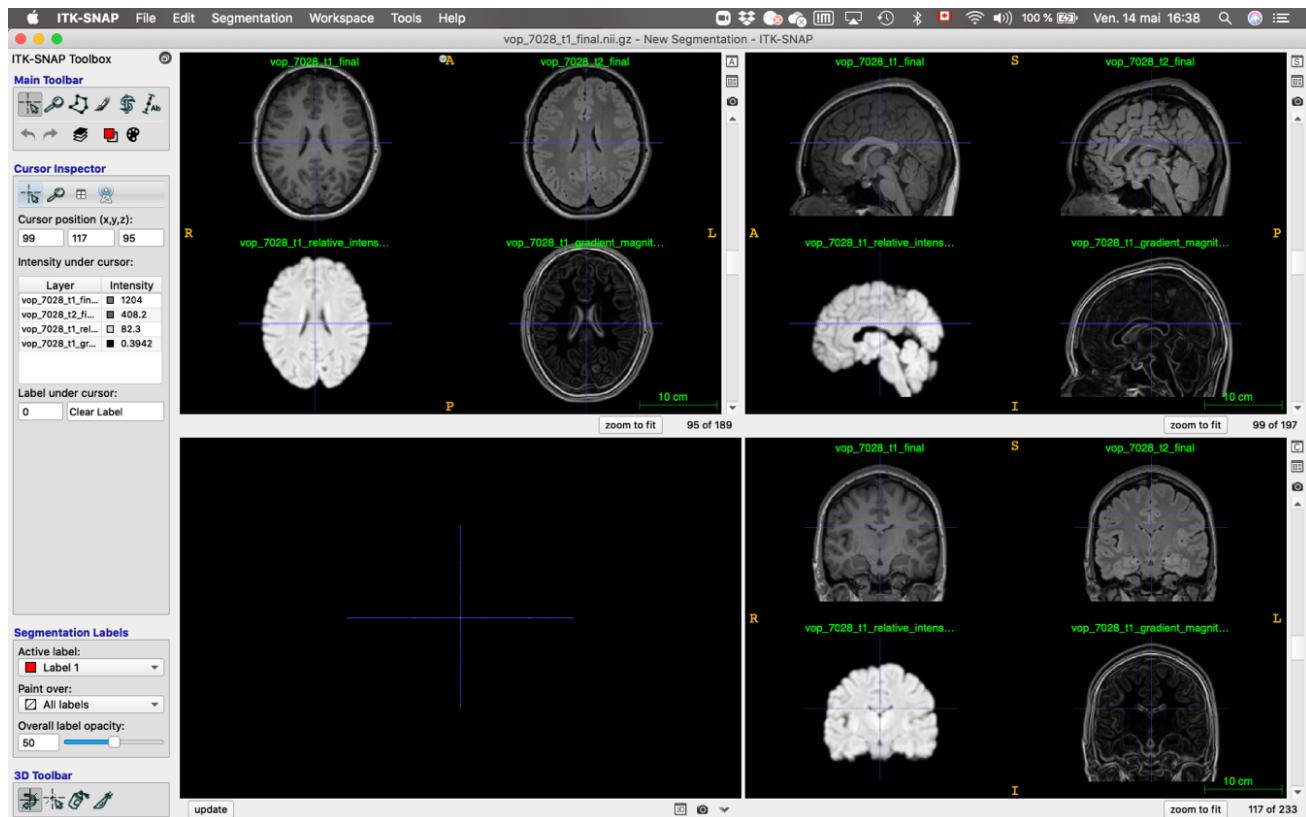


This will result in the following display (when axial view is selected).



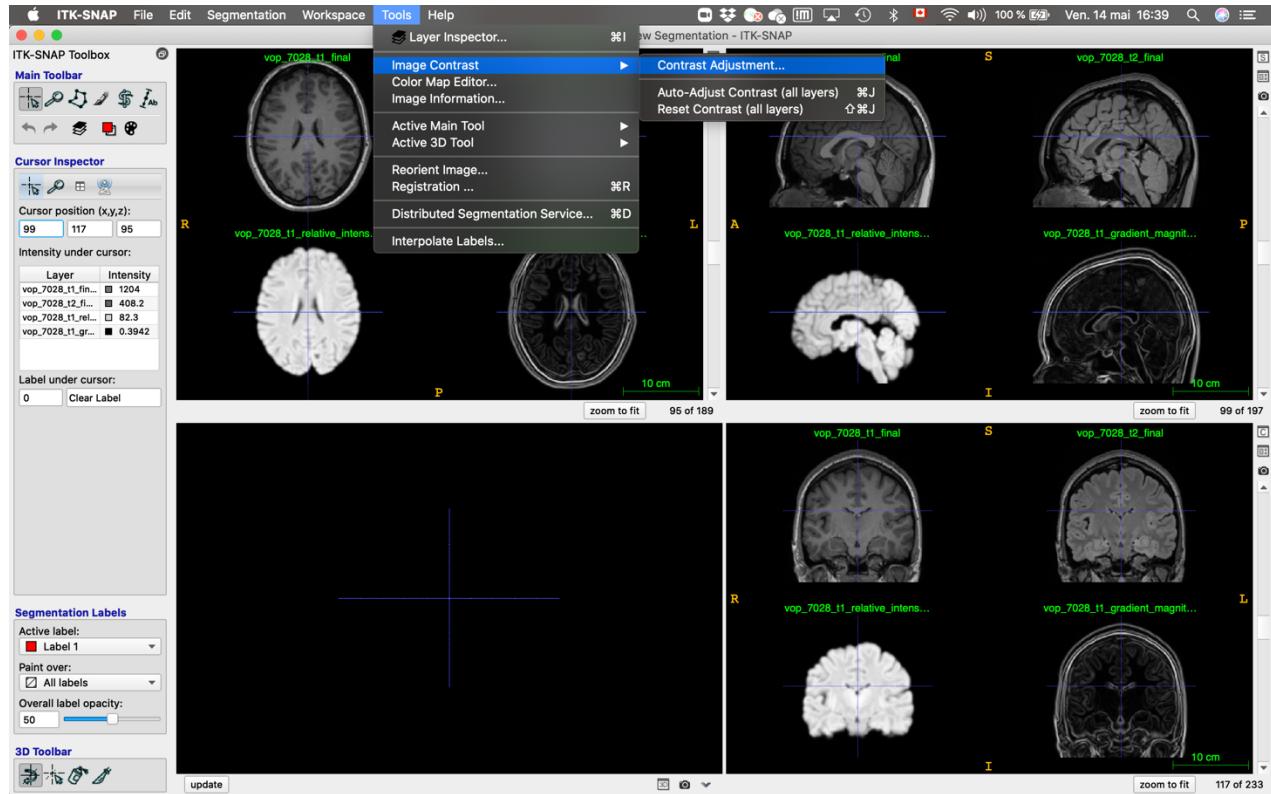
Re-click the same button to revert to the original view.

Repeat the procedure of adding additional images for *relative intensity* and *gradient*.

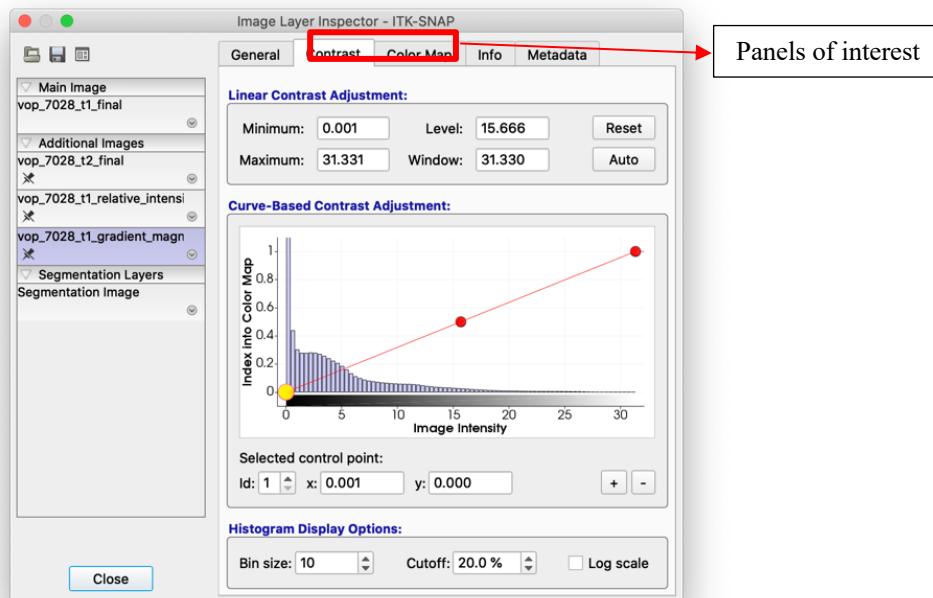


4.2. Adjusting image color and contrast

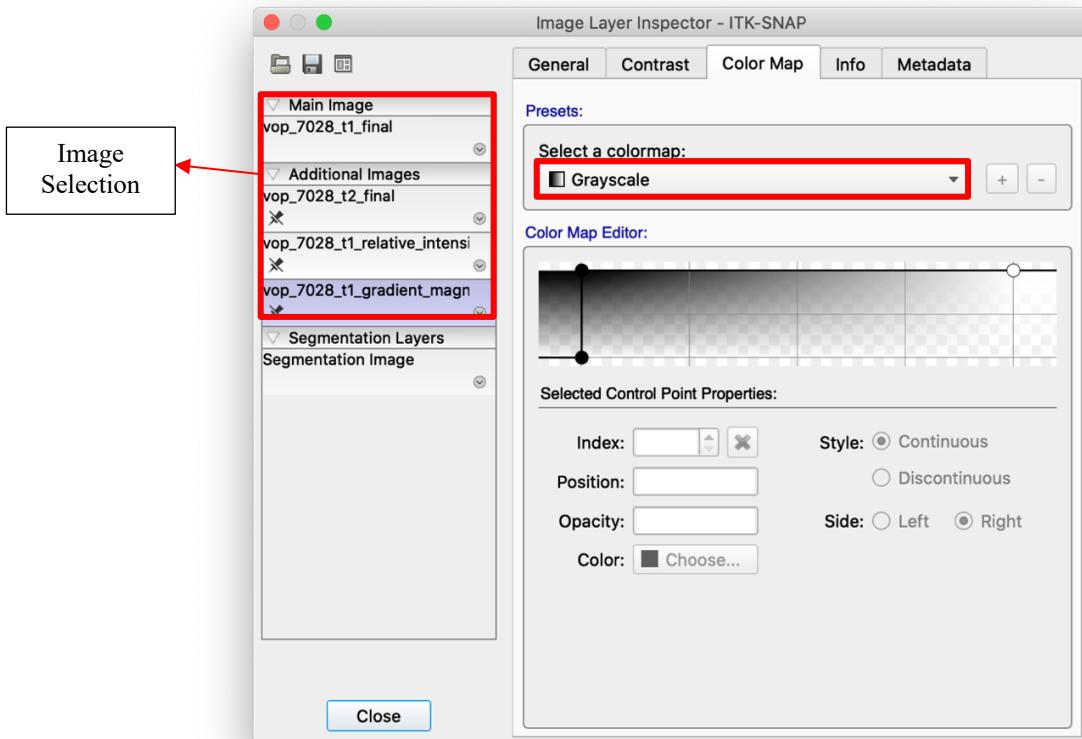
Select “Tools > Image Contrast/Contrast Adjustment...”.



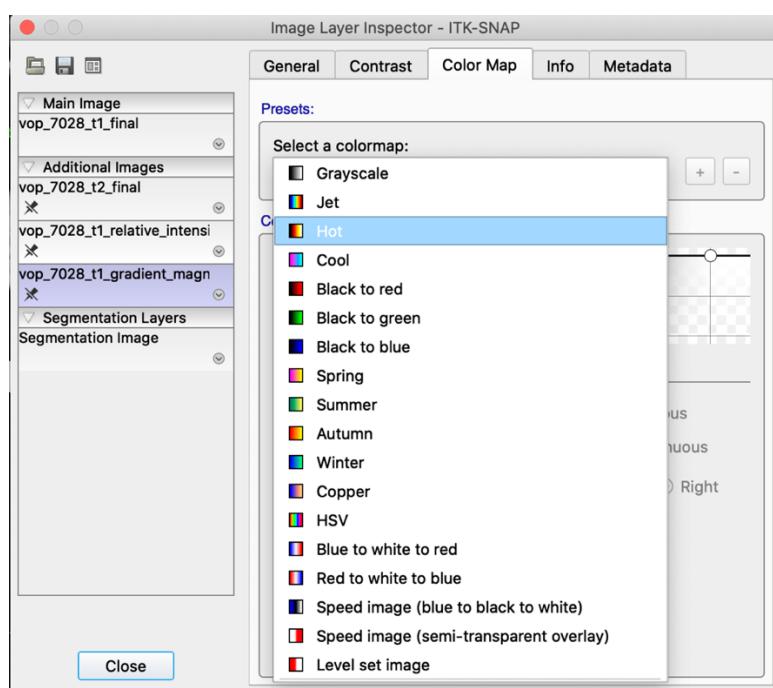
The following dialog box will appear:



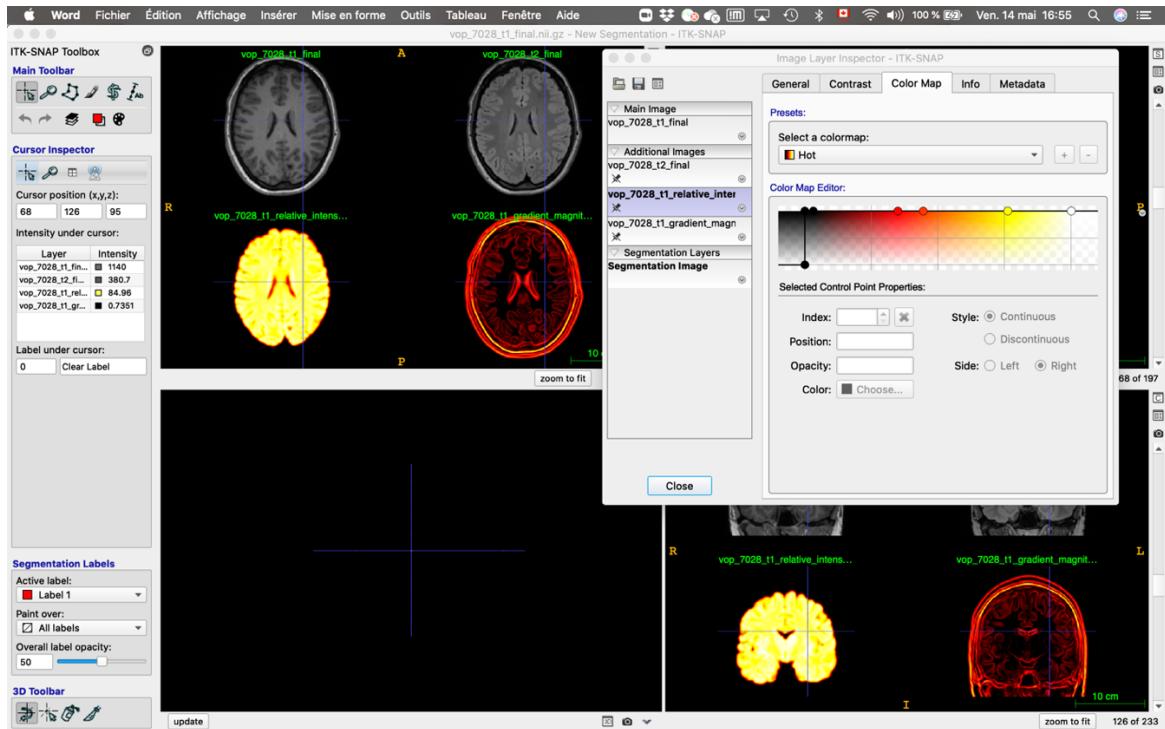
First, select the “Color Map” panel. Then, select *relative intensity* in the left panel.



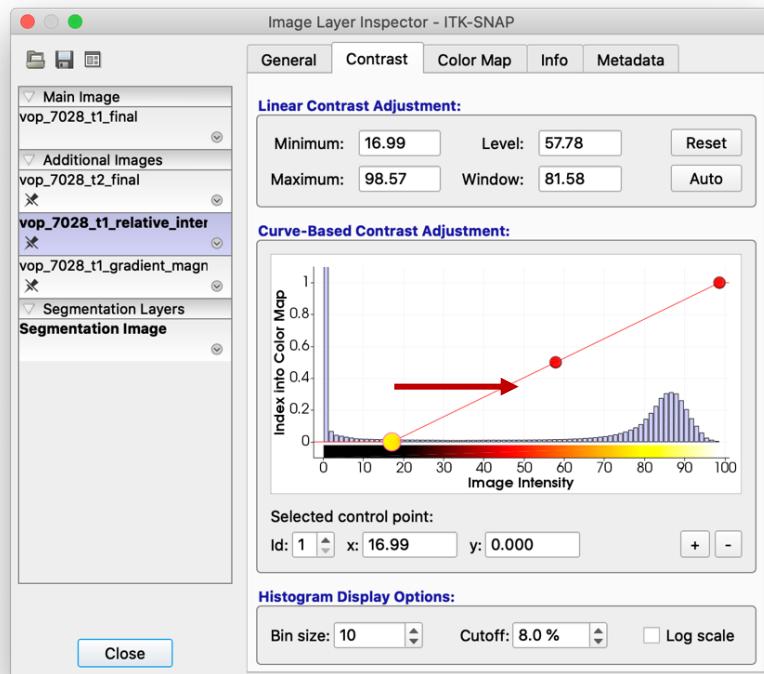
Next, click on “Select a colormap” and select “Hot”.



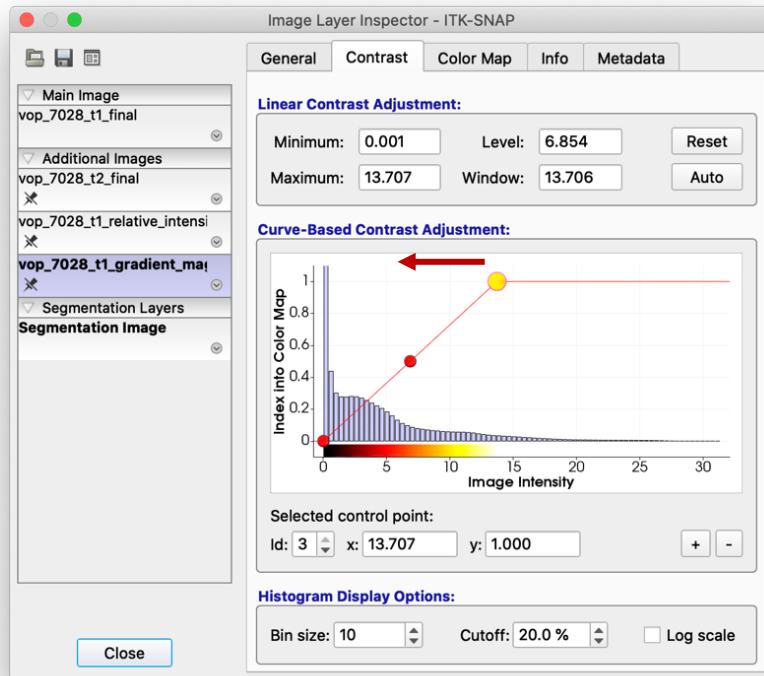
Repeat this for the gradient image. The viewer should now be organized as illustrated below.



Next, access the “Contrast” Panel. To adjust the contrast of the *relative intensity* image, you can slide the point (solid yellow dot) situated on the left of the intensity histogram to the right.



Conversely, slide the point on the right of the intensity histogram towards the left to adjust the gradient contrast.



Your images are now ready to be reviewed.

Important points when evaluating a case

The NOEL Texture Pipeline **should be used only when a focal cortical dysplasia type II is suspected.** It is not recommended to use this pipeline for the detection of any other epileptogenic lesions, such as hippocampal sclerosis, heterotopias or tumors.

The output images of the pipeline include 1) T1; 2) FLAIR; 3) gradient magnitude map (*modeling blurring: darker regions compared to the surroundings*); 4) relative intensity map (*showing the hyperintensity of the FCD relative to the rest of the brain GM and WM*).

Maps are not showing only changes related to the FCD. The relative intensity maps display hyperintensities in other brain regions (including the central areas, insula, mesiotemporal lobe, the basal ganglia and thalamus), due to biological reasons other than the presence of an FCD. Detection of an FCD in these areas are more difficult.

On the gradient magnitude map, the contour of the ventricles appears as a bright signal because of the strong contrast between WM and CSF.

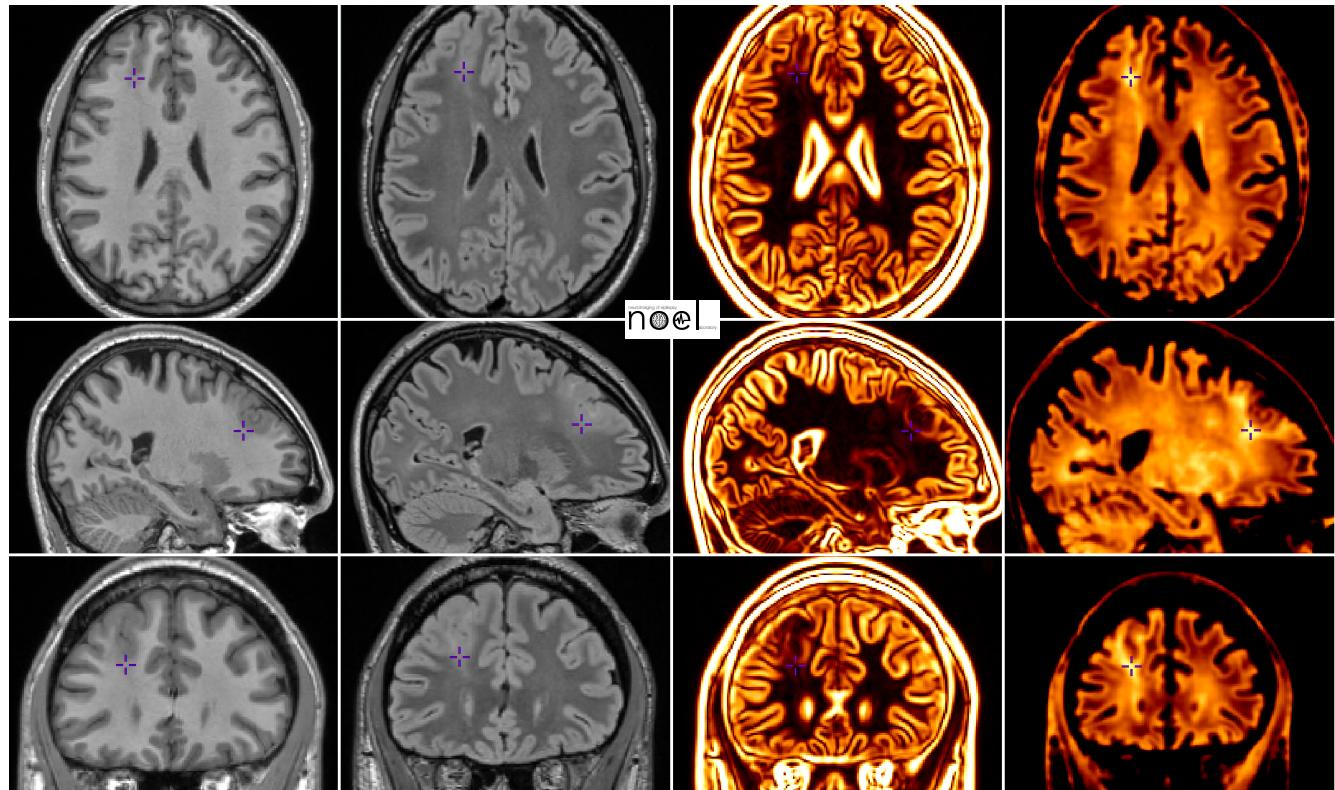
FCD type II lesions are typically characterized by the **co-occurrence of blurring (dark cortex on gradient map) and hyperintensities (on relative intensity map)**.

In general, any lesion highlighted on texture maps should be visible on conventional T1 and FLAIR images.

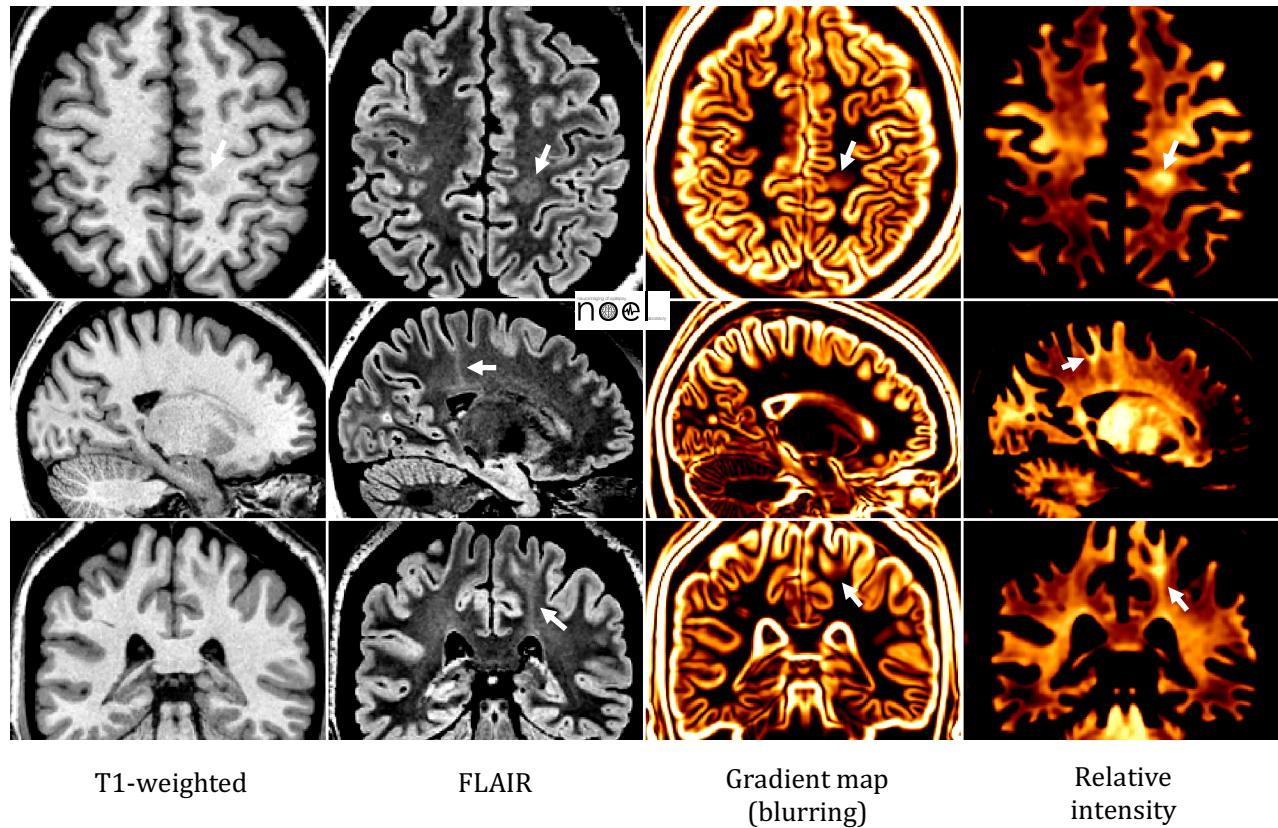
The following pages show several examples of FCD lesions of various size and locations (indicated by the cross or white arrows). All lesions shown here were confirmed histologically after surgical resection to treat drug-resistant seizures.

Case presentations

Case 1

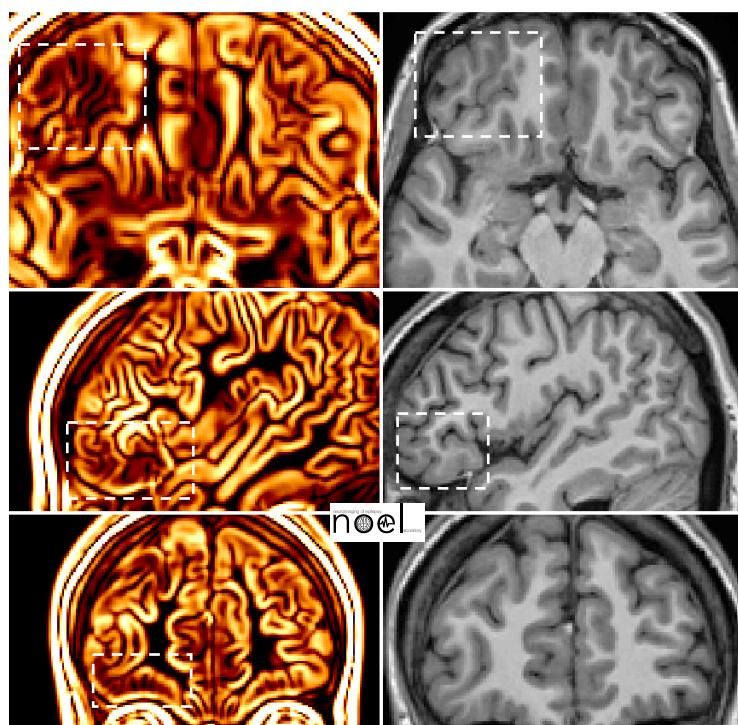
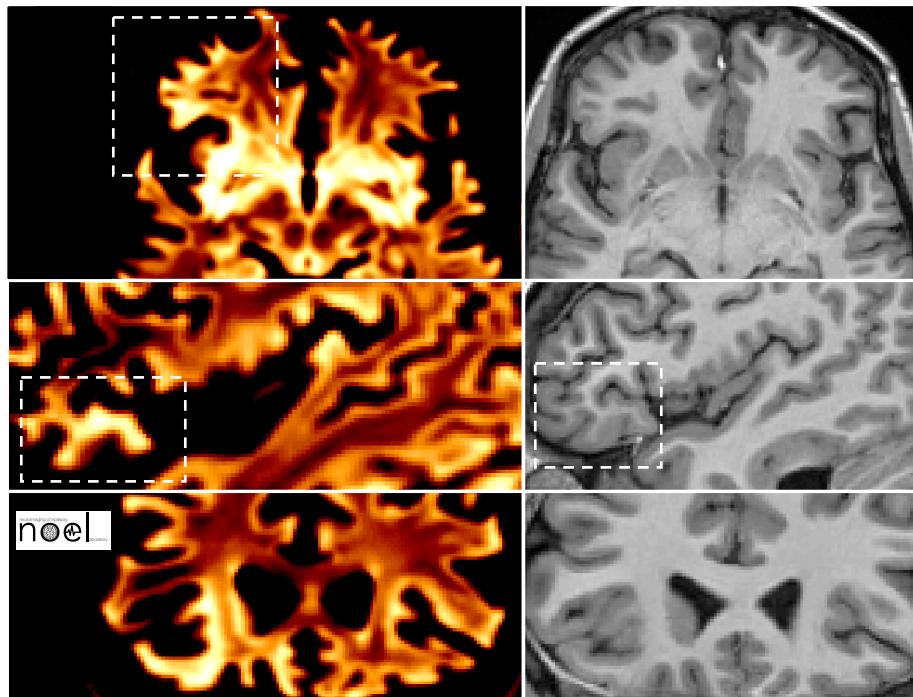


Case 2



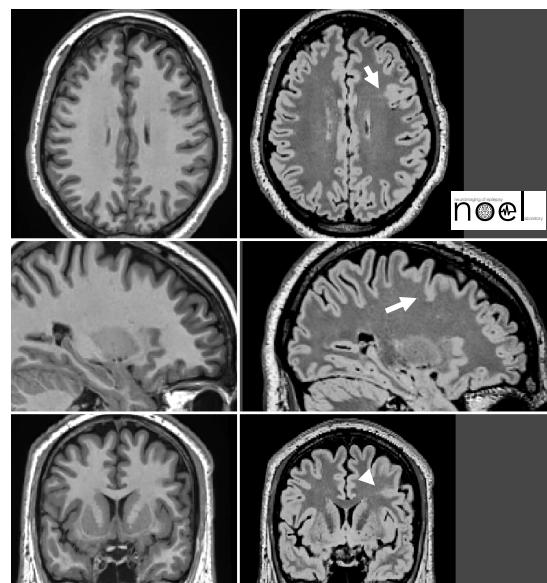
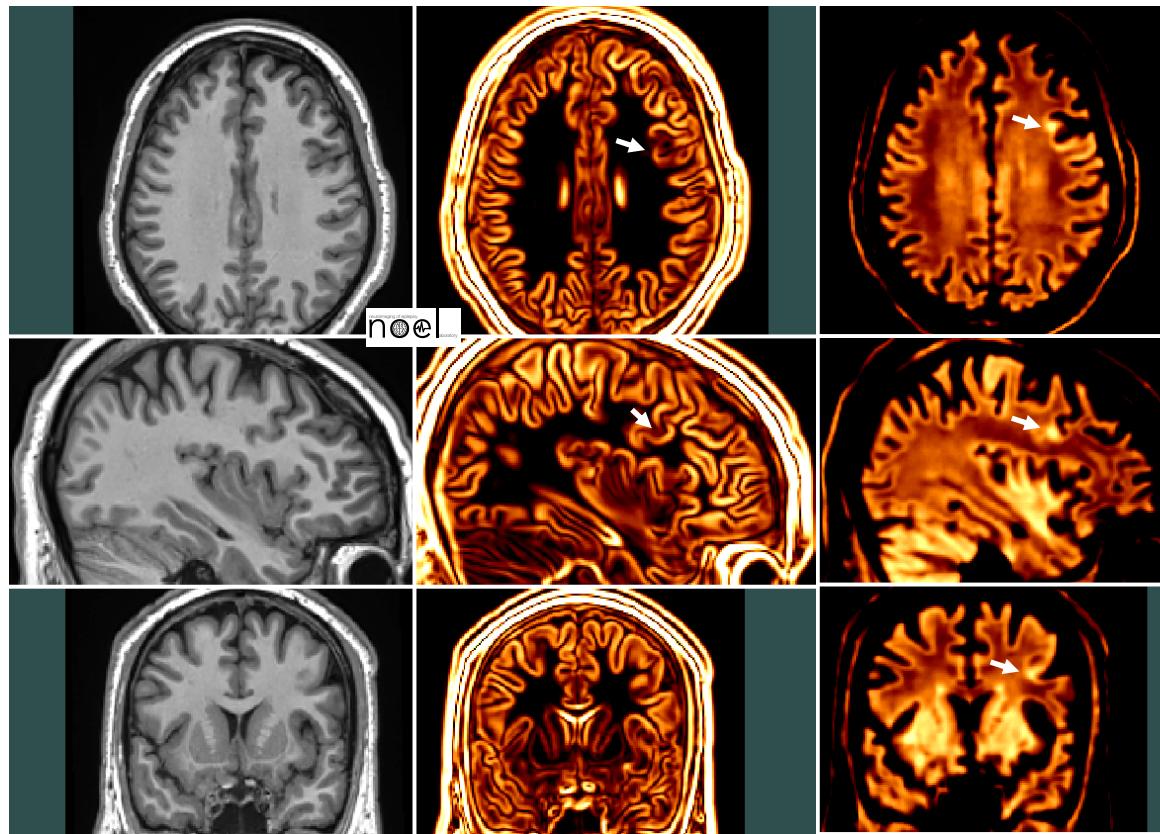
Case 3

This case demonstrates that large lesions may also be difficult to see, not only small lesions



Case 4

Small FCD, not visible on T1; on the other hand, FLAIR images show the subtle transmantle sign. The texture maps show both FCD-related blurring and hyperintensity.



Case 5

